The Development and Evaluation of Gaze Selection Techniques

Martin Stephen van Tonder

Submitted in fulfilment of the requirements for the degree of Philosophiae Doctor in the Faculty of Science at the Nelson Mandela Metropolitan University

April 2009

Promoter: Prof. C.B. Cilliers
Co Promoter: Prof. J.H. Greyling
Acknowledgments

I would like to thank my promoters Prof. Charmain Cilliers and Prof. Jean Greyling for the feedback, guidance and support they provided during this research.

I also wish to thank Danie Venter of the NMMU Unit for Statistical Consultation for providing advice regarding the statistical techniques used to analyse the data.

I would like to thank the NMMU Centre of Excellence (CoE) for the use of the usability lab and eye tracking equipment.
Summary

Eye gaze interaction enables users to interact with computers using their eyes. A wide variety of eye gaze interaction techniques have been developed to support this type of interaction. Gaze selection techniques, a class of eye gaze interaction techniques which support target selection, are the subject of this research. Researchers developing these techniques face a number of challenges. The most significant challenge is the limited accuracy of eye tracking equipment (due to the properties of the human eye).

The design of gaze selection techniques is dominated by this constraint. Despite decades of research, existing techniques are still significantly less accurate than the mouse. A recently developed technique, EyePoint, represents the state of the art in gaze selection techniques. EyePoint combines gaze input with keyboard input. Evaluation results for this technique are encouraging, but accuracy is still a concern. Early trigger errors, resulting from users triggering a selection before looking at the intended target, were found to be the most commonly occurring errors for this technique.

The primary goal of this research was to improve the usability of gaze selection techniques. In order to achieve this goal, novel gaze selection techniques were developed. New techniques were developed by combining elements of existing techniques in novel ways. Seven novel gaze selection techniques were developed. Three of these techniques were selected for evaluation.

A software framework was developed for implementing and evaluating gaze selection techniques. This framework was used to implement the gaze selection techniques developed during this research. Implementing and evaluating all of the techniques using a common framework ensured consistency when comparing the techniques. The novel techniques which were developed were evaluated against EyePoint and the mouse using the framework.

The three novel techniques evaluated were named TargetPoint, StaggerPoint and ScanPoint. TargetPoint combines motor space expansion with a visual feedback
highlight whereas the StaggerPoint and TargetPoint designs explore novel approaches to target selection disambiguation.

A usability evaluation of the three novel techniques alongside EyePoint and the mouse revealed some interesting trends. TargetPoint was found to be more usable and accurate than EyePoint. This novel technique also proved more popular with test participants. One aspect of TargetPoint which proved particularly popular was the visual feedback highlight, a feature which was found to be a more effective method of combating early trigger errors than existing approaches. StaggerPoint was more efficient than EyePoint, but was less effective and satisfying. ScanPoint was the least popular technique.

The benefits of providing a visual feedback highlight and test participants' positive views thereof contradict views expressed in existing research regarding the usability of visual feedback. These results have implications for the design of future gaze selection techniques.

A set of design principles was developed for designing new gaze selection techniques. The designers of gaze selection techniques can benefit from these design principles by applying them to their techniques.

**Keywords:** Gaze Selection Techniques, Eye Gaze Interaction, Interaction Techniques, TargetPoint, EyePoint
# Table of Contents

## Chapter 1: Introduction

1.1 Background ........................................................................................................ 1  
1.2 Relevance of Research ..................................................................................... 3  
1.3 Research Goal and Objectives ........................................................................... 4  
1.4 Structure of Hypotheses .................................................................................... 6  
1.5 Scope and Limitations ....................................................................................... 7  
1.6 Thesis Structure ................................................................................................. 8

## Chapter 2: Introduction to Eye Tracking

2.1 Introduction ....................................................................................................... 10  
2.2 Human Eye Movement .................................................................................... 10  
2.3 Eye Tracking Technology ................................................................................ 12  
  2.3.1 Electrooculography ..................................................................................... 13  
  2.3.2 Contact lens / suction cap techniques ........................................................ 15  
  2.3.3 Video-based Techniques ............................................................................ 15  
2.4 Issues and Limitations ....................................................................................... 17  
  2.4.1 Intrusiveness of Eye Tracking .................................................................... 17  
  2.4.2 Sensor Lag .................................................................................................. 21  
  2.4.3 Cost .............................................................................................................. 21  
  2.4.4 Limited Tracking Robustness ..................................................................... 21  
  2.4.5 Tracking Inaccuracy ................................................................................... 22  
2.5 Conclusions ....................................................................................................... 24

## Chapter 3: Gaze Selection Techniques

3.1 Introduction ....................................................................................................... 25  
3.2 Strengths and Weaknesses of Eye Gaze as an Input Modality ......................... 26  
  3.2.1 Strengths .................................................................................................... 26  
  3.2.2 Weaknesses ............................................................................................... 27  
3.3 Selection Triggers ............................................................................................. 27  
  3.3.1 Blinking ...................................................................................................... 28  
  3.3.2 Dwell Time ................................................................................................. 28
# Table of Contents

1. Introduction........................................................................................................... 3
2. Gaze Selection ....................................................................................................... 3
2.1 Types of Gaze Interfaces ....................................................................................... 3
2.1.1 Gaze Only Interfaces ......................................................................................... 3
2.1.1.1 Eye Typing ........................................................................................................... 3
2.1.1.2 pEyes .................................................................................................................... 3
2.1.2 Gaze Added Interfaces ....................................................................................... 35
2.1.2.1 A Gaze Added Graphical User Interface ............................................................ 36
2.1.2.2 MAGIC Pointing .................................................................................................. 36
2.1.3 Comparison of Gaze Only and Gaze Added Interfaces ....................................... 38
2.2 Interaction Using Target Expansion or Magnification ......................................... 40
2.2.1 Target Expansion ............................................................................................. 40
2.2.1.1 Static Expansion in Motor Space ........................................................................ 41
2.2.1.2 Dynamic Expansion of Menu Items ................................................................. 43
2.2.1.3 Static Visual and Motor Space Expansion ......................................................... 45
2.2.2 Magnification ..................................................................................................... 45
2.2.2.1 Full Screen Magnification .................................................................................. 46
2.2.2.2 ERICA (Pop-Up Magnification) .......................................................................... 47
2.2.2.3 EyePoint (Pop-Up Magnification) ...................................................................... 48
2.2.2.4 Fisheye Lens Magnification ............................................................................... 52
2.2.3 Comparison of Magnification and Target Expansion ........................................ 55
2.3 Intelligent Gaze Interfaces ................................................................................... 56
2.4 Feedback for Gaze Selection ............................................................................... 58
2.5 Conclusions ........................................................................................................... 62

## Chapter 4: Design of Gaze Selection Techniques .................................................... 65

4.1 Introduction ........................................................................................................... 65
4.2 Design Principles .................................................................................................. 65
4.2.1 Design Principles Based on Literature ............................................................. 66
4.2.2 Proposed Design Principles for Visual Feedback ............................................ 67
# Table of Contents

4.3 Novel Gaze Selection Techniques .......................................................... 68
  4.3.1 TargetPoint ......................................................................................... 68
  4.3.2 ScanPoint ............................................................................................ 75
  4.3.3 StaggerPoint ........................................................................................ 78
  4.3.4 Additional Technique Prototypes ......................................................... 81
    4.3.4.1 Gaze Proximity Typing ................................................................. 81
    4.3.4.2 Gaze Scroll Wheel Selection ......................................................... 82
    4.3.4.3 Gaze-Joystick Selection ................................................................. 82
    4.3.4.4 Tagged Pop-up Selection .............................................................. 83

4.4 Conformance of Techniques to Design Principles ..................................... 84

4.5 Conclusions ............................................................................................. 85

Chapter 5: A Framework for Gaze Selection Techniques ................................... 88
  5.1 Introduction ............................................................................................. 88
  5.2 Relevance of Framework ......................................................................... 89
  5.3 Framework Overview ............................................................................. 90
  5.4 Framework Requirements ...................................................................... 92
    5.4.1 General Requirements ..................................................................... 92
    5.4.2 Gaze Library Requirements ............................................................ 92
    5.4.3 Gaze Selection Library Requirements ............................................ 93
  5.5 Design and Implementation .................................................................... 94
  5.6 Gaze Library .......................................................................................... 95
  5.7 Gaze Selection Library ........................................................................... 97
    5.7.1 Implementing Selection Techniques ................................................ 98
    5.7.2 The GazeTarget Class ..................................................................... 98
    5.7.3 Automated Testing .......................................................................... 102
    5.7.4 Data Recording ................................................................................ 105
  5.8 Implementing TargetPoint using the Framework ...................................... 107
  5.9 Conclusions .......................................................................................... 109

Chapter 6: Experimental Design .................................................................... 111
  6.1 Introduction ........................................................................................... 111
  6.2 Hypotheses ............................................................................................ 112
<table>
<thead>
<tr>
<th>Chapter 6: Pilot Evaluation</th>
<th>112</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3 Pilot Evaluation</td>
<td></td>
</tr>
<tr>
<td>6.4 Usability Evaluation Design</td>
<td>113</td>
</tr>
<tr>
<td>6.4.1 Test Venue and Equipment</td>
<td>113</td>
</tr>
<tr>
<td>6.4.2 Naming of Techniques Evaluated</td>
<td>114</td>
</tr>
<tr>
<td>6.4.3 Test Procedure</td>
<td>115</td>
</tr>
<tr>
<td>6.4.4 Collection of Usability Metrics</td>
<td>118</td>
</tr>
<tr>
<td>6.4.5 Data Analysis Tools and Methods</td>
<td>120</td>
</tr>
<tr>
<td>6.5 Conclusions</td>
<td>121</td>
</tr>
</tbody>
</table>

**Chapter 7: Experimental Results**                                      123

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Introduction</td>
<td>123</td>
</tr>
<tr>
<td>7.2 Test Participants</td>
<td>123</td>
</tr>
<tr>
<td>7.3 System Recorded Metrics</td>
<td>124</td>
</tr>
<tr>
<td>7.3.1 Effectiveness</td>
<td>124</td>
</tr>
<tr>
<td>7.3.2 Efficiency</td>
<td>131</td>
</tr>
<tr>
<td>7.4 Self Reported Metrics</td>
<td>134</td>
</tr>
<tr>
<td>7.4.1 Perceived Effectiveness</td>
<td>135</td>
</tr>
<tr>
<td>7.4.2 Perceived Efficiency</td>
<td>138</td>
</tr>
<tr>
<td>7.4.3 Satisfaction</td>
<td>140</td>
</tr>
<tr>
<td>7.4.4 Usefulness of Visual Feedback</td>
<td>148</td>
</tr>
<tr>
<td>7.5 Qualitative Data</td>
<td>148</td>
</tr>
<tr>
<td>7.5.1 TargetPoint</td>
<td>148</td>
</tr>
<tr>
<td>7.5.2 EyePoint</td>
<td>151</td>
</tr>
<tr>
<td>7.5.3 Mouse</td>
<td>152</td>
</tr>
<tr>
<td>7.5.4 StaggerPoint</td>
<td>153</td>
</tr>
<tr>
<td>7.5.5 ScanPoint</td>
<td>154</td>
</tr>
<tr>
<td>7.6 Testing of Hypotheses</td>
<td>155</td>
</tr>
<tr>
<td>7.6.1 TargetPoint versus EyePoint</td>
<td>156</td>
</tr>
<tr>
<td>7.6.2 StaggerPoint versus EyePoint</td>
<td>157</td>
</tr>
<tr>
<td>7.6.3 ScanPoint versus EyePoint</td>
<td>158</td>
</tr>
<tr>
<td>7.6.4 TargetPoint versus Mouse</td>
<td>159</td>
</tr>
<tr>
<td>7.6.5 StaggerPoint versus Mouse</td>
<td>160</td>
</tr>
<tr>
<td>Chapter</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>7.6.6</td>
<td>ScanPoint versus Mouse</td>
</tr>
<tr>
<td>7.7</td>
<td>Design Principles Revisited</td>
</tr>
<tr>
<td>7.8</td>
<td>Conclusions</td>
</tr>
<tr>
<td>Chapter 8: Conclusion</td>
<td>Conclusion</td>
</tr>
<tr>
<td>8.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>8.2</td>
<td>Summary of Findings</td>
</tr>
<tr>
<td>8.2.1</td>
<td>Hypotheses</td>
</tr>
<tr>
<td>8.2.2</td>
<td>Goal and Objectives</td>
</tr>
<tr>
<td>8.2.3</td>
<td>Design Principles</td>
</tr>
<tr>
<td>8.3</td>
<td>Summary of Contributions</td>
</tr>
<tr>
<td>8.3.1</td>
<td>Practical Contributions</td>
</tr>
<tr>
<td>8.3.2</td>
<td>Theoretical Contributions</td>
</tr>
<tr>
<td>8.4</td>
<td>Limitations of Investigation</td>
</tr>
<tr>
<td>8.5</td>
<td>Future Research</td>
</tr>
<tr>
<td>References</td>
<td>References</td>
</tr>
<tr>
<td>Appendix A: Pre-test Questionnaire</td>
<td>Appendix A: Pre-test Questionnaire</td>
</tr>
<tr>
<td>Appendix B: Post-test Questionnaire</td>
<td>Appendix B: Post-test Questionnaire</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1: An Eye Tracking System from the 1960s (Yarbus 1967)..................................................1
Figure 2.1: The Structure of the Human Eye (Ohno et al. 2002)...............................................................11
Figure 2.2: The Gaze Direction Vector (Ohno and Mukawa 2004).........................................................13
Figure 2.3: An EOG Tracker (AG Research 2008)..................................................................................14
Figure 2.4: Eye Tracking using a Contact Lens and Wire Coil (MIT 2008)..............................................14
Figure 2.5: Bright Pupil Image (Ji and Zhu 2002)..................................................................................16
Figure 2.6: Dark Pupil Image (Ji and Zhu 2002)..................................................................................16
Figure 2.7: An Eye Tracker Calibration Screen (Ohno and Mukawa 2004)..............................................19
Figure 2.8: An Example of a Head-Mounted Eye Tracker (Yun et al. 2008)............................................19
Figure 2.9: A Tobii T60 Eye Tracker........................................................................................................20
Figure 2.10: An Eye Tracker with a Chin Rest (ASL 2008)....................................................................21
Figure 2.11: The Effect of Saccade Detection and Fixation Smoothing..............................................23
Figure 3.1: Gaze Selection Triggers........................................................................................................28
Figure 3.2: An On-Screen Keyboard Eye Typing System (Majaranta et al. 2004).................................33
Figure 3.3: pEYEwrite: Text Input using pEyes ....................................................................................35
Figure 3.4: Target Expansion Techniques.............................................................................................40
Figure 3.5: Motor Space Expansion.........................................................................................................41
Figure 3.6: Static Target Expansion (Miniota et al. 2004)......................................................................42
Figure 3.7: Target Expansion during Menu Item Selection (Špakov and Miniota 2005).....................43
Figure 3.8: A Taxonomy of Existing Magnification Techniques...............................................................46
Figure 3.9: Zoom Levels Selected by Users for Controls of Various Sizes (Bates 1999).....................47
Figure 3.10: Dwell Selection with Magnification ....................................................................................48
Figure 3.11: The Magnified EyePoint Popup with Gaze Marker.............................................................49
Figure 3.12: The Magnified EyePoint Popup without Gaze Marker (Kumar et al. 2007).......................50
Figure 3.13: The Effect of Early Trigger Correction on Error Rates (Kumar et al. 2008)......................52
Figure 3.14: A Fisheye Lens Magnifying at the Point of Fixation............................................................53
Figure 3.15: Efficiency – Mean Selection Times with Fisheye Lens (Ashmore et al. 2005)..............54
Figure 3.16: The Size of Targets in Motor Space Varies Based on Current Task Context ..............57
Figure 3.17: Animation of Visual Feedback – Shrinking Letter (Majaranta et al. 2003)....................59
Figure 3.18: Eye Typing Feedback Options (Majaranta et al. 2004).....................................................60
Figure 4.1: Motor Space Expansion Example - TargetPoint....................................................................70
Figure 4.2: Magnification Example - EyePoint.....................................................................................70
Figure 4.3: TargetPoint Pop-up Activation and Visual Feedback Highlight........................................72
List of Figures

Figure 4.4: Target Placement Within TargetPoint Pop-up..................................................74
Figure 4.5: A Selection Highlight for ScanPoint.................................................................75
Figure 4.6: Initial Design Featuring Moving Items..............................................................79
Figure 4.7: Improved Design Featuring Static Items..........................................................80
Figure 4.8: The Tagged Pop-up Technique.........................................................................83
Figure 5.1: Framework Overview....................................................................................91
Figure 5.2: The GazePoint Interface..............................................................................95
Figure 5.3: The GazePointListener Interface.................................................................96
Figure 5.4: The GazeDataProvider Interface......................................................................97
Figure 5.5: The GazeTarget Class and Related Subclasses..............................................98
Figure 5.6: The GazeTarget Class...................................................................................99
Figure 5.7: The Selection Sequence................................................................................100
Figure 5.8: The TestManager Class...............................................................................103
Figure 5.9: A Test Planner Built Using the Framework......................................................104
Figure 5.10: An Example of a Visualisation of Captured Data.........................................106
Figure 5.11: Customised Analysis Software Example.......................................................107
Figure 5.12: TargetPoint Implementation Screen Capture.............................................108
Figure 6.1: A Tobii T60 Eye Tracker in the NMMU Usability Laboratory........................114
Figure 6.2: The Test Setup (image cropped)....................................................................116
Figure 7.1: Error Percentage for Accuracy Condition......................................................125
Figure 7.2: Selection Errors by Type (all test conditions)...............................................127
Figure 7.3: Trigger Errors versus Trigger Time Shift......................................................128
Figure 7.4: Average Distance from Target Centre.............................................................129
Figure 7.5: The Effect of Staggered Gaze Interpretation..................................................130
Figure 7.6: The Effect of Timed Highlight Correction......................................................130
Figure 7.7: Mean Selection Times (Speed Condition).......................................................131
Figure 7.8: Mean Selection Rate (Speed Condition).........................................................133
Figure 7.9: Perceived Accuracy.....................................................................................137
Figure 7.10: Perceived Efficiency..................................................................................138
Figure 7.11: Ease of Use............................................................................................140
Figure 7.12: Satisfaction..............................................................................................142
Figure 7.13: User Preference Ranking Distribution.........................................................143
Figure 7.14: Perceived Mental Effort (lower is preferred)...............................................145
Figure 7.15: Perceived Physical Effort (lower is preferred).............................................147
List of Tables

Table 3.1: Advantages and Disadvantages of Different Selection Triggers ........................................31
Table 3.2: Comparison of Gaze Interfaces .........................................................................................39
Table 3.3: Advantages and Disadvantages of Magnification and Target Expansion .........................55
Table 3.4: Advantages and Disadvantages of Intelligent Gaze Interfaces ........................................58
Table 3.5: Advantages and Disadvantages of Feedback in an Eye Gaze Interface .........................62
Table 7.1: Test Participant Summary .................................................................................................124
Table 7.2: Mean Error Percentage (Accuracy Condition) .................................................................126
Table 7.3: Mean Selection Times in Milliseconds (Speed Condition) ..............................................132
Table 7.4: Mean Efficiency (Correct Selections per Minute) (Speed Condition) ..............................134
Table 7.5: Self Reported Metrics (5 point Likert scale) .....................................................................135
Table 7.6: Average Perceived Accuracy Ratings ..............................................................................136
Table 7.7: Average Ratings for Perceived Accuracy Decline for Smaller Targets ............................137
Table 7.8: Average Perceived Efficiency Ratings .............................................................................139
Table 7.9: Average Ease of Use Ratings (5 Point Likert Scale, Higher Values Preferred) .............141
Table 7.10: Average Satisfaction Ratings (5 Point Likert Scale, Higher Values Preferred) ..............142
Table 7.11: Average Preference Rankings (Rankings from 1 to 5, lower values preferred) ............144
Table 7.12: Average Perceived Mental Effort Ratings .....................................................................146
Table 7.13: Average Perceived Physical Effort Ratings .................................................................147
Table 7.14: Results for Hypotheses H0.x (i, EyePoint) .....................................................................159
Table 7.15: Results for Hypotheses H0.x (i, Mouse) .......................................................................163
Chapter 1: Introduction

1.1 Background
In 1965 the idea of using eye movements as a form of input was born (Sutherland 1965). The thought of using your eyes to control a computer was appealing, but the tracking equipment available at the time was clearly not suitable for the task (Figure 1.1).

![Figure 1.1: An Eye Tracking System from the 1960s (Yarbus 1967)](image)

The concept of eye gaze interaction received renewed attention in the 1980s and 1990s. Papers published during this period explored different possibilities for the use of gaze input (Bolt 1981, 1982, Ware and Mikaelian 1987, Jacob 1990). This research,
together with improvements in eye tracking technology, led to the development of eye
gaze interaction techniques for a variety of tasks.

Several terms have been used by researchers to describe interaction using eye gaze.
Examples include eye gaze interaction, eye gaze based input, gaze based interaction,
gaze input, gaze interaction and eye interaction. For the purpose of consistency, the
term eye gaze interaction (or gaze interaction for short) will be preferred for this
manuscript. Some gaze interaction techniques can be used for selecting discrete
targets, such as buttons or menu items. The term gaze selection technique will be
used to describe such techniques. This class of eye gaze interaction technique is the
focus of this research. Another common type of interaction technique within the
domain of eye gaze interaction is eye pointing. This term refers to the use of gaze as a
general purpose pointing device. Eye pointing techniques can also be used for
selecting discrete targets.

People with disabilities are the primary users of gaze interaction systems today. Eye
typing is the most common application of gaze interaction (Majaranta et al. 2002).
Users with disabilities are able to enter text using their eyes. Systems designed for
disabled users can also support a wider range of interactions, including pointing and
selection (Lankford 2000). Recent eye gaze interaction research has focused on
practical interaction for able bodied users (Kumar et al. 2007, Kumar et al. 2008).

Literature suggests that eye gaze interaction may provide usability benefits over
traditional modalities. One advantage is that, unlike with the mouse, users do not need
to move their hands between the mouse and keyboard (Ware and Mikaelian 1987,
Salvucci and Anderson 2000). When using the mouse, users normally need to look at
an object before using the cursor to select (Zhai et al. 1999, Smith et al. 2000). Using
gaze interaction, the need to move the cursor (using the mouse) to where the user is
looking could potentially be eliminated. It has also been argued that the goal of eye
tracking interfaces is not improved efficiency, but to provide the user with the
perception of a very responsive system, which creates the impression that it is reading
the mind of the user (Sibert and Jacob 2000, Jacob and Karn 2003). Therefore an eye
gaze system which can match the efficiency of a mouse is viewed as performing well,
and any efficiency gains are a bonus. The ease with which users can simply look at objects, and the ability to get a better idea of where the user’s attention is focused are further advantages of gaze interaction (Jacob 1993). The eye can also move very rapidly (Zhai et al. 1999) and can free the users' hands for other tasks (Sibert and Jacob 2000, Zhai et al. 1999). As stated earlier, eye gaze interaction can also improve accessibility for users with disabilities (Pomplun et al., 2006), although this is not a goal of this research. Another benefit of eye gaze interaction is that it can reduce the risk of repetitive stress injuries (Zhai et al. 1999).

Disadvantages include difficulties controlling the eye, including involuntary eye movement, the always on nature of eye tracking, the lack of a button compared to that offered by the mouse, and the inaccuracy and instability of eye tracking compared to traditional input devices (Jacob 1993). One well known problem, also discussed by Jacob is the “Midas touch”. If commands can be triggered by simply looking at an interface then users can unintentionally execute commands wherever they look. The use of eye gaze as input can also result in a decrease in the efficiency of a user interface (Pomplun et al. 2001).

A recent development in the field of eye gaze interaction is EyePoint (Kumar et al. 2007). This technique represents a practical eye pointing technique with high levels of user satisfaction, efficiency, and reasonable effectiveness (including accuracy). Results published for this technique indicate a combination of accuracy, efficiency and satisfaction which has yet to be equalled (Kumar et al. 2007, Kumar et al. 2008). The limited accuracy of this technique, although equal to or better than existing gaze selection techniques, is still a cause for concern when compared with the mouse.

1.2 Relevance of Research

The primary challenge currently facing interaction designers wishing to incorporate eye gaze interaction into a system is the limited accuracy of eye tracking equipment (Chapter 2). The designs of existing gaze selection techniques reflect the need to compensate for this limitation (Chapter 3). Enlarged targets (Salvucci and Anderson 2000), full screen magnification (Bates and Istance 2002), fisheye lens magnification (Ashmore et al. 2005), and intelligent gaze interpretation (Salvucci and Anderson
2000) are just some of the approaches which have been employed in order to reduce the number of selection errors. Higher accuracy is sometimes achieved at the cost of efficiency (Špakov and Miniotas 2005). In order to improve accuracy, a selection operation may require more than one step (Lankford 2000, Kumar et al. 2007). Researchers need to strike a balance between efficiency, effectiveness (including accuracy) and satisfaction.

Despite recent advances towards more practical and usable techniques, such as EyePoint, the mouse is still significantly more accurate than gaze for selecting targets (Kumar et al. 2007). The most common cause of selection errors in EyePoint is early trigger errors (Kumar et al. 2008). Users release a trigger key (Section 3.3.4) before they are actually looking at a target, resulting in a selection error. EyePoint (Section 3.5.2.3) arguably represents the state of the art in gaze selection techniques, but like other techniques suffers from accuracy limitations compared to the mouse.

Given the limitations of existing techniques, there is a clear need for new or improved interaction techniques for selecting objects using gaze. Minimising selection errors is an important goal, but other aspects of usability, such as efficiency and satisfaction, also need to be considered.

The lack of visual feedback in many existing gaze selection techniques is also a cause for concern as this goes against two of Nielsen's heuristics (Nielsen 1994), namely visibility of system status and error prevention. Given the importance of reducing the number of selection errors for gaze selection techniques, it would make sense to consider providing the user with visual feedback. The principle of providing visual feedback during gaze selection is not widely supported as some researchers consider it undesirable for this input modality (Kumar et al. 2007, Zhai et al. 1999).

1.3 Research Goal and Objectives

The main goal of this research is to improve the usability of gaze selection techniques. This may be achieved by developing one or more new technique(s), which represent novel combinations of promising design elements from existing techniques.
Chapter 1: Introduction

The research objectives are divided into primary objectives and secondary objectives. A more specific set of hypotheses is presented in Section 6.2.

The **primary objectives** relate to the development of novel gaze selection techniques and comparing the usability of alternative designs to existing techniques. The primary objectives are to:

1. Develop novel gaze selection techniques which minimise selection errors and maximise usability
2. Explore novel approaches to target selection disambiguation\(^1\)
3. Compare the usability of the proposed techniques to that of existing techniques
4. Investigate the effect of visual feedback on selection errors and user satisfaction

The following **secondary objectives** are also identified in order to achieve the primary objectives:

1. Propose a set of design principles for designing gaze selection techniques
2. Develop a framework for the purpose of implementing and evaluating\(^2\) the proposed techniques

In order to achieve these objectives, a constructivist research methodology is applied. This methodology incorporates a substantial component of exploratory research. A literature survey of existing gaze selection techniques is conducted in order to determine current developments in gaze selection techniques. This survey also serves to inform the design of novel gaze selection techniques.

These novel techniques are developed based on a process of iterative refinement of prototypes. A software framework is developed and used for the implementation and evaluation of the gaze selection techniques in order to ensure a common test platform for a valid comparison. In the comparison, the mouse is used as a representative

---

\(^1\) Determining which item a user wishes to select when it is unclear due to eye tracking inaccuracy

\(^2\) Required evaluation functionality limited to automated presentation of test targets to test participants and data collection
benchmark for traditional input modalities, with EyePoint being a representative benchmark for gaze selection techniques.

A within subjects comparative usability evaluation based on existing experimental designs is conducted (Chapter 6). This experiment serves two purposes, namely to compare the usability of the novel gaze selection techniques developed with the identified benchmark techniques, and as exploratory research to identify interesting observations. Simulations are used to evaluate certain aspects of these techniques based on the data gathered by the framework during the experiment. This method of data collection and analysis is used primarily to evaluate the effectiveness of novel gaze selection disambiguation mechanisms.

A set of design principles for the design of gaze selection techniques is proposed based on documented research, prototyping of techniques and the results of the usability evaluation.

1.4 Structure of Hypotheses

The hypotheses are all related to usability. Usability consists of three components, namely effectiveness, efficiency and satisfaction (ISO 9241 1997). The International Standards Organisation (ISO) defines usability as follows:

“Usability is the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments; where effectiveness is the accuracy and completeness with which specified users achieve specified goals in particular environments; efficiency is the resources expended in relation to the accuracy and completeness of goals achieved; and satisfaction is the comfort and acceptability of the work system to its users and other people affected by its use” (ISO 9241 1997).

For each novel gaze selection technique evaluated, the usability of the technique is compared to that of EyePoint and the mouse. The hypotheses are all presented as null hypotheses to be rejected should sufficient evidence be found.

---

1 Methods for reducing target selection errors when selections are ambiguous due to tracking inaccuracy
H_0 (i, j) : Technique i is not more *usable* than technique j, i \in \{\text{techniques developed}\}, j \in \{\text{EyePoint, mouse}\}

H_{0.1} (i, j) : Technique i, is not more *efficient* than technique j,
i \in \{\text{techniques developed}\}, j \in \{\text{EyePoint, mouse}\}

H_{0.2} (i, j) : Technique i, is not more *effective* than technique j,
i \in \{\text{techniques developed}\}, j \in \{\text{EyePoint, mouse}\}

H_{0.3} (i, j) : Technique i, is not more *satisfying* than technique j,
i \in \{\text{techniques developed}\}, j \in \{\text{EyePoint, mouse}\}

The presentation of a more specific set of hypotheses is deferred until Chapter 6 (Section 6.2).

1.5 Scope and Limitations

The scope of this research is limited to gaze selection techniques for selecting discrete targets, rather than the more general concepts of eye pointing and eye gaze interaction. Three dimensional environments, such as virtual reality, and techniques which incorporate 3D graphics (Hansen *et al.* 2008) are beyond the scope of this research. Combining gaze with modalities other than the keyboard, mouse and joystick as well as the adherence of gaze selection to Fitts law (Fitts 1954) will not be investigated.

The development of gaze selection techniques designed specifically for users with disabilities is not considered. The view of Kumar *et al.* (2007) (that eye gaze interaction techniques which incorporate keyboard triggers (Section 3.3.4) can easily be adapted for disabled users should the need arise), is supported.

The goal of this research is to implement gaze selection techniques for research and evaluation purposes only, rather than for use in existing systems. One advantage of this approach is that it enables the development of techniques which incorporate semantic information regarding the underlying user interface. Evaluation tasks are restricted to the selection of rectangular targets.
In order to ensure that the scale of the research remains manageable, only the mouse and EyePoint will be used as a basis for comparison during evaluation (Chapters 6 and 7). EyePoint was selected as it currently represents the state of the art in eye gaze interaction (Chapter 3). The mouse is included in the evaluation phase as this represents the current standard for users interacting with graphical user interfaces. Despite seven novel gaze selection techniques being developed, a maximum of three techniques will be evaluated in order to limit the scope of this research.

1.6 Thesis Structure

Chapter 2 provides an introduction to eye tracking. Topics such as human eye movement, eye tracking technology and algorithms are essential for understanding some of the limitations related to eye gaze systems. The limited accuracy of eye trackers is a particularly important issue which is highlighted in this chapter.

A literature review of gaze selection techniques is provided in Chapter 3. This information is particularly important as it provides a theoretical foundation for the design of the gaze selection techniques presented in Chapter 4. The information is categorised according to the characteristics of different techniques including the different approaches to countering tracker inaccuracy. A discussion of visual feedback is provided which is relevant to the design of the techniques in Chapter 4.

The designs of the three novel gaze selection techniques which were evaluated are presented in Chapter 4. TargetPoint, StaggerPoint, and ScanPoint are all described in this chapter. The logic and theory behind the design of these techniques are discussed, as well as the iterative refinements which were implemented before the final evaluation. Additional techniques, which were implemented during the initial stages of the research (but not evaluated due to scope limitations), are briefly covered. A set of design principles for designing gaze selection techniques is proposed. These principles are used to motivate design decisions regarding the techniques proposed.

In Chapter 5 the design and development of a framework for implementing gaze selection techniques and automating the presentation of test targets to users is described. This framework is used to implement all of the interaction techniques
The purpose of the framework is to provide a common testing and implementation platform for all of the gaze selection techniques to be evaluated in this research. A common platform ensures consistency in data collection and testing. The data gathered by the framework is used during the analysis phase (Chapter 7). A secondary objective is to maximise code reusability when implementing new techniques.

The experimental design and research hypotheses are presented in more detail in Chapter 6. This is followed by a discussion of the experimental results in Chapter 7. Each of the hypotheses formulated in Chapter 6 is revisited in Chapter 7.

A summary of the findings and contributions of the research is provided in Chapter 8. The research objectives are revisited and discussed and opportunities for future research are identified.
Chapter 2: Introduction to Eye Tracking

2.1 Introduction

Eye gaze interaction relies on gaze point data provided by eye tracking equipment. This chapter provides a brief introduction to human eye movement (Section 2.2) followed by a more detailed discussion on eye tracking technology (Section 2.3). The limitations of this technology and the properties of human eye movement have a negative impact on the effectiveness of gaze interaction. Understanding these limitations (Section 2.4) is therefore essential for designing effective gaze selection techniques (Chapter 3).

2.2 Human Eye Movement

Understanding human eye movement is essential as all eye gaze interaction techniques rely on data which is based on eye movements. A brief description of the physiology of the human eye is provided, as this information is important for understanding the accuracy limitations of eye tracking technology.

Light enters the human eye though the pupil (Figure 2.1). The crystalline lens focuses this light on the retina, an area of the eye containing light receptors. There are two types of receptors, namely rods and cones (Yarbus 1967). Rods exhibit a much higher degree of sensitivity to light, whereas the cones are responsible for colour vision. Cones also enable high acuity visual perception (Zhu and Yang 2002). These receptors are not evenly distributed (Jacob 1995). The densest concentration of receptors is in an area called the fovea, which covers approximately one degree of visual angle. The fovea is located close to the centre of the retina. This area provides greater visual acuity than any other region of the retina. When the eye gazes in a particular direction, the eyeball is rotated so that the light rays fall on this region (Zhu and Yang 2002). The usable accuracy limit of eye tracking equipment is limited by the size of the
fovea, as the user can focus on a target anywhere within this part of the retina (Jacob 1995). This area is roughly equivalent to a target as wide as a thumb at arms length (Sibert and Jacob 2000).

Eye movements which are significant in human computer interaction include fixations, jitter and saccades (Jacob 1991, 1995). Other types of eye movement, such as smooth pursuit and nystagmus, exist but are not considered relevant to gaze interaction.

A fixation is a period in which there is comparatively little eye movement. It is during a fixation that the target object is seen. A fixation typically lasts between 200 and 600 milliseconds. During this period, the eye is not completely stationary, but jitters. Small eye movements which occur during a fixation include drifting, very small

---

Figure 2.1: The Structure of the Human Eye (Ohno et al. 2002)
saccades (microsaccades), and high frequency tremors. These movements are needed for perception, as objects which are fixed relative to the retina are not perceived correctly (Yarbus 1967).

A **saccade** is a sudden movement of the eye which serves to rotate the eye to focus on a target of interest. Saccades are characterised by rapid acceleration and deceleration. The duration of a saccade is typically between 30 and 120 milliseconds. During this period visual processing is limited. A saccade is followed by either another saccade, or a fixation. The speed of saccades enables the eye to spend approximately 95% of its time fixating (Yarbus 1967). The main purpose of saccades is to shift the gaze to new fixation points so that light from points of interest falls on the fovea.

**Smooth pursuit** is eye movement which occurs when the eye tracks a moving object. The eye moves smoothly and slowly (compared to a saccade). Smooth pursuit can only take place if there is a moving object present. This type of eye movement is involuntary (Yarbus 1967). Smooth pursuit movements adjust for when the retinal image leaves the centre of the fovea (Lencer and Trillenberg 2008).

**Nystagmus** is a sawtooth type movement caused by the tracking of objects while the head is moving. Once the target being tracked moves out of view the eyes quickly move to the next target.

Eye movement is tracked using specialised eye tracking equipment. Research into improved eye tracking systems is ongoing.

### 2.3 Eye Tracking Technology

Eye tracking equipment is used to determine the direction in which the subject is looking by calculating a gaze direction vector (Figure 2.2). The point where this gaze direction vector intersects with the screen is called the gaze point (or gaze position), and is of particular interest for HCI applications.
Chapter 2: Introduction to Eye Tracking

Eye trackers are either monocular\(^1\) (tracks one eye) or binocular\(^2\) (tracks both eyes). Tracking both eyes and taking the average of the two has been found to be more accurate than tracking only one eye, whether it be the left or the right (Cui and Hondzinski 2005). A second advantage of binocular trackers is that if the tracker is temporarily unable to track one of the eyes, the gaze data from the other eye can be used (Tobii 2007).

Eye tracking techniques can be divided into three main categories, namely electrooculography, contact lens / suction cap techniques and video-based techniques.

### 2.3.1 Electrooculography
Electrooculography (EOG) measures electric potential differences using electrodes placed on the skin near the eyes (Duchowski 2003). This eye tracking technique was widely used around the early 1960s, but is less popular today. EOG has been used for eye gaze interaction (Borghetti \textit{et al.} 2007, Patmore and Knapp 1998). One of the advantages claimed for EOG is lower cost (Borghetti \textit{et al.} 2007, Kaufman \textit{et al.}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{gaze_direction_vector.png}
\caption{The Gaze Direction Vector (Ohno and Mukawa 2004)}
\end{figure}

---

1. Munn and Pe\text{\textsuperscript{e}}n \textit{et al.} 2008, Kohlbecher \textit{et al.} 2008

13
An obvious disadvantage of this technique is that electrodes need to be attached to the test subject (Figure 2.3).

Figure 2.3: An EOG Tracker (AG Research 2008)

Figure 2.4: Eye Tracking using a Contact Lens and Wire Coil (MIT 2008)
2.3.2 Contact lens / suction cap techniques
The contact lens technique is accurate, but requires the test subject to wear a large contact lens on one eye to which a reference object, such as a marker or a wire coil (Figure 2.4), is attached to measure eye movements (Duchowski 2003). Disadvantages include the intrusive nature of this approach and the fact that eye position is only measured relative to the head. It is unclear whether this technique has ever been used for eye gaze interaction applications.

2.3.3 Video-based Techniques
Video-based techniques rely on a video of the eye to determine the gaze direction vector. The first step in the process is to detect the eyes. There are two approaches to detecting the eyes, namely, passive and active (Zimet et al. 2003).

Passive systems rely on standard video images without additional illumination (Zhu and Yang 2002). As an example, the gaze of a user can be determined based on the vector between the inside corner of the eye and the centre of the iris (Zhu and Yang 2002). Data gathered during a calibration procedure is used to calculate screen coordinates based on this vector. The edge of the iris is detected based on the contrast between it and the surrounding white area (sclera). An ellipse is fitted to the boundary of the iris to calculate its centre.

Active systems make use of additional illumination of subjects' eyes, typically infrared illumination (Borghetti et al. 2007). These systems rely on the fact that the human eye reflects light back to its source. This results in the pupil appearing as a bright object when viewed along the same axis as the light source. An image taken along this axis is referred to as a bright image, whereas one taken from an off-axis angle is referred to as a dark image (Ji and Zhu 2002). These images form the basis of a popular approach to detecting pupils. In a bright image, the pupils appear bright, contrasting with the surrounding iris (Figure 2.5).
In a dark image, the pupil appears darker than the area around it (Figure 2.6). People with blue eyes tend to have a strong “bright-eye response”, whereas those with dark eyes tend to have a strong “dark-eye response” (Tobii 2007). Some systems only use the dark pupil method to detect pupils (Vidaurre et al. 2006, Nilsson et al. 2007, Droege et al. 2008).

Other active illumination video systems, such as the Tobii T60 and T120 (Tobii 2008b) support both light and dark pupil tracking. This technique requires both an on-axis and an off-axis infrared (IR) light source. One way of using both the bright and dark images is to obtain a difference image. This image is obtained by subtracting the dark image from the bright image (Zimet et al. 2003, Hennessey et al. 2006, Hansen...
Chapter 2: Introduction to Eye Tracking

and Hammoud 2007, Chen et al. 2008). This operation helps to filter out spurious bright objects, resulting in an image clearly showing the pupils.

Once the pupil has been located, its centre needs to be determined. The centre is usually calculated by fitting an ellipse around the image of the pupil. The ellipse fitting method is widely used (Ohno et al. 2002, Li et al. 2006, Böhme et al. 2008, Nagamatsu et al. 2008).

The glint, which results from light reflecting off the cornea, also needs to be located (Figure 2.5, Figure 2.6). This reflection is referred to as the purkinje image, which can be used, together with the pupil image, to calculate the gaze direction vector (Ohno and Mukawa 2004, Morimoto et al. 2002). The gaze direction vector is calculated by subtracting the centre of curvature of the cornea vector from the centre of the pupil centre vector. The centre of curvature of the cornea and the centre of the pupil are calculated from the purkinje and pupil images respectively, and a model of the eye. The resulting gaze direction vector intersects both the centre of the pupil, and the centre of the corneal curvature (Figure 2.2).

The electrooculography and contact lens tracking techniques are more intrusive than the video-based techniques as equipment has to be attached to the users. Intrusive eye tracking is just one of the limitations of eye tracking technology.

2.4 Issues and Limitations

Some of the limitations associated with eye tracking technology are due to the nature of the human eye and the characteristics of eye movement. Other issues relate to the eye tracking equipment itself. Issues include the intrusiveness of eye tracking, sensor lag, cost, limited tracking robustness and tracking inaccuracy.

2.4.1 Intrusiveness of Eye Tracking

Recent eye tracking research has focused on making eye tracking less intrusive (Zhu and Yang 2002, Pérez et al. 2003, Ohno and Mukawa 2004, Ronsse 2007). Three requirements for a non-intrusive eye tracker are that the equipment should be simple to set up, devices should not be attached to the user's head, and head movement should be unrestricted (Ohno and Mukawa 2004).
Chapter 2: Introduction to Eye Tracking

An important challenge in simplifying the setup of equipment is the fact that calibration of every user is required for most eye tracking systems. A calibration procedure has to be followed before a user can be tracked reliably. This procedure introduces an additional step not required for other common input devices such as the mouse. Calibration is needed to compensate for measurement errors and individual eye difference. The two main reasons for calibration are variations in the radius of the eye between different users, and difficulty in determining the location of the fovea (Ohno et al. 2002). The visual axis is the actual line of sight, which passes through the fovea. It differs from the estimated gaze direction vector (Figure 2.1). The calibration data is used to compensate for this difference. Calibration procedures require the user to look at specific points on the screen (calibration markers), in order to obtain a set of reference points (Figure 2.7). The user's reported gaze position points differ from the locations of the calibration markers. This information is used to adjust the tracking algorithms in order to track the user more accurately.

In order to simplify the setup of the tracking equipment, the calibration process should not be complex. Systems with simpler calibration procedures which require users to look at fewer calibration markers have been developed (Ohno and Mukawa 2004). The need for calibration has been removed from some systems, simplifying the setup procedure. The accuracy of earlier calibration-free systems was, however, limited (2.5 degrees for Morimoto et al. 2002, five degrees for Ji and Zhu 2002). More recently an accuracy of one degree has been achieved with an auto-calibration tracker (Yun et al. 2008). An eye tracker with a single point calibration procedure has also been developed (Nagamatsu et al. 2008) with a claimed accuracy of less than one degree. Some eye trackers, such as the Tobii T60 and T120 (Tobii 2008b) only require a user to be calibrated once. This calibration can be reused every time the same user uses the system with minimal drift effect (Section 2.4.5).
In order to minimise intrusiveness devices should not be attached to the user's head. Systems which require the user to attach a device to their head are more intrusive (Figure 2.8). Common examples of such systems are those which require the user to wear goggles (Pelz et al. 2000, Selker et al. 2001, Li et al. 2006, Lin et al. 2007).

![Figure 2.7: An Eye Tracker Calibration Screen (Ohno and Mukawa 2004)](image)

![Figure 2.8: An Example of a Head-Mounted Eye Tracker (Yun et al. 2008)](image)
The majority of current systems rely on video-based tracking which eliminates the need for head attachments\(^1\). One example of a system which does not require devices to be attached to the user's head is the Tobii T60 eye tracker, which incorporates a tracker into a 17” LCD display (Figure 2.9).

**Restrictions on head movement** can make eye tracking equipment more intrusive. Some systems require users to keep their head stationary by means of a chin rest (Figure 2.10). Eliminating this restriction ensures that the tracking is less intrusive, but complicates the calculation of the user's gaze point. Head movement introduces an additional level of complexity into this calculation. Research in this area is ongoing\(^2\). Recent systems provide support for a reasonable degree of head movement (SMI 2008, Tobii 2008a).

2.4.2 Sensor Lag
There is a delay between the user looking at a new location, and the tracker reporting the new position, referred to as sensor lag. Lag times of between 5-33 milliseconds, depending on tracking frequency, are claimed for current equipment (Ashmore et al. 2005). This lag has a negative impact on the performance of gaze selection techniques (Kumar et al. 2008).

2.4.3 Cost
Eye tracking equipment is currently expensive, well beyond what the average computer user can afford (Tobii 2008a). This is considered a temporary obstacle (Sibert and Jacob 2000), and research into low cost tracking equipment is ongoing (Hansen et al. 2004, Li et al. 2006, Borghetti et al. 2007, Yun et al. 2008).

2.4.4 Limited Tracking Robustness
One limitation of eye trackers is that they do not work for all users. The limitations depend on the type and model of eye tracker. Issues may arise for users with dry eyes, hard contact lenses, glasses with light-sensitive tinting, or corneas which do not reflect well (Sibert and Jacob 2000). Newer tracking equipment is becoming increasingly robust (Tobii 2007, SMI 2008).
2.4.5 Tracking Inaccuracy

The accuracy of eye tracking equipment is limited because of the properties of the human eye, calibration drift and noisy tracking data. Smoothing algorithms are used to compensate for noisy tracking data.

The human eye imposes an upper limit on tracking accuracy (Jacob 1995). This is due to the size of the fovea, which spans approximately one degree of visual angle. A human looks at a target by focusing light from the target anywhere on this portion of the retina (Section 2.2). When looking at an object, the light could be focused anywhere within this region. The implication of this characteristic is that no matter how advanced eye tracking equipment may be, there will always be an accuracy ceiling of one degree (half a degree either side of the centre of the fovea). This limit has significant implications for gaze selection techniques (Chapter 3).

Another issue relating to accuracy is calibration drift. As time passes since a calibration was performed, the reported gaze point can begin to drift away from the actual gaze point (Duchowski 2003). This issue is not as significant as it used to be. Trackers such as the Tobii T60 and T120 are claimed to exhibit less than 0.3 degrees of drift due to improved drift compensation (Tobii 2008b).

Tracking data is noisy, which also affects tracking accuracy. Whenever a person blinks, the tracker will report invalid gaze point data. Even when users have their eyes open, the data is noisy due to fixation jitter (Sibert and Jacob 2000). A gaze smoothing algorithm can be used to reduce the effects of fixation jitter (Lankford 2000, Kumar 2007, Graupner et al. 2008) and can improve tracking accuracy (Zhang et al. 2008). Smoothing typically averages the raw gaze point data in order to obtain a clearer indication of where the user is actually looking (Figure 2.11). Raw gaze point data tends to have a more jagged appearance when plotted alongside smoothed data.

The use of smoothing algorithms for eye gaze interaction is quite common\(^1\). The complexity and extent to which the algorithms used are documented varies widely. The simplest approaches average the last N gaze points, where typically \(5 \leq N \leq 7\).

---

Chapter 2: Introduction to Eye Tracking

(Hornof and Cavender 2005). Other techniques incorporate saccade and/or fixation detection and/or more advanced filtering techniques (Kumar 2007, Kumar et al. 2008, Graupner et al. 2008, Wobbrock et al. 2008). Fixation detection algorithms were originally developed for off-line analysis of data (Salvucci and Goldberg 2000), but have since been adapted to form part of on-line smoothing algorithms.

The smoothing algorithm of Kumar (Kumar 2007, Kumar et al. 2008) is one of the best documented algorithms, and as such, is used here as an illustrative example. A weighted mean is used to smooth out fixation jitter. The smoothing effect can clearly be seen by comparing the blue (raw) and orange (smoothed) lines (Figure 2.11). A sliding window of 400 milliseconds is used to limit the duration of a fixation. This window ensures that the algorithm remains responsive to small shifts. A velocity-based algorithm is used to detect saccades based on the distance between the current fixation and the most recent gaze point. Once a saccade is detected the current fixation (a set of points) is cleared, and only new data is considered. In order to filter out false saccades, two successive points need to fall further than the threshold distance (the default threshold is 40 pixels) from the current fixation before the algorithm considers these points to be the start of a saccade. This process reduces the risk of noise being confused with a saccade.

Figure 2.11: The Effect of Saccade Detection and Fixation Smoothing
(adapted from Kumar et al. 2008)
2.5 Conclusions

An understanding of current eye tracking technology and its limitations is essential for designing workable gaze selection techniques. The design of existing gaze selection techniques (Chapter 3), is based on the limitations and characteristics of eye tracking and human eye movement. The different types of human eye movement which are relevant to gaze interaction are fixations, saccades, and involuntary jitter movements.

The limited accuracy of eye tracking equipment is the most important constraint associated with the technology. An accuracy limit of 0.5 to one degree of visual angle represents a significant obstacle for gaze selection techniques. This limit is imposed on eye tracking equipment by the size of the fovea. Gaze selection techniques need to be developed which can compensate for this accuracy limitation.

Additional eye tracking issues and limitations include tracking robustness issues, further accuracy issues due to drift and fixation jitter, sensor lag, and the intrusiveness of some eye tracking systems.

A trend towards non-intrusive eye tracking equipment (typically video-based remote units) and simpler calibration procedures, has emerged. Over-simplifying or removing the tracker calibration procedures may make the system less intrusive, but can impact negatively on accuracy. Similarly, permitting head movement is more convenient for the user, but complicates the design of the tracker.

Gaze point data from eye trackers is often smoothed for eye gaze interaction applications to reduce the effects of fixation jitter in order to obtain more accurate tracking data.

In the next chapter gaze selection techniques are explored. Gaze selection techniques, which represent a subclass of eye gaze interaction techniques, are the focus of this study. Eye gaze interaction refers to any interaction in which human eye movement is used as a form of input. As gaze selection techniques rely on gaze point data provided by eye tracking equipment, it is important to bear in mind the limitations and characteristics of this modality.
Chapter 3: Gaze Selection Techniques

3.1 Introduction

Developing accurate gaze selection techniques in the face of noisy, inaccurate gaze point data is the primary challenge facing eye gaze interaction researchers today. Limited accuracy leads to input selection errors which have a negative impact on the user experience. The importance of improving selection accuracy is clearly demonstrated by the designs of existing gaze selection techniques. Elements aimed at improving the accuracy of selections are central to these designs. The designs of existing techniques are also based on the other weaknesses and strengths of gaze as an input modality (Section 3.2).

The purpose of this chapter is to discuss the characteristics of existing gaze selection techniques and how these characteristics affect usability. This review serves to inform the design of new gaze selection techniques (Chapter 4).

One of the weaknesses of gaze input is the lack of an obvious means of triggering a selection. Various triggering options such as dwell time triggering and hardware button triggering are attempts to address this limitation (Section 3.3). Selection triggers are typically used for selecting items on a display in conjunction with gaze.

Gaze interfaces can be classified as either gaze only (Section 3.4.1) or gaze added (Section 3.4.2). A gaze only interface is one which only supports gaze input. A gaze added interface is one in which additional input modalities are supported. One of the best known applications of gaze interaction is eye typing for disabled users (Section 3.4.1.1). Eye typing interfaces are examples of a gaze only interfaces. As eye typing involves the selection of discrete targets it can be considered a specialised application of gaze selection.
Accuracy limitations result in ambiguities when selecting targets. If target items are smaller than the resolution of the eye tracker, it is difficult to determine which item the user intends to select. In order to determine the intended target item, a selection disambiguation mechanism is required. The two most common selection disambiguation mechanisms are target expansion and magnification (Section 3.5). A particularly important gaze selection technique which uses magnification is EyePoint. This technique is discussed in detail since it is used as a basis for a comparative evaluation (Chapters 6 and 7). Intelligent gaze interpretation is another alternative for addressing the problem of ambiguity (Section 3.6).

Another aspect of gaze selection techniques is feedback. The role of feedback in gaze selection techniques is examined in Section 3.7. This discussion is particularly relevant to the design of the novel gaze selection techniques (Chapter 4).

3.2 Strengths and Weaknesses of Eye Gaze as an Input Modality

The design of any gaze selection technique needs to be based on an understanding of the strengths and weaknesses of eye gaze as an input modality. Existing designs are influenced by the constraints imposed by gaze input.

3.2.1 Strengths

Looking at objects is a natural human activity. It follows logically that users do not require training as they already know how to move their eyes (Jacob 1993). Eye movements result in very little physical fatigue (Bates and Istance 2002). The use of gaze reduces the risk of repetitive stress injuries as reliance on the mouse is decreased (Zhai et al. 1999). A further advantage is that gaze provides a better indication of the user’s focus of attention than other techniques (Jacob 1993, Starker and Bolt 1990).

A natural advantage of gaze input is that the eye can be moved very rapidly (Zhai et al. 1999). With mouse input users usually look at objects before selecting them with the mouse. This results in users having to point at the target with their eyes, and then move the mouse to the target area, representing a duplication of effort. Eye gaze
therefore has the potential to be a very efficient mode of interaction. Perceived efficiency is another benefit (Jacob and Karn 2003).

One particularly useful characteristic of eye gaze is that it frees the user's hands for other tasks (Sibert and Jacob 2000, Zhai et al. 1999). This characteristic provides the basis for what is currently the primary application of gaze as an input modality, namely providing accessibility for disabled users (Pomplun et al. 2006).

3.2.2 Weaknesses

The primary weakness of eye gaze as an input modality is the inaccuracy of eye tracking data (Section 2.4.5). This limitation has the greatest influence on the design of gaze selection techniques. Inaccurate gaze data results in target selection errors.

Another important issue is the “Midas touch problem” (Jacob 1990). This problem occurs if users can trigger actions simply by looking at controls on the screen (for example, command buttons). If no suitable dwell time threshold or other selection triggering mechanism is employed, users can easily begin to trigger unintentional actions. This problem stems from the fact that, unlike the mouse, there is no button to confirm a selection (Jacob 1993). Designers have to adopt an alternative selection trigger (Section 3.3).

Although efficiency is a strength, gaze input can also be less efficient than the mouse (Pomplun et al. 2001, Kumar et al. 2007). The relative efficiency of gaze compared to the mouse varies due to factors such as task type and complexity, target size and differences between interaction techniques.

3.3 Selection Triggers

The gaze selection techniques described in this chapter rely on one or more selection trigger mechanisms to counter the Midas touch problem. A taxonomy of current selection triggers includes blinking, dwell time, screen button and hardware button, amongst others (Figure 3.1).
3.3.1 Blinking
With blink-based selection, the user looks at a target object and blinks to select it. This method has not proven to be very popular. Deliberate blinks were rejected as being unnatural (Jacob 1990). Humans blink involuntarily every few seconds which would result in unintentional selections. One option which was proposed, and rejected, by Lankford (2000), was to use a prolonged blink to differentiate between deliberate and involuntary blinks. Ji and Zhu (2002) developed a system which uses three deliberate blinks to indicate a selection, but this is unnatural.

3.3.2 Dwell Time
A common approach to gaze selection, is to use fixations for selection\(^1\). A selection occurs when an object is fixated upon for a period of time exceeding a specified threshold. If the dwell time threshold is set too low, the user may unintentionally activate commands. This is referred to as the *Midas touch problem* (Section 3.2.2). A

---

longer threshold time reduces the risk of this occurring, but results in a less responsive system. Researchers differ on what is considered an optimal dwell time threshold. Examples include 120 milliseconds (Pomplun et al. 2001), 150 to 250 milliseconds (Jacob 1990), 400 milliseconds (Ware and Mikaelian 1987) and 750 to 1250 milliseconds (Miniotas et al. 2004). The use of the above-mentioned threshold range of 150 to 250 milliseconds can result in a very responsive system (Jacob 1990). Short dwell times were used to improve efficiency in cases where an incorrect selection could easily be corrected. The effect of short dwell time thresholds on the efficiency of gaze selection has proven to be statistically significant (Miniotas et al. 2004).

3.3.3 Screen Button
A seldom used gaze selection trigger is the screen button. With the screen button, users perform a selection by looking at the object they wish to select, and then fixating on a region of the screen depicting a select button. The last object which was looked at (before looking at the select button) is then selected. This technique is considered inferior to dwell time and hardware button selection (Ware and Mikaelian 1987).

3.3.4 Hardware Button
A popular selection trigger, which can be used to counter the Midas touch problem, is the hardware button$^1$. The user presses a button while looking at the target object. Various hardware buttons have been used as selection keys. Examples include the control key (Salvucci and Anderson 2000) and the space bar (Fono and Vertegaal 2005, Kumar et al. 2007). A hardware button can also be used as a two-stage triggering mechanism. A “look-press-look-release” technique was used by Kumar et al. (2007). The user presses the button to view a magnified view and releases it to make a selection within that view. The use of a hardware button is preferred over dwell time selection for cases where an incorrect selection is difficult to correct (Jacob 1990).

---

3.3.5 Other Modalities

Other modalities, such as speech or joystick input, can also be used in conjunction with gaze to facilitate object selection (Bolt 1981). Gaze has been used in conjunction with speech for moving objects (Kaur et al. 2003). The user looks at the target object, says “move it”, looks at the target location, and says “there”. Gaze has also been used in combination with speech to select items (Miniotas et al. 2006). When the user fixates on a region of the screen, the buttons in that area are highlighted in different colours. The colours serve as a selection disambiguation mechanism. In order to select an item, the user says the name of the colour associated with that item out loud. Frowning has been used as a selection trigger for disabled users (Surakka et al. 2004, Mateo et al. 2008). Modalities such as speech and frowning are beyond the scope of this research (Chapter 1).

3.3.6 Comparison of Selection Triggers

The main advantages and disadvantages of the different selection trigger types are summarised in Table 3.1. In this section the relative merits of the different trigger types are discussed.

The two most widely used selection triggers are dwell time and hardware button triggers. An obvious disadvantage of a hardware button trigger is that it requires the use of the hands. The relevance of this issue is, however, generally limited to applications involving users with disabilities. The dwell time trigger, the on-screen button and blinking all free up the user's hands. Other modalities, for example frowning or speech, also do not require the use of hands. An important advantage of a hardware button trigger is that it provides a solution to the Midas touch problem. A dwell time trigger only represents a partial solution as a user may still accidentally trigger a selection by inadvertently looking at a control for too long.

The efficiency and accuracy of dwell time and hardware button triggers have also been compared. Ware and Mikaelian (1987) compared these trigger types for targets of various sizes. A dwell time threshold of 400 milliseconds was used. Overall, the hardware button approach was found to be the more efficient method. This was true
for all of the target sizes tested. Dwell time selection was, however, found to produce fewer errors than the hardware button trigger.

<table>
<thead>
<tr>
<th>Selection Trigger</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dwell Time</td>
<td>No hands required.</td>
<td>Can be slower (depending on dwell time threshold)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Only a partial solution to the Midas touch problem</td>
</tr>
<tr>
<td>Hardware Button</td>
<td>Efficient. A solution to the Midas touch problem.</td>
<td>More errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requires hands</td>
</tr>
<tr>
<td>Screen Button</td>
<td>No hands required</td>
<td>Slow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requires more screen space</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User has to look away from the intended target</td>
</tr>
<tr>
<td>Blinking</td>
<td>No hands required</td>
<td>Unnatural</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Need to differentiate between intentional and unintentional blinks</td>
</tr>
<tr>
<td>Other Modalities</td>
<td>Varies according to modality</td>
<td>Varies according to modality</td>
</tr>
</tbody>
</table>

Table 3.1: Advantages and Disadvantages of Different Selection Triggers

The higher error rate of the hardware button was attributed to users attempting to time button presses to coincide with the anticipated arrival of their gaze on a target. The majority of selection errors, when using a hardware button, are due to users shifting their gaze from a target shortly before pressing the selection button (Salvucci and Anderson 2000). This issue was explored further by Xiao et al. (2005), who confirmed the existence of a so-called eye-hand span. Users also often look ahead of where they think they are looking, resulting in input errors. These types of errors can be classified into two categories: early trigger errors and late trigger errors (Kumar et al. 2008). An early trigger error occurs when a user presses the trigger key before looking at the intended target. Similarly, late trigger errors occur when users press the trigger key
Chapter 3: Gaze Selection Techniques

*after* their gaze has shifted away from the intended target item. Early trigger errors are the more common of the two error types (Kumar *et al.* 2008).

The efficiency of dwell time selection has been compared to selection using the mouse. Dwell time selection (with a 150 millisecond threshold) was found to be more efficient than mouse selection (Sibert and Jacob 2000). The mean selection time with the mouse was 931.9 milliseconds, compared to 503.7 milliseconds for dwell time selection.

A hardware button trigger was favoured over *automatic* gaze activation for window activation by Fono and Vertegaal (2005), despite higher levels of user satisfaction and efficiency with the automatic technique. The hardware trigger was favoured by these researchers due to the Midas touch problem causing difficulties with the automatic technique. It should be noted that the *automatic* technique employed an *immediate* dwell time trigger, implying a very small dwell time threshold (if any at all).

The use of blinks to trigger selections is unnatural (Section 3.3.1). It is also difficult to distinguish unintentional from intentional blinks. Blinking is consequently seldom used as a selection trigger.

### 3.4 Types of Gaze Interfaces

Some systems are designed solely for eye gaze interaction (*gaze only interfaces*). Eye typing systems (Section 3.4.1.1) are examples of gaze only interfaces (Majaranta *et al.*, 2004, Hansen *et al.* 2008).

Not all eye gaze interfaces are designed to completely replace interaction using the mouse. Eye gaze interaction can be used together with traditional pointing devices. Users are free to switch between traditional modalities and gaze input as it suits them. These types of systems are referred to as *gaze added interfaces* (Salvucci and Anderson 2000).
Chapter 3: Gaze Selection Techniques

3.4.1 Gaze Only Interfaces

Gaze only interfaces are quite common, particularly for systems designed for disabled users\(^1\). Eye typing (Section 3.4.1.1), which involves users entering text using their eyes, is a common application of gaze only interfaces. This discussion is not intended to be an exhaustive survey of eye typing systems, but rather a brief discussion focusing on common approaches to implementing these types of systems. The use of pEyes (Section 3.4.1.2) for interaction is another example of a gaze only approach to interaction.

3.4.1.1 Eye Typing

Eye gaze is often used by users with disabilities to enter text (Istance et al. 1996, Majaranta and Räihä 2002, Majaranta et al. 2003, 2004). The most common method is to enter text using an on-screen keyboard (Figure 3.2). Keys are enlarged statically in both motor and visual space\(^2\). Users type by looking at the keys and triggering them with a dwell time trigger. Some eye typing systems provide feedback (visual and/or auditory) during typing (Section 3.7).

![Figure 3.2: An On-Screen Keyboard Eye Typing System (Majaranta et al. 2004)](image)

---

2 These concepts are defined in Section 3.5.1
One example of an alternative approach to eye typing, is a system developed by Isokoski (2000). Users look at targets placed around the frame of the monitor in a specific order to enter text. This approach is similar to techniques which use eye gestures on the screen to input text (Wobbrock et al. 2008, Porta and Turina 2008).

Despite efforts to improve the efficiency of eye typing systems using predictive text input (Hansen et al. 2003, MacKenzie and Zhang 2008) and a reduced dwell time (Majaranta et al. 2004), this type of interaction is inefficient compared with the keyboard. Typing speeds reported for eye typing systems typically average around ten words per minute or less. The efficiency trade-off is viewed as acceptable as these systems are not intended for use by able bodied users. The efficiency gap has been narrowed considerably with the introduction of a system called Dasher (Ward and MacKay 2002). This system uses gaze for directional input together with a probabilistic model to select characters which float from right to left across the screen. The typing speed of this system has been measured to be as high as 23 words per minute in independent testing (Tuisku et al. 2008).

Another technique used for inputting text for disabled users is scanning. With scanning systems a highlight is used to indicate the current item (Ten Kate et al. 1979, Shein 1997, Nisbet and Poon 1998). After a period has passed, the highlight automatically moves to the next item. The process continues until the correct item is highlighted and the user makes a selection using a trigger of some sort. Scanning has been used in a very limited manner together with eye movements. Ten Kate et al. (1979) described two systems which use scanning together with eye movements as triggers to select characters to type (a trigger/switch consists of the user looking to the left or to the right). These simple eye movements were used as a basis for communication as the intended users suffered from severe disabilities. Scanning systems usually place items which are more likely to be used often near the starting point of the scanning highlight. The purpose of this arrangement is to minimise the time spent waiting for the highlight to fall on the correct item.
3.4.1.2 pEyes
One technique which uses only gaze is pEyes (Huckauf and Urbina 2008). Hierarchical pie-shaped menus are used for tasks such as eye typing (Figure 3.3) and general command menus. A slice can be selected using a dwell time threshold of 700 milliseconds. This results in a sub-menu (another smaller pie) appearing. A selection can be accelerated by looking at the outer edge of a pie slice, resulting in a selection being made before the dwell time threshold has been exceeded. Slices can be used to represent commands, enabling users to issue commands by looking at a particular pie slice. Applications which have been used to evaluate pEyes include an eye typing system, pEYEwrite (Figure 3.3) and a simplistic desktop simulation, pEYEtop, containing files, folders and links to applications. pEyes were reported to be well liked by the users who found this approach to be easy and fast. User feedback was collected using a questionnaire.

3.4.2 Gaze Added Interfaces
In contrast to gaze only interfaces, users are free to switch between traditional modalities and gaze input with gaze added interfaces. A gaze added graphical user
Chapter 3: Gaze Selection Techniques

interface is discussed in Section 3.4.2.1, followed by a discussion of MAGIC pointing (Section 3.4.2.2).

3.4.2.1 A Gaze Added Graphical User Interface
An example of a basic gaze added graphical user interface, is a system developed by Salvucci and Anderson (2000). As an example of a gaze added interface, one of the distinguishing characteristics of this system is that users are free to switch back and forth between manual (mouse) and gaze interaction. Users interact with the system by gazing at objects and pressing the control key on the keyboard. These interaction techniques are based on the mouse interaction techniques also supported by the system. Button presses are equivalent to mouse clicks. Double-clicking is also supported by pressing the button twice. A circle is used to indicate the user's current gaze point on the screen. The object at which the user is currently looking is highlighted with a yellow background. Large target sizes are used regardless of whether the user is using gaze or mouse interaction.

In an evaluation to determine an indication of user preference between gaze and mouse interaction it was found that 80% of test participants used gaze more than half of the time (Salvucci and Anderson 2000). There was a clear preference for the gaze modality for basic single-click type operations. Gaze was less popular for more complex operations such as drag and drop.

3.4.2.2 MAGIC Pointing
A further example of gaze added interaction is that of Manual and Gaze Input Cascaded (MAGIC) pointing (Zhai et al. 1999). Eye gaze is used in conjunction with a notepad pointing stick. The cursor is moved to an approximate location using gaze. A notepad pointing stick is then used to manipulate the pointer more accurately. The mouse is also used to select the target item, eliminating the need for a dwell time threshold or keyboard button. A mouse was found to be inappropriate as an input device for MAGIC pointing, as it would often be pushed to the edge of the mouse pad, requiring the user to lift it. This occurred if the mouse manipulation phase required the user to repeatedly make corrections in the same direction. One advantage of MAGIC
pointing is that, due to the use of a traditional pointing device, the accuracy of the eye tracking equipment is not a major concern.

Two types of MAGIC pointing were explored. In the *liberal* approach, the cursor is automatically moved to wherever the user focused. In the *conservative* approach, the cursor is only moved to where the user focused once the traditional pointing device was moved.

The performance of this technique was evaluated experimentally (Zhai *et al.* 1999). The two MAGIC pointing techniques (liberal and conservative) were compared with each other and with a traditional pointing device. Liberal MAGIC pointing was found to be the fastest technique, followed by conservative MAGIC pointing. Manual selection using the pointing stick was slowest. The accuracy of both MAGIC pointing techniques was equivalent to the accuracy of selection using the mouse.

Some interesting observations made concern users' perceptions. The MAGIC pointing techniques were perceived by the users to be more efficient, even by users whose performance was worse compared to manual pointing. Overall the users' reaction to MAGIC pointing was reported to be positive. Reasons cited included the novelty of this type of interaction, and reduced fatigue and physical effort.

A similar technique was designed specifically for clicking on buttons (Yamato *et al.* 2000). Three technique variations were implemented, namely the *combination* technique, the *automatic adjustment* technique and the *manual adjustment* technique. With the combination technique, the user looks at a button on the screen which they intend to select, and then clicks one of the mouse buttons. This is effectively an implementation of the hardware button approach (Section 3.3.4). In a pilot study this technique was found to be ineffective for small buttons commonly found in desktop applications. Two improved techniques were developed. With the automatic adjustment technique, the on-screen button nearest to the user's gaze was pressed when the user clicked the mouse button. The other adjustment technique implemented, manual adjustment, works like liberal MAGIC pointing. Eye gaze is used to move the cursor to the vicinity of a target button. The mouse is then used to make small cursor
adjustments before the user clicks on the target. Unlike MAGIC pointing, the mouse was used instead of a pointing stick.

### 3.4.3 Comparison of Gaze Only and Gaze Added Interfaces

A comparison of the strengths and weaknesses of gaze added and gaze only interfaces appears in Table 3.2.

MAGIC pointing, an example of a gaze added approach, demonstrated that gaze can be combined with a traditional input modality to improve efficiency, providing greater accuracy than other gaze selection techniques. This technique does, however, require the use of a traditional pointing device, which generally implies that users need to use their hands. This technique is consequently unsuitable for hands-free tasks and disabled users.

The idea of a gaze added interface is appealing as it allows users to freely choose which modality they prefer for each task. If a user is more comfortable using the mouse, gaze interaction need not be used at all. Results presented by Salvucci and Anderson (2000) indicate that users typically do not use gaze (or the mouse) exclusively when provided with a gaze added interface, but rather choose a modality based on the task they wish to perform.

A disadvantage of gaze added interfaces is that this approach can constrain user interface designers, as the user interface has to work with both the mouse and gaze. A user interface designed solely for gaze interaction, such as pEYEwrite (Figure 3.3), is likely to differ significantly from one designed to accommodate traditional manual input. For users who only wish to use the mouse, modifications made to facilitate gaze interaction in a gaze added interface may not suit the mouse, and may detract from the user experience. Disadvantages of a gaze only interface include users not being able to use their existing skills with the mouse, and not being able to choose the most appropriate input modality based on the task.
Chapter 3: Gaze Selection Techniques

<table>
<thead>
<tr>
<th>Gaze Added</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Users can switch between input modalities and select the modality that they most comfortable with for each task.</td>
<td>Must support interaction with the mouse, limiting changes which may suit gaze interaction better.</td>
</tr>
<tr>
<td></td>
<td>Greater efficiency vs the mouse (in the case of MAGIC pointing).</td>
<td>MAGIC pointing does not free the user's hands.</td>
</tr>
<tr>
<td></td>
<td>Greater accuracy vs other gaze techniques (in the case of MAGIC pointing)</td>
<td></td>
</tr>
</tbody>
</table>

| Gaze Only  | More flexibility, the interface is only designed for gaze input and does not need to support mouse input. | Users cannot use their existing skills with the mouse or choose a suitable modality based on the task. |
|            | Best suited to situations where the user is unable to use their hands.                                         |                                                                              |

Table 3.2: Comparison of Gaze Interfaces

3.5 Interaction Using Target Expansion or Magnification

In order to design usable eye gaze interaction techniques, usability researchers have had to develop techniques which are able to cope with inaccurate gaze point data. One approach to counteracting the limited accuracy of eye tracking equipment, is simply to magnify the screen, or a portion thereof (Section 3.5.2). A second technique, target expansion, involves expanding the individual targets (for example buttons), rather than magnifying an area of the screen (Section 3.5.1).

The use of enlarged targets or magnification for eye gaze interaction is usually motivated by Fitts' law (Fitts 1954). Applied to pointing devices, Fitts' law indicates that targets which are larger are faster to select. Targets which are closer are also faster to select. Ashmore et al. (2005) and Kumar et al. (2007) use Fitts' law to motivate the design of their eye gaze interaction techniques. As noted by Ashmore et al. (2005), there is some debate as to whether Fitts' law applies to eye pointing. Ware and Mikaelian (1987) appear to be the first researchers to relate eye gaze to Fitts' law.
3.5.1 Target Expansion

Target expansion can be classified into two types: **motor space expansion** and **visual space expansion** (Figure 3.4).

![Figure 3.4: Target Expansion Techniques](image)

**Motor space** is defined as: “the set of all possible positions of the pointing device [or the physical space that the user’s limb moves through]” (McGuffin and Balakrishnan 2005, p 401).

With **motor space** expansion, the selectable area of a target is increased (Figure 3.5). On the left of Figure 3.5 is a green button. In order to “click” on this button a user has to click within the green area. Once motor space expansion is applied, the button still appears to be the same size (the green area). The difference is that in order to “click” on this button the user can now click anywhere inside the green area or anywhere inside the enlarged blue area which surrounds it. The expansion of the target in motor space is invisible to the user (the blue area is shaded for illustrative purposes). All input events, such as mouse clicks and fixations, occurring in the enlarged area **around** a control are processed as if they occurred **within** the original bounds of that control. As motor expansion does not have to be accompanied by visual expansion (or vice versa), a target may appear smaller on the screen than its size in motor space.
Visual space is defined as: “[the location] where visual feedback is displayed, for example, the set of pixels on a raster display” (McGuffin and Balakrishnan 2005, p 401).

Visual target expansion results in a target which appears larger on the screen. This type of expansion is usually accompanied by motor space expansion, which increases the selectable area of a control to match its enlarged visual size. The distinction between magnification and target expansion may appear blurred, especially if motor space expansion of targets is accompanied by visual expansion. Magnification refers to an enlarged view of an area of the screen, irrespective of what components may be found within that area. Visual expansion refers to the enlargement of one or more components individually.

Expansion can take place statically or dynamically. With static target expansion, targets are permanently enlarged. An alternative technique, dynamic target expansion, expands targets temporarily in response to user interaction.

3.5.1.1 Static Expansion in Motor Space

An interaction technique involving static expansion and an algorithm to counteract fixation jitter, the Grab and Hold Algorithm (GHA), was developed by Miniotas et al. (2004). The static expansion of targets involved motor space expansion, but no corresponding visual expansion. It was argued that static target expansion was more appropriate for eye gaze interaction than dynamic target expansion. It was claimed that users would not notice changes in the size of a target during a saccade, limiting the usefulness of dynamic expansion.
Chapter 3: Gaze Selection Techniques

The accuracy of eye tracking equipment is limited by the properties of the human eye (Section 2.2, Section 2.4.5). The GHA was designed to compensate for this inaccuracy. In a pilot study without the algorithm, it was found that gaze positions reported by eye tracking equipment often fell outside the area where users were looking (Figure 3.6). The solid rectangle is the item the user is actually looking at.

To counteract this inaccuracy, static target expansion in motor space was used together with the GHA. The expansion of the target takes place invisibly in motor space. In Figure 3.6 the visual target has a width of W. It is expanded in motor space by an expansion factor (EF), increasing its width in motor space to (W x EF). The advantage of only expanding the target in motor space, and not visually, is that users would focus on the center of the target, improving accuracy.

The idea behind the GHA is to interpret the gaze points intelligently when detecting fixations. Once a gaze point was detected within the expanded target area, the GHA interprets subsequent gaze points as if they fell within that area even if they did not. Only when a saccade is detected is the fixation considered terminated by the algorithm. Using this technique, users can use dwell time selection to select an object even if their gaze wanders slightly or is reported inaccurately by the tracking equipment.

Figure 3.6: Static Target Expansion (Miniotas et al. 2004)
One disadvantage was that static target expansion in motor space requires the expansion area around each control to be free of other interactive widgets (Miniotas et al. 2004). It was also noted that a more advanced algorithm would be necessary for handling interfaces with multiple targets.

Evaluation results reported by Miniotas et al. (2004) and Miniotas and Špakov (2004) indicate that target expansion resulted in improved accuracy and efficiency, with larger expansion factors resulting in greater benefits. Applying the GHA algorithm improved accuracy at the expense of efficiency.

### 3.5.1.2 Dynamic Expansion of Menu Items

Dynamic target expansion has been used to improve the accuracy of menu item selections (Špakov and Miniotas 2005). At a height of 20 pixels, it was claimed that the average menu item is too small to select accurately (Figure 3.7). The solution proposed was to expand menu items during selection.

![Figure 3.7: Target Expansion during Menu Item Selection (Špakov and Miniotas 2005)](image)

In the first frame (a), the menu appears unchanged. The user intends to select menu item 2 and gazes at it. Due to calibration drift, the eye tracker provides incorrect gaze
point data. This results in the system outlining menu item 3. After a predefined dwell time passes, the outlined menu item (3) is expanded and highlighted. Menu items which are below menu item 3 shift downwards, and those above it shift upwards. The identified menu item is enlarged in motor space relative to the other items. In this example, the user actually intended to select menu item 2, and was gazing at it. After the expansion of menu item 3, the user's gaze needs to shift upwards in order to continue gazing at menu item 2. This gaze shift is recognised by the algorithm as an indication that the wrong menu item was selected.

The algorithm relies on a predetermined threshold for identifying corrective shifts made by the user. If the magnitude of the shift exceeds the threshold value, it is considered significant. Corrective action is taken if a significant shift is detected. This entails returning the expanded item to its original size, as well as expanding and highlighting the item above or below it instead. The new candidate item is determined based on the direction of the gaze shift. In this example the user's gaze shifted upwards, so menu item 2 is expanded. If no further gaze shifts occur, the currently expanded menu item is selected. The process is repeated if additional gaze shifts occur. Each time a significant gaze shift is detected using this procedure, the calibration of the eye tracking equipment is adjusted by the system. This adjustment is intended to improve the accuracy of subsequent selections.

Optimal values for the parameters which govern the operation of the algorithm were determined based on a pilot study. A gaze shift threshold of 15 pixels was found to be optimal for a menu sized target. An expansion factor of 4.5 was also selected. Users took between 220 and 400 milliseconds to respond to an item moving on the screen. Based on this result, a time window of 500 milliseconds was used for detecting gaze shifts. A dwell time of one second was chosen for selection operations.

This technique was found to significantly improve the accuracy of selections. With dynamic expansion enabled, the gaze selection error rate falls from 55.3% to 9.2% for menu sized targets (Špakov and Miniotas 2005). In comparison, an error rate of 3.8% was recorded for mouse selection. Based on these results, dwell time selection of
menu items without any enhancements is clearly not reliable. Even with the expansion technique, the error rate was more than double that of the mouse.

The main weakness of dynamic menu expansion is its inefficiency. This technique was 39% slower than gaze without dynamic expansion, taking over two and a half seconds to select an item. The mouse was significantly faster than either gaze technique with an average selection time of 772 milliseconds.

### 3.5.1.3 Static Visual and Motor Space Expansion

A system combining static visual and motor space expansion was developed by Salvucci and Anderson (2000). All of the widgets (including buttons and icons) were enlarged, both visually and in motor space. Users could select items by looking at them and pressing a trigger key on the keyboard. A version of this system was also developed which incorporates intelligent gaze interpretation (Section 3.6).

### 3.5.2 Magnification

An alternative to target expansion is magnification, which is one of the simplest ways to make targets bigger and easier to select. The main advantage of a magnification approach, is that it can be applied to existing systems as these techniques only rely on screen capture images and mouse event APIs.

Figure 3.8 provides a taxonomy of how existing magnification techniques can be classified. The two main categories of magnification techniques are **basic magnification** and **fisheye lens magnification**.

Most techniques involving magnification rely on basic magnification, namely taking a screen capture image and scaling it. This could be done by magnifying the whole screen (Section 3.5.2.1) or by magnifying a portion of the screen via a pop-up. ERICA (Section 3.5.2.2) and EyePoint (Section 3.5.2.3) are two examples of pop-up magnification. A more complex approach is to use a fisheye lens for magnification (Section 3.5.2.4).
Chapter 3: Gaze Selection Techniques

3.5.2.1 Full Screen Magnification

One approach to magnification is to magnify the entire screen (Bates 1999, Bates and Istance 2002). Bates experimented with a shoulder-mounted spatial tracker to control screen zooming. Users would raise their left shoulder two millimetres in order to magnify the screen by a factor of two. For every additional two millimetres the shoulder is raised, an additional level of zoom is triggered. Thus if the left shoulder is raised by four millimetres, the zoom level would be three times.

Bates (1999) evaluated the shoulder controlled magnification system with targets measuring between four and thirteen millimetres across (selected because common controls in the Windows operating system were claimed to fall within this range). This is equivalent to between 0.4 and 1.2 degrees of visual angle. Items were selected using a dwell time trigger with a threshold of 750 milliseconds. The user was free to choose what level of magnification, if any, to use. Based on the results (Figure 3.9), it is clear that users selected larger zoom factors for smaller targets. An interesting observation made by Bates, was that multiplying the target size by the magnification factor...
selected resulted in a near constant value of 12.38 millimetres. This value is the effective target size once magnification has been applied.

Trials were conducted with (multimodal) and without (monomodal) the magnification control. The magnification control reduced the error rate significantly for the smaller target sizes. This technique reduced the mean number of errors for four millimeter targets by 65%. Although no selection times were provided, it was stated that there were no significant differences between the selection times for the magnified and non-magnified cases.

### 3.5.2.2 ERICA (Pop-Up Magnification)

An example of a screen magnification-based interaction technique is that of Lankford (2000). The ERICA (Eye-gaze Response Interface Computer Aid) system was developed for disabled users and supports pointing operations. This system magnifies a portion of the screen in a pop-up window (Figure 3.10). This pop-up is triggered by using a dwell time trigger.
Within the pop-up, users can trigger an on-screen menu of mouse actions to perform using a dwell time trigger. Once the menu has appeared, users fixate on the item representing the action they wish to perform. Based on this description, it is clear that this system is not very efficient for able bodied users. Lankford mentions an alternative scheme in an earlier system, whereby able bodied users could press a button on the keyboard to trigger a selection, but no mention of this alternative selection trigger is made regarding the magnification system.

### 3.5.2.3 EyePoint (Pop-Up Magnification)

An eye pointing technique named EyePoint was developed and evaluated by Kumar et al. (2007). EyePoint combines a localised magnification pop-up with a hardware button trigger. Users look at the target that they wish to select and press the trigger key.
down (without releasing). This action results in a pop-up window being displayed containing a magnified view of the area where the user was looking (Figure 3.11). In order to select the target, the user looks at it in the magnified view and releases the trigger key. This process is described as a “look-press-look-release action”. Various mouse actions are mapped to number pad keys on the keyboard. The trigger key button to use depends on which action the user wishes to perform.

When a trigger key is pressed, EyePoint captures a square area around where the user is currently looking, with the default capture size extending 60 pixels to the left, right, above and below this point. A default magnification factor of four is used for the magnification pop-up. The purpose of this magnification is to compensate for the inaccuracy of the eye tracking data. In order to counteract fixation jitter a smoothing algorithm was developed (Kumar 2007, Kumar et al. 2008, Section 2.4.5).
The magnification pop-up is semi-transparent (Figure 3.11 and 3.12), enabling users to contextualise the pop-up. Another feature is the grid of focus points painted over the magnified pop-up. The points are designed to help users to focus on a specific location.

Three variations of EyePoint were evaluated against the mouse (Kumar et al. 2007). One variation was EyePoint with the focus points and a gaze point cursor (“gaze marker”) (Figure 3.11). A second variation was EyePoint without the focus points or the gaze point cursor. The third (standard) version of EyePoint included the focus points, but not the gaze point cursor (Figure 3.12).

Figure 3.12: The Magnified EyePoint Popup without Gaze Marker (Kumar et al. 2007)
Chapter 3: Gaze Selection Techniques

The mouse was found to be more efficient than any of the three variations of EyePoint for pointing tasks, but less efficient for a mixed task involving the user switching between the mouse and the keyboard to type. The average time to perform a selection using EyePoint ranged from 1.3 to 1.9 seconds, depending on the task and instructions. The error rate using the mouse varied between 1% and 3%, whereas the error rate for EyePoint ranged between 12% and 24%, depending on the task and the size of the target. No statistically significant differences in accuracy were found between the variations with focus points and those without. A majority of users preferred having focus points. The gaze point cursor was found to decrease efficiency.

When asked to rank the techniques, EyePoint was preferred over the mouse by three quarters of the test participants for the pointing and selection tasks. For a mixed task involving the keyboard, over 90% of test participants preferred EyePoint and felt that it was faster and easier to use. For one of the pointing and selection tasks the mouse was perceived to be faster, for the other there was an even split. The mouse was perceived to be more accurate for all the test scenarios.

The source of errors in EyePoint was analysed and it was found that early trigger errors were more common than late trigger errors (Section 3.3.6). Using simulations on actual test data, it was found that by shifting the trigger point to 80 milliseconds after the user released the trigger button, the number of early trigger errors was reduced (Figure 3.13). The corrective shift was named “early trigger correction” (Kumar et al. 2008). The term trigger point time shifting is used in this manuscript to describe this technique1, in order to facilitate the discussion of an alternative method of early trigger correction.

---

1 The term “early trigger correction” refers to a specific technique, although the name implies a class of techniques.
Chapter 3: Gaze Selection Techniques

3.5.2.4 Fisheye Lens Magnification

Ashmore et al. (2005) used fisheye lens magnification to facilitate object selection. Three different techniques, namely omnipresent, MAGIC and GHA, were developed and evaluated. A fourth option, nonexistent, where no lens was used at all, served as a baseline for comparison. A dwell time threshold trigger was employed for all of the techniques. The GHA fisheye and MAGIC fisheye techniques are so named due to similarities to the MAGIC pointing technique of Zhai et al. (1999) (Section 3.4.2.2), and the grab and hold algorithm (GHA) of Miniotas and Špakov (2004) (Section 3.5.1.1).

In the omnipresent fisheye technique, the lens is displayed at all times, and follows the user's gaze around the screen. With the MAGIC fisheye technique, the lens is only displayed when a fixation is detected. Once the lens is visible, its position is continuously updated based on the user's gaze during the fixation. Similarly, for the GHA fisheye technique the lens is only displayed when a fixation is detected. Unlike the MAGIC fisheye, the location of the lens does not change once it is visible.
Chapter 3: Gaze Selection Techniques

The goal of the fixed lens approach of the GHA fisheye is to counteract the Gutwin effect (Gutwin 2002). This effect occurs when objects viewed through a fisheye lens appear to move in the opposite direction to the movement of the lens, potentially confusing the user.

With the GHA and MAGIC fisheye techniques the lens is *morphed into view*, starting 100 milliseconds into a fixation. After another 400 milliseconds of fixation the lens is fully visible. Figure 3.14 depicts the fisheye lens magnifying test targets. The lens vanishes as soon as a saccade is detected (except in the omnipresent lens technique). This design is intended to enable the user to look around and preview without having to permanently view the screen through the lens. Feedback to the user is provided in the form of a green dot indicating the current position of the user's gaze.

![Figure 3.14: A Fisheye Lens Magnifying at the Point of Fixation (adapted from Ashmore et al. 2005)](image)

Figure 3.14: A Fisheye Lens Magnifying at the Point of Fixation (adapted from Ashmore et al. 2005)
Selection without a fisheye lens was evaluated, along with the omnipresent, MAGIC and GHA fisheye lens techniques. A grid of nine by nine boxes was presented to the user (Figure 3.14). Users were tasked with selecting target boxes marked with an 'X' by fixating on the centre of boxes for 500 milliseconds. The boxes were all permanently expanded in motor space, but not visually. The targets would only appear larger when magnified by the fisheye lens. A green dot was used to indicate the current gaze point.

MAGIC and GHA fisheye selection were the most efficient in terms of time taken to perform a selection (Figure 3.15). The omnipresent fisheye and nonexistent fisheye techniques were less efficient. The lack of a preview for the permanent lens was cited as a reason for the poor performance of the omnipresent lens technique. This canceled out any efficiency benefits derived from the fisheye lens magnification. The selection times also included the time it took the user to locate the target box within the grid. Average selection times ranged from approximately 3.4 to 4.8 seconds.

The mean distance between the fixation points and the target centres was used to determine the margin of error in pixels. The GHA fisheye (15.3 pixels) and
nonexistent (15.26 pixels) modes fared worse than the MAGIC fisheye (14.26 pixels) and omnipresent (14.8 pixels) modes, although the differences appear small. The reason provided for the inferior accuracy of the GHA fisheye mode, was that the lens was fixed at the location where users started their fixation, even if they were slightly off target. Users would then select the target box in the distorted area at the edge of the lens (Figure 3.14).

### 3.5.3 Comparison of Magnification and Target Expansion

Magnification and target expansion can both provide accuracy and efficiency benefits for gaze selection. In some cases efficiency can be traded for greater accuracy (Špakov and Miniotas 2005), and in other cases improvements in both areas are possible (Ashmore et al. 2005). The advantages and disadvantages of magnification and target expansion are summarised in Table 3.3.

<table>
<thead>
<tr>
<th></th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Expansion</strong></td>
<td>Can improve accuracy</td>
<td>Difficult to apply to existing systems</td>
</tr>
<tr>
<td></td>
<td>Can improve efficiency</td>
<td>May require changes to GUI layout</td>
</tr>
<tr>
<td></td>
<td>Motor space-only expansion</td>
<td>May be less efficient</td>
</tr>
<tr>
<td></td>
<td>may draw user's gaze closer to centre of target</td>
<td>(depending on how it is used)</td>
</tr>
<tr>
<td><strong>Magnification</strong></td>
<td>Can improve accuracy</td>
<td>Limited opportunities for feedback unless additional</td>
</tr>
<tr>
<td></td>
<td>Can improve efficiency</td>
<td>information is used</td>
</tr>
<tr>
<td></td>
<td>Simpler to implement</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Easier to apply to existing systems (minimal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>information regarding underlying system</td>
<td></td>
</tr>
<tr>
<td></td>
<td>required to implement)</td>
<td></td>
</tr>
</tbody>
</table>

*Table 3.3: Advantages and Disadvantages of Magnification and Target Expansion*

Magnification is best applied to gaze interfaces when needed, rather than always having a level of magnification present (Ashmore et al. 2005, Section 3.5.2.4). This approach enables the user to preview when searching without magnification. With EyePoint (Section 3.5.2.3), localised magnification is also applied selectively, using a
key triggered pop-up, yielding a practical technique which proved popular with able bodied users. Given the accuracy, efficiency and user satisfaction of this technique it could reasonably be argued that this technique currently represents the state of the art in gaze selection techniques.

One of the disadvantages of target expansion when compared with magnification is that it is more difficult to apply target expansion to an existing system. The only functionality required for magnification techniques is the ability to take a screen capture and to simulate mouse events (Kumar et al. 2007). Extensive user interface modifications would be required to implement some of the existing target expansion techniques. The target expansion technique of Miniotas et al. (2004, Section 3.5.1.1), for example, requires the area around a control to be free of other interactive controls. The dynamic menu expansion technique of Špakov and Miniotas (2005) requires menu items to be dynamically resized and shifted. These modifications have obvious implications for the design and layout of graphical user interfaces.

One of the more interesting arguments in favour of motor space only target expansion was that of Miniotas et al. (2004), who argued that this type of expansion would draw the user's gaze closer to the centre of a target. It is interesting to note that target expansion techniques typically do not use any sort of localised pop-up (as is popular with magnification techniques) for applying target expansion selectively.

3.6 Intelligent Gaze Interfaces

One method of improving the accuracy of gaze selection is to use an intelligent user interface. Maybury (1998, p 2) defines intelligent user interfaces as “human-machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media (e.g., graphics, natural language, gesture)”.

The concept of an intelligent user interface was applied by Salvucci and Anderson (2000), who developed a probabilistic model of user behaviour for intelligently interpreting gaze points. A hardware button trigger was used. Gaze points were mapped onto targets using a probabilistic model which modelled user behaviour for a
specific application. The current gaze location and the current task context were taken as input. A formula was used to select the most likely intended target.

The effect of this model was to expand or contract targets in motor space, based on the current task context. To illustrate this, Figure 3.16 depicts the top right corner of a window with a close button (the grey square). A grid of circles and crosses indicates the effective size of each control in motor space. The shaded red circles represent gaze points which are assigned to the close box using the model. Open circles and crosses are assigned to other targets. The image on the right in the figure is for a default probability for the close button. On the left is an image for a case where a smaller probability was assigned to the same button. The size of the close button in motor space is smaller in the left image (fewer red shaded circles), despite its visual size remaining fixed. This is due to the assignment of a smaller probability value.

The efficiency of this interface was compared to selection using the mouse (Salvucci and Anderson 2000). Mouse interaction was slightly more efficient on average, but the difference between the two modes was not statistically significant. The percentage of correct selections was recorded for each modality. Gaze selection (using intelligent
gaze interpretation) produced more errors than selection using the mouse. The number of incorrect selections increased dramatically without intelligent gaze interpretation. With the initial intelligent gaze system, 83% of tasks were completed correctly on average. Performance without intelligent gaze interpretation was very poor. Only 17% of tasks were completed correctly in this mode.

In effect, intelligent gaze interfaces (as implemented by Salvucci and Anderson 2000), provide an additional layer of target expansion or contraction in motor space. One disadvantage of this approach, is that it requires some assumptions on the part of the interface designer about how users will interact with the system. The model of user behaviour and tasks needs to be accurate, otherwise the intelligent interpretation may prove to be counter-productive. Knowledge of the application domain, and the steps users follow to perform various tasks, is therefore essential for effectively designing this type of system. The operating system interface used to demonstrate the model relies on fairly conservative assumptions about user behaviour, but the accuracy benefits are significant. The advantages and disadvantages of intelligent gaze interfaces are summarised in Table 3.4.

<table>
<thead>
<tr>
<th></th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligent Gaze Interface</td>
<td>Improved selection accuracy</td>
<td>Requires a model of user behaviour for every application.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>May be counter-productive if model assumptions are inaccurate.</td>
</tr>
<tr>
<td>Regular Gaze Interface</td>
<td>Does not require a user model.</td>
<td>Less accurate</td>
</tr>
</tbody>
</table>

Table 3.4: Advantages and Disadvantages of Intelligent Gaze Interfaces

### 3.7 Feedback for Gaze Selection

Feedback during gaze selection can be implemented in two ways. Firstly users can be provided with feedback during a selection as to which target they are about to select. Feedback can also be provided to confirm that a particular target has been selected, or simply to confirm that some target has been selected. In some cases only type one of
feedback is provided. Feedback may only be provided during target selection, but without providing a confirmation after a selection has been performed (or vice versa).

Eye gaze interaction techniques typically do not provide gaze point feedback during a selection operation\(^1\). This is due to the “feedback loop” (Jacob 1993, Lankford 2000) which results from drawing the user's gaze point on the screen. The user attempts to look at this point, which does not coincide perfectly with the actual gaze point (due to tracking inaccuracies), resulting in the gaze point being drawn even further away. Gaze point feedback was also found to have a detrimental effect on performance when applied to EyePoint (Kumar et al. 2007). This type of feedback was also provided by Ashmore, et al. (2005), but users were specifically instructed not to look at this point, but rather to use it to make corrections in their peripheral vision.

An alternative approach to providing feedback is to highlight the item at which the user is currently looking (Majaranta and Räihä 2002). This form of feedback has been explored by a number of researchers\(^2\), and is most commonly found in eye typing systems\(^3\).

A feedback highlight can take a number of forms. One example is to display a rectangle in the top right hand corner of the menu item upon which the user fixated most recently (Ware and Mikaelian 1987). A second approach is to outline the target item and then gradually shrink the text of that item when the user fixates upon it (Figure 3.17).

![Figure 3.17: Animation of Visual Feedback – Shrinking Letter (Majaranta et al. 2003)](image)

The eye typing component of ERICA incorporated a shrinking rectangle around the target item in a similar manner (Lankford 2000). Feedback can be provided once a

---

2 Ware and Mikaelian 1987, Majaranta et al. 2004, MacKenzie and Zhang 2008
selection has been performed (1-level feedback, Figure 3.18) or also when focusing on a target (2-level feedback, Figure 3.18). Visual feedback in the form of a colour coded highlight outline has also been explored (MacKenzie and Zhang 2008). The colour of the outline changes when fixating upon and selecting a button on the on-screen keyboard.

Another type of feedback is auditory feedback. Examples of auditory feedback include reading the text of a button with speech synthesis (Figure 3.18) or playing a “click” sound when it is selected (Majaranta et al. 2003).

<table>
<thead>
<tr>
<th>Feedback mode</th>
<th>While focused</th>
<th>When selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech</td>
<td>none</td>
<td>letter spoken</td>
</tr>
<tr>
<td></td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>1-Level Visual</td>
<td>none</td>
<td>red background</td>
</tr>
<tr>
<td></td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>2-Level Visual</td>
<td>highlight</td>
<td>red background</td>
</tr>
<tr>
<td></td>
<td>h</td>
<td>h</td>
</tr>
</tbody>
</table>

*Figure 3.18: Eye Typing Feedback Options (Majaranta et al. 2004)*

The effects of visual and auditory feedback on eye typing have been explored (Majaranta et al. 2003, 2004). Auditory feedback performed best compared to visual feedback when a 900 ms dwell time was used. When a shorter dwell time (450 ms) was used, visual feedback performed best and was preferred by the majority of users. It was stated that many test participants believed visual feedback to be very important as auditory feedback was not sufficient. Selection without any feedback was not
evaluated. One reason provided for the effect of the type of feedback on performance was that feedback provided confirmation that a letter had been typed correctly. With this confirmation, the user does not need to check the field containing the typed text to see if the letter was typed correctly.

When feedback (visual or auditory) is evaluated in eye typing systems, it is not usually evaluated against a version of the system without any feedback (Majaranta et al. 2003, 2004), making it difficult to draw conclusions regarding the merits of having feedback versus not having feedback.

Outside of the domain of eye typing, the prevailing view appears to be that visual feedback, when linked to eye movements, detracts from the usability of a system. Some researchers have gone so far as to cite this view as their primary design principle: “...our primary design principle of not slaving any visual feedback to eye movements” (Kumar et al. 2007, p 5). The only form of visual feedback evaluated by Kumar was a gaze cursor displaying the user's gaze point at all times. Negative comments from users regarding this type of feedback and inferior performance were cited as evidence supporting this design principle. Kumar et al. (2007) cited a similar view against visual feedback expressed by the creators of MAGIC pointing to support their argument: “In short, to load the visual perception channel with a motor control task seems fundamentally at odds with users’ natural mental model in which the eye searches for and takes in information and the hand produces output that manipulates external objects” (Zhai et al. 1999, p 247). This view was argued based on general observations regarding gaze interaction and related research, rather than as a result of a comparative evaluation.

A summary of the advantages and disadvantages of feedback in eye gaze interfaces is provided in Table 3.5.
Chapter 3: Gaze Selection Techniques

<table>
<thead>
<tr>
<th>Feedback</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feedback</td>
<td>User has an indication of which item they are about to select.</td>
<td>Target highlighting requires information regarding the location and size of targets.</td>
</tr>
<tr>
<td></td>
<td>Provides confirmation of a selection</td>
<td>A gaze cursor can cause a feedback loop.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>May be against users' natural mental model.</td>
</tr>
<tr>
<td>No feedback</td>
<td>Simpler to implement than target highlighting as no information regarding the location and size of targets is required.</td>
<td>User has no feedback indicating which item they are about to select.</td>
</tr>
</tbody>
</table>

Table 3.5: Advantages and Disadvantages of Feedback in an Eye Gaze Interface

3.8 Conclusions

Despite more than two decades of research, a standard approach to eye gaze interaction, and gaze selection in particular, has yet to emerge. Usability researchers continue to develop new gaze interaction and selection techniques. These designs are based on the limitations of eye tracking equipment and the human eye (Chapter 2). This has led to the development of interaction techniques designed to make the selection of objects easier and more accurate.

Approaches such as screen magnification, target expansion and intelligent gaze interfaces all have one feature in common. They all result in the enlargement of targets, either in visual or motor space, or both. In the case of the intelligent gaze interface (Section 3.6), the expansion is not as obvious. Techniques which require all targets to be expanded permanently, widely spaced, or both, place severe limitations on the design of GUIs as screen space is limited. This is especially true for applications which need to display many controls and large quantities of information. Dynamic expansion, or localised magnification via a pop-up can be used to localise the expansion, easing the layout restrictions to some extent.

Interpreting usability evaluation results for eye gaze interaction techniques is a complex task. Published results are difficult to compare. Factors such as target sizes,
dwell time thresholds, users, and tasks vary greatly between experiments. Elements of different interaction techniques are sometimes combined, making it difficult to discern the extent to which individual factors influenced the experimental results. Thus, it is often not possible to make direct comparisons between techniques based on the available data. Although the reliance on published data presents a challenge, it is still possible to draw some conclusions about the merits of different approaches.

Of the six types of gaze selection triggers reviewed (Section 3.3), the two most popular techniques are currently dwell time and hardware button selection. Based on existing studies, these two trigger types appear to be the most practical. Experimental results suggest that object selection using these approaches can outperform selection using the mouse in some cases. Hardware button selection is more efficient than dwell time selection, but less accurate due to early and late trigger errors. While the implication that gaze can be a more efficient modality than the mouse is appealing, some comparisons are conducted using large targets. Comparisons between gaze selection techniques reveal that efficiency and selection accuracy degrade dramatically as target sizes are reduced to more realistic sizes. In most evaluations selection using the mouse is more efficient and accurate than gaze selection. Accuracy in particular is a major issue as error rates for gaze are significantly higher than those of the mouse.

Results published for magnification and target expansion techniques (Section 3.5) are particularly encouraging, as significant improvements in selection accuracy have been reported. The effect of these methods on efficiency depends on how they are applied. In some cases efficiency benefits are claimed. The results reported for EyePoint indicate that a magnification pop-up may provide a practical and efficient method for gaze selection, although accuracy is still a significant concern when compared to the mouse. Based on the evaluation results for EyePoint it can be argued that EyePoint represents the state of the art in gaze selection techniques.

Intelligent gaze interfaces (Section 3.6) rely on the presence of application-specific models of user behaviour. It is reasonable to assume that this behaviour may also vary between different users of the same system. This design could be problematic, unless the model developed is restricted to a small set of conservative assumptions about
user behaviour. Having to develop a model for each application also increases the complexity of incorporating eye-gaze interaction into a system. Results indicate that intelligent gaze interfaces can significantly reduce selection errors.

The idea of a gaze added interface (Section 3.4) is appealing as it lets users choose which modality to use for different tasks based on their personal preferences. Gaze added interfaces also allow interaction designers to combine the strengths of eye gaze interaction with those of traditional modalities. A gaze added approach could be used to add eye gaze interaction incrementally to existing systems. Eye gaze could be added for a limited set of cases were it is advantageous, focusing on its strengths.

Providing visual feedback for eye gaze interaction (Section 3.7) is viewed in a negative light. A gaze cursor indicating the user's current gaze point is viewed as particularly problematic due to feedback effects. There have been some positive results obtained for visual feedback in eye typing systems, but this positive view of visual feedback does not appear to be shared by the developers of gaze selection techniques in general.

While eye gaze is a natural and potentially efficient mode of interaction, based on the interaction techniques and systems reviewed in this chapter, there are still many obstacles to overcome before this type of interaction becomes practical for most tasks. Accuracy in particular is a concern, especially for smaller targets. The mouse is significantly more accurate than existing gaze selection techniques. Investigations have shown that it remains to be seen whether eye gaze interaction will gain widespread acceptance outside of its current primary application domain of supporting interaction for users with disabilities.
Chapter 4: Design of Gaze Selection Techniques

4.1 Introduction
New gaze selection techniques tend to incorporate elements from existing techniques in new ways. A set of design principles is proposed for developing new gaze selection techniques (Section 4.2). Four of the principles are based on a review of related literature, while three require further evaluation.

Most of this chapter consists of a description of the design of three new gaze selection techniques, namely StaggerPoint, ScanPoint and TargetPoint (Section 4.3). TargetPoint has a number of features in common with EyePoint (Section 3.5.2.3). The extent to which the new techniques developed and reported on in this chapter conform to the proposed principles is discussed (Section 4.4). An overview is given of additional techniques which were developed during the prototyping phase, but not evaluated formally due to scope limitations (Section 4.3.4).

4.2 Design Principles
The design principles for gaze selection techniques are intended for developers of new gaze selection techniques. These principles are not intended to be an exhaustive list covering every aspect of design, but rather a list of points to consider along with existing theory. Seven principles were developed, and are discussed below. The principles are divided into two categories, namely design principles based on literature and a proposed set of design principles for visual feedback which requires further evaluation.
4.2.1 Design Principles Based on Literature

The first four design principles are backed by existing research literature (Chapter 3). These principles cover topics related to disambiguation, gaze added interfaces, button triggers and selective magnification/expansion.

1. **Support target selection disambiguation to improve selection accuracy**

The limited accuracy of eye tracking equipment is a well known problem (Section 2.4.5). This limitation has had a significant effect on the design of virtually all existing eye gaze interaction techniques. Despite efforts to counter this problem, and the significant progress made, the accuracy of gaze selection techniques is still significantly lower than that of the mouse (Chapter 3). The selection of a suitable method for disambiguating target selections is therefore important. Direct selection of small controls, without any disambiguation mechanism (such as magnification), is impractical. This design principle is a generalisation of one of the EyePoint design principles, namely “use zooming/magnification in order to overcome eye tracker accuracy issues” (Kumar *et al*. 2007, p 4). Magnification is just one method of disambiguating target selections.

2. **Favour gaze added interfaces for able bodied users**

A gaze added interface is beneficial as it enables able bodied users to select an input modality which suits them based on their current task and personal preferences (Section 3.4). A system designed only for use by disabled users has no need for supporting gaze added interaction.

3. **Favour a hardware button trigger for able bodied users**

Given the Midas touch problem (Section 3.2.2), a suitable selection trigger needs to be chosen (Section 3.3). Hardware button and dwell time triggers are currently the most popular and practical. For able bodied users, a hardware button trigger (Section 3.3.4) is best as it provides a complete solution to the Midas touch problem. Fono and Vertegaal (2005) favoured a hardware button over a dwell time trigger for this reason. A dwell time trigger (Section 3.3.2), although more practical for disabled users, only offers a partial solution to the Midas touch problem. Unintentional double selections, for example, are still possible (Majaranta *et al*. 2004). This design principle relates to
one of the EyePoint design principles, namely “provide a fluid activation mechanism that is fast enough to make it appealing for able-bodied users and simple enough for disabled users.” (Kumar et al. 2007, p 4). EyePoint (Section 3.5.2.3) uses a hardware button trigger.

4. Magnification and target expansion should only be applied when performing a selection

The advantages of selective magnification (not “always on”) for target selection were clearly demonstrated by Ashmore et al. (2005) (Section 3.5.2.4). This principle should also apply to target expansion, as the effect of some target expansion techniques on the layout of a user interface are similar to those of magnification. In this manner the appearance of a user interface remains largely unchanged when the gaze modality is not being employed. This approach ensures that interaction with the mouse is not negatively affected with the addition of a gaze modality (Section 3.4).

4.2.2 Proposed Design Principles for Visual Feedback

Guidelines 5, 6 and 7 are based upon the argument that visual feedback in gaze selection techniques may provide usability benefits. Visual feedback is not generally favoured by researchers (Section 3.7). As these design principles are based on a contentious argument (with some supporting theory), additional research is required in order to determine whether this approach to feedback is effective (Chapter 7).

5. Visual feedback is useful, but should only be applied when performing a selection

The idea of providing users with visual feedback is appealing as it relates to two of Nielsen's ten heuristics (Nielsen 1994), namely visibility of system status and error prevention. If users are able to see that the item which they are about to select is incorrect, they can make a correction. One of the arguments against visual feedback is that the eyes should not perform a motor task (Section 3.7). Visual feedback should not be “always on”, but should be restricted to when a user is actually performing a selection to limit the frustration which may result from feedback permanently following the user's gaze. If restricting visual feedback in this manner is incompatible with a particular interaction technique, then it is best not to provide visual feedback.
6. Visual feedback should be in the form of a target highlight
The use of a cursor which follows the users' gaze point is problematic as it leads to a feedback loop. A more successful approach, which has been used in eye typing systems, involves highlighting the item which the user is currently looking at (Section 3.7).

7. Visual feedback should only be applied to sufficiently spaced/enlarged targets
A further restriction regarding the use of a feedback cursor, is that it should be restricted to situations where targets are sufficiently large or spaced (Section 3.5). In this manner, the risk of observing feedback effects similar to the gaze cursor (Section 3.7) are minimised.

These seven design principles were considered during the design of novel gaze selection techniques.

4.3 Novel Gaze Selection Techniques
The design of three new gaze selection techniques is discussed. The design of additional techniques which have not been evaluated, but were prototyped during this research, is briefly described. The primary focus is on three gaze selection techniques, namely TargetPoint, StaggerPoint and ScanPoint. The two latter techniques are designed to explore alternative methods for target disambiguation.

Although the techniques described in this chapter are largely compatible with a gaze added approach, users will not be provided with the opportunity to switch modalities during the full evaluation (Chapters 6 and 7). This is to ensure that the users evaluate the gaze selection techniques and do not simply fall back to the mouse for most activities.

4.3.1 TargetPoint
The design of TargetPoint is based on existing eye gaze interaction research. TargetPoint combines visual feedback, motor space expansion and a pop-up approach.

TargetPoint was developed to create a new technique combining the best elements of existing techniques. The three visual feedback highlight design principles (Section
4.2.2), were developed based on theoretical arguments and incorporated into TargetPoint.

Target magnification pop-ups have formed the basis for at least three existing gaze interaction techniques (Lankford 2000, Ashmore et al. 2005, Kumar et al. 2007) designed to improve performance for target selections (Section 3.5.2). Based on the literature surveyed, it would appear that the idea of a pop-up has not yet been applied to target expansion. One of the main advantages of a pop-up approach (as opposed to permanent) magnification is that it is preferred due to the ability to preview during visual search tasks (Ashmore et al. 2005). Furthermore, components on the screen appear as they normally would (not enlarged) until the pop-up is activated. This is different to permanent visual magnification/expansion. The existing design of the user interface is thus not affected for users who prefer not to employ the gaze modality. If all components have to be enlarged to facilitate eye gaze interaction, then fewer components and less information can be displayed on the screen at a time, potentially compromising the user interface design for other modalities. Providing support for traditional modalities (in addition to gaze) is important as it provides users with a choice (Section 3.4).

The motor space expansion of targets without corresponding visual expansion has also been demonstrated to improve target selection efficiency and accuracy (Section 3.5.1.1). The lack of corresponding visual expansion provides the additional advantage of drawing the users' gaze closer to the centre of the target, providing a further accuracy benefit over traditional magnification (Section 3.5.1.1).

Given the benefits of a pop-up based approach, and motor space expansion, these two approaches are combined to form a single technique named TargetPoint. The “look-press-look-release” key trigger model of EyePoint (Section 3.5.2.3) has been adopted for TargetPoint as this approach solves the Midas touch problem (Section 3.2.2), and has proven successful in user testing (Section 3.5.2.3). This selection triggering technique is designed primarily for use by able bodied users, but can be substituted by alternative triggers more suitable for disabled users (Kumar et al. 2007).
In order to simplify development and to facilitate a comparative evaluation with EyePoint (Chapters 6 and 7), TargetPoint incorporates a number of elements from EyePoint. The default values for magnification factor (four) and the screen area to be magnified (120x120 pixels) used by EyePoint have been adopted. An implementation of the gaze smoothing algorithm used by EyePoint was implemented based on the published pseudocode (Kumar 2007). Unlike EyePoint, TargetPoint expands targets in motor space, but not visually (Section 3.5.1.1). This means that a target, such as a button, appears exactly the same size when expanded in motor space (Figure 4.1) as it does before expansion. The difference is that it is surrounded by an area which is considered to be part of that button. If the user “clicks” within this surrounding area, the system reacts exactly as it would have if the user had “clicked” on the button itself. In comparison, contrast this with the visual magnification used by EyePoint (Figure 4.2). With EyePoint the target region is magnified both visually and in motor space.

![Figure 4.1: Motor Space Expansion Example - TargetPoint](image1)

![Figure 4.2: Magnification Example - EyePoint](image2)

Given the fact that the feedback loop for cursors is due to the inaccuracy of eye trackers, this issue is less likely to occur if a more general item highlight is used within a magnified view. The most common application of this type of feedback is in
the domain of eye typing (Section 3.7). TargetPoint provides a visual feedback highlighting for the pop-up window. By only providing the highlight in the pop-up window, the user does not have to see the highlight when they are simply looking around (a similar effect to the preview benefit noted by Ashmore et al. 2005). A semi-transparent blue highlight is painted over the item that the user is currently looking at (Figure 4.4 b and c). Initially the control closest to the gaze point reported by the tracker when the trigger key was pressed is highlighted. Users can then simply shift the highlight by looking at a different control. No dwell time threshold was employed for highlighting, as the Midas touch problem is avoided through the use of a key trigger. The absence of a dwell time threshold also aims to maximise the responsiveness of the system.

One of the motivations behind the use of the highlight, was the issue of selection errors resulting from the eye-hand span noted by Salvucci and Anderson (2000), Xiao, et al (2005) and more recently Kumar, et al. (2008) who classified these errors as either early trigger errors or late trigger errors (Section 3.3.6). The advantage a highlight provides is that users are able to see which item will be selected when a selection is triggered, potentially reducing the number of errors. For the design of TargetPoint, it was hypothesised that early trigger errors in particular are less likely to occur. Users are less likely to release the trigger key before looking at an item if the key is only released after the correct item has been highlighted. In the case of the EyePoint study (Kumar et al. 2008) early trigger errors were more common than late trigger errors.

When the trigger key is pressed, the TargetPoint pop-up is displayed in two phases. Before the trigger key is pressed, the set of targets is displayed (Figure 4.3 a). In the first phase, called the elimination phase, the pop-up window is displayed over the target area, but no motor space magnification (spacing) occurs (Figure 4.3 b). This occurs when the user presses (and holds) the trigger key down.
The effect of this first phase is that target items in the vicinity of the users’ gaze which fall outside the area to be magnified are hidden from view. After a delay of 100 ms has passed, the motor space magnification takes effect, and the remaining targets are spaced (Figure 4.4 c). The purpose of this sequence is to aid users in seeing how the magnified pop-up relates to the initial view. The actual selection occurs when the user releases the trigger key. When this occurs, the item which is currently highlighted in the pop-up gets selected.

Initial versions of TargetPoint followed a basic strategy of centering the pop-up window around the user's gaze point, as other pop-up techniques also do (Ashmore et al. 2005, Kumar et al. 2007). Prototyping revealed this strategy to be problematic for TargetPoint. When the user looked at a component, the pop-up would typically display the component offset by a distance from its initial location, both vertically and horizontally, resulting in users having to shift their gaze to continue looking at a
control which they were already looking at when the trigger key was pressed. In the case of EyePoint, the magnified component overlaps with the original component due to the visual magnification. This is, however, less likely to be the case with TargetPoint, as each component is displayed in the *centre* of an area equivalent in size to the magnified component size used by EyePoint. This central portion often did not overlap with the original bounds of the component. In order to counter this problem, an alternative pop-up placement strategy was employed. This strategy is based on the reasonable assumption that the component which the user is looking at when the trigger key is pressed (to display the pop-up) is the one most likely to be the intended target. With this in mind, the pop-up is positioned in such a manner that the view of this likely target in the pop-up overlaps perfectly with the corresponding component obscured by the pop-up.

A second issue observed with initial prototypes was that displaying the target control (for example a button) in the centre of the expanded target area was not always the best strategy. If the expanded target area is only partially visible within the pop-up (along the edge), then placing the button in the centre of the expanded area would often place the control outside the bounds of the pop-up (and therefore not visible to the user at all). A better strategy of drawing the components in the centre of the visible portion of the expanded area was adopted. The visible portion represents the intersection of the expanded target area and the bounds of the pop-up. Consider the example of button 7 (Figure 4.4). Without this refinement, button 7 would have been invisible (Figure 4.4 b). Instead, button 7 is drawn higher up (Figure 4.4 a). With the magnified pop-up of EyePoint this situation results in only part of the component being displayed. This effect is due to the expanded area corresponding with the visual size of the targets.
The visual feedback highlight of TargetPoint can be disabled. In pilot testing there was little interest in this variation of TargetPoint, so this approach was not adopted. The popularity of the feedback highlight became clear when the majority of pilot test users wanted to know why a similar highlight hadn't been provided for EyePoint, and regarded the TargetPoint version with feedback as superior. TargetPoint was only evaluated with the feedback highlight activated (Chapters 6 and 7). There is also a second reason why the no-feedback version of TargetPoint was not evaluated in the full evaluation. It was to avoid any learning effect advantage being given to TargetPoint by testing two versions of the same technique. Using TargetPoint twice as much would have resulted in test participants becoming more familiar with this technique than with the others.

![Figure 4.4: Target Placement Within TargetPoint Pop-up](image)
4.3.2 ScanPoint

ScanPoint was developed to explore the use of timing information, together with visual feedback, for disambiguating target selections. If a user can press a button and simply wait until the correct item is highlighted, it should be possible to achieve a high degree of accuracy.

ScanPoint uses a scanning concept similar to that used in systems for the disabled (Section 3.4.1.1). Instead of starting the scanning process from a fixed point, ScanPoint begins the scan from the item which is closest to the user's most recent gaze point. Gaze data is used to select the starting point for scanning, and is not simply used as a trigger. The idea is to minimise the time needed for scanning. A two-stage hardware button trigger is used to perform a selection. The user presses the trigger key down to commence scanning and releases it once the intended target item is highlighted to select it. As noted in the TargetPoint design discussion (Section 4.3.1) a hardware button trigger can be substituted by an alternative trigger more suitable for disabled users.

Instead of using a magnification or target expansion pop-up, ScanPoint begins by determining which control/target is closest to the most recent gaze point when the button is pressed. The item identified is highlighted by a semi-transparent blue highlight (Figure 4.5). As there is no expansion or magnification it is unlikely that the initial estimate will be correct often enough to provide sufficient accuracy.

![Figure 4.5: A Selection Highlight for ScanPoint](image)
In order to disambiguate the selection, a scanning process begins. If the initial item which is highlighted is correct, the user simply releases the trigger key to select it. If not, the user keeps the trigger key held in and waits until the correct item is highlighted before releasing.

Existing scanning systems start scanning from a fixed point, whereas ScanPoint starts scanning from the target closest to the user's most recent gaze point. It is reasonable to assume that there is a good chance that the item that the user wishes to select is close to this point. If scanning proceeds in a linear fashion it will need to do so in some direction away from this point. Proceeding in only one direction away from this point, however, is problematic. If the intended target is close to the initial target highlight, but in the opposite direction to the linear scan it may be a long time before the scanning highlight wraps around to reach the correct item. A more sensible option, based on the heuristic that the intended item is near the initial gaze point, is to have the scanning highlight radiate outwards from the initial point.

As ScanPoint does not require any modifications to the layout of an existing user interface it is ideal for use in a gaze added context (Section 3.4). An existing system can therefore be adapted with relative ease to use this technique. One possible advantage of this technique is that, if the accuracy of the eye tracking is good, there is a good chance that the initial target highlight will be correct and the user will simply release the trigger key immediately without having to fixate a second time in a pop-up. Another possible strength of this technique is relevant to users whom a tracker tracks poorly. These users may be unable to interact using other gaze selection techniques due to accuracy issues. With ScanPoint, the selection highlight will always reach the intended target eventually.

The prototype system is designed for a linear grouping of targets, so the second item to be highlighted in the scanning process is either the target above or below the initially highlighted item. If it is possible to determine whether the item above or the item below the initially highlighted item is more likely to be the intended target, the scanning time can be reduced by ensuring that the highlight moves to that item next.
The first attempt at determining whether the intended target was above or below the initial highlight was based on the idea of measuring gaze shifts as used by Špakov and Miniotas (2005). This idea is based on the fact that the user should be looking at the intended target when they pressed the trigger key. If the highlight highlights the incorrect item, the user's gaze should shift in the direction of the target highlight (away from the intended target). By determining the direction of this shift it should be possible to obtain a more reliable indication of where the intended target is relative to the current highlight. This method was implemented, but did not work well in practice. The most likely reason for this is that small shifts in the user's gaze were lost in the general noise associated with tracking data (Section 2.4.5).

A second design was based on an observation in pilot testing that it was common for corrections to always be made in the same direction by the same user. This is most likely a calibration issue. Based on this observation, it was decided to keep a count of the number of times a user made a correction in a downwards direction and compare it to the number of times a correction was made in an upwards direction. A correction in this case refers to the user waiting to release the trigger key until a target other than the initially highlighted item is selected. By comparing these two values it is possible to determine which direction is more common and adjust the scanning sequence accordingly.

The time delay for the scanning process represents a trade-off. If a delay which is too short is used, the users may not be able to react in time to select the correct item. A longer delay leads to user frustration as they have to wait longer for the scanning highlight to reach the item which they want to select. A delay of 600 ms was selected for this technique.

In order to compensate for users' reaction times, it was decided that if an item was selected within 100 ms of it being highlighted then the selection should revert to the previous item. Pilot testing revealed that this was not always a good idea as users would anticipate when a highlight was about to fall on an item. For this reason, this feature was disabled.
4.3.3 StaggerPoint

StaggerPoint represents an attempt to minimise the use of screen space while retaining a simple single step selection trigger. Unlike the previous two techniques, a selection is performed by simply pressing the trigger key. Releasing the trigger key plays no role. The original idea behind this technique was to harness smooth pursuit eye movements to disambiguate target selection. After experimenting with prototypes an alternative design was adopted.

The original design for this technique was based on the literature survey of eye movements (Section 2.2). Smooth pursuit eye movements can only result from observing a moving target, and are not relevant to gaze interaction. It was hypothesised that if controls/widgets were to continuously move in different directions, it would be possible to disambiguate target selections based on the direction or smooth pursuit eye movements resulting from the user tracking the intended target item before selecting it. In order to provide room for movement, the buttons were staggered by 33% (Figure 4.6). Alternative rows would move in opposite directions. When the first button was moving to the left, the one below would be moving to the right. Once the button reached the end of its movement in one direction (run out of space to move further within the 33% slack space), it would reverse direction.

Due to the noisy nature of eye tracking data (Section 2.4.5) it was necessary to use linear regression to determine the least squares line through the x-coordinates versus time of the observed gaze points. The gradient of this line was used to determine the direction of the eye movement along the horizontal plane (the button movements were all horizontal movements). The gradient of this line was compared to the velocity of the buttons. As described by Mendenhall and Sincich (2003), the Pearson product moment coefficient of correlation was used as a measure of the straightness of this line. The intention was to use this measure to help determine how many preceding gaze points to include in the calculation. The line should be reasonably straight due to the fact that the buttons are moving at a constant velocity.
Informal testing of a prototype revealed this design to be flawed. Noise in the gaze data meant that even though linear regression had been used, the gradient of the x-coordinates of the gaze points versus time plot could not be matched to the velocity of the buttons. Even the direction of the eye movement, as represented by the gradient, proved problematic as a disambiguation mechanism. Noise in the tracking data, the buttons switching directions during selection and the general annoyance factor of having buttons “dancing” around the screen led to this design being abandoned in favour of an alternative.

An alternative design was implemented in which the buttons were fixed in place. Instead of using smooth pursuit eye movements to disambiguate selections, the x-coordinate of the user's gaze point was used. This design, and the previous (discarded) design, are both designed based on the same observation. Controls, such as buttons are often wide enough not to pose a challenge when selecting using gaze. The problem is that these buttons are not usually tall enough to be selected unambiguously using gaze. In the current design (Figure 4.7), even number buttons always appear further to the left, and odd number buttons further to the right.
By taking into account whether the user is looking further left or further right it is possible to get a better idea of where the user is looking. Consider the example of a user wishing to select button 5 (Figure 4.7). The user looks at button 5 and presses the trigger key. Due to the limited height of button 5 it is not clear whether the user intended to select button 4, 5 or 6. The y-coordinate of the user's gaze point (as reported by the tracker) may in fact fall within the top half of button 6. A red mark is used to mark the user's gaze point for illustrative purposes in Figure 4.7. In order to disambiguate the selection, the x-coordinate of the gaze point is taken into consideration. If the user's gaze falls within the top half of button 6, but is closer to the center of button 5 in the horizontal plane, then button 5 will be selected. In this manner the effective height of each button is doubled in motor space. The cost of this optimisation is a 10% increase in space due to the need for staggering. Initially a 33% staggering gap was used, but this was reduced and the text on each button justified in the direction of the staggering.

The main advantage of this technique is that a selection consists of only a single step, with the staggering effect used to disambiguate the selection instead of a pop-up. This
technique is less general than the other techniques described in this chapter as it requires a stacked set of rectangular targets. It also requires modifications to the layout of an existing user interface (staggering, text justification). Modifying the layout of an existing user interface in this manner also results in a loss of aesthetic appeal.

\section*{4.3.4 Additional Technique Prototypes}

In this section additional gaze selection techniques which were prototyped, but not formally evaluated are briefly discussed. The purpose of this section is to provide some additional background on the various ideas which were considered during the early prototyping stages. These techniques were not formally evaluated to keep the scale of the evaluation phase of this research tractable (Chapters 6 and 7). Reasons why each of these techniques was not selected for evaluation are provided in the discussion.

\subsection*{4.3.4.1 Gaze Proximity Typing}

Gaze proximity typing is designed for cases where target items have text labels associated with them, for example buttons and menu items. The user looks at the item he or she wishes to select and starts typing the text in the label associated with that item. A text prefix, based on the text the user types is built. A dictionary is constructed from the items in the vicinity of the user's gaze. Once enough prefix text is provided to narrow down the dictionary to a unique item, that item is selected.

Issues to consider include determining when one command ends and the next starts. A time delay is used, so that if the user stops typing for a period exceeding a threshold value, any text entered after that point is considered a new command prefix.

This technique would be difficult to evaluate as the delay would depend on the typing proficiency of the test participant. A second issue is the text labels associated with the targets. The number of keystrokes required for a selection depends on the similarity of the text labels of the different targets. This technique is not applicable in cases where some targets do not have text labels associated with them.

A similar technique involving speech is that of Miniotas et al. (2006), which is described in Section 3.3.5. The concept is also similar to that of Zhang et al. (2004)
who used gaze to improve the accuracy of speech input based on where the user was looking.

4.3.4.2 Gaze Scroll Wheel Selection

Gaze Scroll Wheel Selection combines the mouse scroll wheel with gaze to select an item. To select an item the user looks at it and presses down the mouse scroll wheel. A mouse with a scroll wheel that can act as a middle mouse button is required for this to function correctly. While the scroll wheel is depressed, the item closest to the user's gaze point is highlighted. If the correct item is highlighted, the user releases the scroll wheel to make the selection. In the case of an incorrect item being highlighted the user rotates the scroll wheel to cycle through alternative targets in the direction the scroll wheel is rotated. Once the correct item is highlighted the user releases the scroll wheel.

This technique was not selected for evaluation as it requires the user to use the mouse in a manner which they do not normally use it (scrolling the mouse wheel while pressed in). Pressing the mouse scroll wheel in can also trigger other actions on some systems.

4.3.4.3 Gaze-Joystick Selection

Gaze-Joystick Selection is similar to Gaze Scroll Wheel Selection, except that it relies on a joystick for making corrections. Unlike the mouse scroll wheel, a joystick can be used more easily to provide directional input. To select a target item, the user looks at it and presses and holds in the fire button on the joystick. If the correct item is highlighted the user releases the fire button to select it. If an incorrect item is highlighted the joystick is used to steer the highlight towards the correct item. Once the correct item is highlighted the fire button is released.

This technique was not selected for evaluation as it is somewhat clumsy to use with a full-sized joystick (the current implementation). Ideally a low-profile thumb joystick would be used that supported pressing and releasing. A joystick of this type, although not involving gaze, is found on some digital cameras (Panasonic 2008). Pressing and releasing the fire key on a traditional joystick could be replaced by the press and
release action supported by the low profile joystick cited above. Developing the hardware required for this refinement is beyond the scope of this research.

### 4.3.4.4 Tagged Pop-up Selection

The tagged pop-up selection technique was developed before TargetPoint. As with TargetPoint motor space expansion is used to improve selection accuracy. When the user presses down the designated trigger key, a transparent pop-up appears around the area where he or she is looking (Figure 4.8).

![Figure 4.8: The Tagged Pop-up Technique](image)

To select an item, the user looks at the tag (blue rectangle) which corresponds to the item they wish to select and releases the trigger key. The tags are arranged around the area where the user was looking when the trigger key was pressed. Although the tags appear small, motor space expansion has taken place, ensuring that the actual size of each tag in motor space is much larger. The default motor space size of each tag is 90 x 80 pixels, compared to the default visual size of 20 x 10 pixels.

The first implementation of this technique used red lines to connect all tags to the corresponding targets. In pilot testing it became clear that the item which the user's gaze point was centred on was often the intended target. To minimise the number of saccades needed to make a selection, the tag for the initial target is placed directly
over the target. This is the reason for the blue tag appearing over Button 6 in Figure 4.8.

It was hypothesised that providing a feedback highlight would reduce the number of errors with this technique. Before this functionality was implemented though, it was decided that TargetPoint would be a better technique as the mapping between the motor space expanded pop-up targets and the corresponding controls/buttons was more natural with TargetPoint. The visual feedback highlight was added to TargetPoint. In pilot testing participants found the tagged pop-up technique to be accurate but awkward.

4.4 Conformance of Techniques to Design Principles

The conformance of the techniques proposed in this chapter to the seven design principles (Section 4.2) is summarised in Table 4.1. All of the techniques conform to the first three design principles regarding target selection disambiguation, favouring gaze added interfaces and hardware button triggers. StaggerPoint is the only technique which does not adhere to the fourth design principle, that target expansion should only be performed during a selection. This principle is designed to ensure a higher degree of compatibility with a gaze added approach ensuring an unchanged user interface. The permanent staggered arrangement of controls for StaggerPoint violates this principle. The reason for this design is to experiment with an alternative selection disambiguation mechanism (one of the objectives of this research) which simplifies selections to a single step.

Only four techniques provide visual feedback, so the visual feedback principles only apply to these techniques. All four of these techniques conform to principles 5 and 6. Visual feedback is limited to the period when users are busy performing selections (principle 5). In all cases feedback is provided in the form of a target highlight and not a gaze cursor (principle 6). Three of the techniques which provide feedback do not conform to principle 7, namely that visual feedback should only be applied to sufficiently spaced/enlarged targets. Among these techniques is ScanPoint, which provides visual feedback based on a timed delay (scanning). The location of the initial highlight does not have a sufficiently high probability of being correct as the targets...
are not enlarged or spaced (motor space expansion). The joystick and mouse scroll wheel techniques also do not conform to this principle for this reason. As with StaggerPoint, ScanPoint violates a design principle in order to explore a novel approach to target selection disambiguation (one of the objectives of this research, Section 1.3).

TargetPoint is the only technique to conform to all of the visual feedback design principles. The performance and user acceptance of this technique will therefore provide important supporting or contradicting evidence for these design principles (Chapter 7).

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Design Principles</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>General Principles</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>TargetPoint</td>
<td>Visual Feedback Principles</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>StaggerPoint</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>ScanPoint</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Gaze Proximity Typing</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Gaze Scroll Wheel Selection</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Gaze-Joystick Selection</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Tagged Pop-up Selection</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: Conformance of Techniques to Design Principles

**4.5 Conclusions**

A set of design principles for developing gaze selection techniques was proposed in this chapter. Four of these principles are purely based on theory and results from existing research studies (Section 4.2.1). Additional design principles, which relate to visual feedback (Section 4.2.2) are also proposed. These additional principles are based on an argument in favour of visual feedback, with some supporting theory. As
such, these principles require more supporting evidence of their efficacy (Chapter 7). TargetPoint is the only technique to conform to all of the visual feedback design principles. As such, positive results regarding visual feedback in TargetPoint would provide evidence to support these design principles (Chapter 7).

Gaze added interaction, whereby gaze is provided as an option in addition to existing modalities, is also emphasised in the principles and the design of the interaction techniques in this chapter.

The main focus of this chapter is on three new gaze selection techniques, namely TargetPoint (Section 4.3.1), ScanPoint (Section 4.3.2) and StaggerPoint (Section 4.3.3). TargetPoint combines motor space expansion, a visual feedback highlight, a hardware button trigger, and a pop-up based approach to target expansion. This represents a unique combination of features selected based on theory. TargetPoint also leverages existing research by incorporating various design elements which have proved popular with an existing technique called EyePoint.

All of the novel techniques proposed in this chapter require information regarding the bounds of each potential target item. This information is required for features such as motor space expansion and target highlights. EyePoint is designed to work with existing applications without any additional information. The techniques proposed in this chapter are not designed with this constraint in mind. Most existing applications would need to be modified in order to support the techniques proposed.

Due to the importance of providing a method for disambiguating target selections, two alternative approaches for selection disambiguation were explored. These are represented by ScanPoint and StaggerPoint (Section 4.3.2, 4.3.3). Many methods for disambiguating selections already exist, hence the need to explore more unorthodox approaches. ScanPoint uses a method called scanning which has previously only been applied to eye gaze interaction in a very limited manner. What makes ScanPoint unique is the use of the gaze point to indicate the starting point for scanning. StaggerPoint uses gaze information in the horizontal plane to improve selection accuracy in the vertical plane.
Chapter 4: Design of Gaze Selection Techniques

The three main techniques described in this chapter (TargetPoint, StaggerPoint and ScanPoint) are evaluated against the mouse and EyePoint in Chapters 6 and 7. The role and importance of visual feedback and users' perceptions thereof are also explored.
Chapter 5: A Framework for Gaze Selection Techniques

5.1 Introduction

The difficulty of comparing existing gaze selection techniques based on published results is clear (Chapter 3). Direct comparisons between techniques based on results reported for test setups which vary greatly are impractical. In order to effectively compare the usability of gaze selection techniques, a common test setup is required. Implementations of all the techniques to be evaluated and a common method of collecting test data are also required.

Given the need to develop various prototypes of gaze selection techniques (Chapter 4) and implement existing techniques for comparative evaluation (Chapters 6 and 7), it would make sense to program these techniques in such a manner that common code and functionality is shared between the implementations. This approach maximises code reusability. A solution to these issues is a software framework designed for implementing and testing gaze selection techniques.

In order to simplify maintenance and development of gaze selection techniques for this research, and to ensure a common platform for comparative testing, a software framework was designed and implemented. A common software framework for automating tests furthermore ensures that data is stored in a common format for easier analysis. Providing a framework which records data which is relevant to gaze selection techniques in particular ensures that essential information is available for analysis. It should be noted that the framework is designed purely for implementing

---

techniques for research purposes (evaluation) rather than practical implementations of
gaze selection techniques for existing applications.

Existing gaze selection techniques are typically implemented using proprietary
software development kits (Tobii 2008c) or interfaces designed for specific eye tracker
models (Chapter 3). These toolkits provide basic support for accessing gaze point data
without additional support for implementing gaze selection techniques. Implementing
a technique based on a tracker-specific interface also makes it more difficult to
upgrade to a tracker made by a competing manufacturer. The availability of a common
framework will address this problem.

A discussion of the framework is relevant because it represents part of the
implementation and testing methodology used for this research. The implementation
of TargetPoint is described as an illustrative example of the use of the framework to
implement a gaze selection technique (Section 5.8).

**5.2 Relevance of Framework**

Gaze selection techniques are typically implemented using interfaces or software
development kits (SDKs) provided by the manufacturers of eye tracking equipment
(Chapter 3). No mention is made of any common framework for implementing
functionality specific to gaze selection techniques or automating the presentation of
targets and gathering of relevant test data. Due to the apparent absence of such a
framework, and the need to prototype and evaluate multiple gaze selection techniques
for this research, it was decided to design and implement a software framework.

Libraries for obtaining gaze data from tracking equipment are typically specific to
each manufacturer of eye tracking equipment. This makes it more difficult to switch to
a new tracker from a different manufacturer. As commercial eye tracking technology
improves, it is sometimes necessary to switch from one manufacturer to another. If
classes included in the proprietary libraries (for example a class representing a gaze
point) are referenced throughout the source code it is more difficult to port the code to
a new library provided by a different manufacturer. During initial phases of this
research a monocular tracker from SMI was to be used, but later a binocular tracking
system, a Tobii T60, was purchased. Decoupling the implementation of the gaze selection techniques from a specific eye tracker ensured greater flexibility when considering a hardware upgrade.

The evaluation of gaze selection techniques requires a system for presenting target items to test participants. Relevant data, such as the time taken to perform each selection and the correctness of the selection, amongst other important values (Chapter 6) must be recorded for later analysis. In order to ensure a fair comparison, the testing framework used to evaluate each technique should be the same. Researchers typically have to develop their own data gathering algorithms. The data that is typically gathered by commercial usability testing software for eye trackers is inadequate for this purpose. Data typically gathered includes raw gaze point data, screen video captures and key events. Additional data, such as information about the placement of targets, the correctness of selections, filtered gaze point data and the configuration data for each technique is not recorded. Commercial software is designed for the evaluation of traditional user interfaces rather than the evaluation of eye gaze interaction techniques. In addition, visualisations do not support custom gaze point filtering algorithms (Section 2.4.5). If test data can be recorded in a simple machine readable format, such as XML, it should be simpler for researchers to develop their own custom visualisation and analysis tools which are specifically suited to their purposes.

To this end, a software framework for implementing gaze selection techniques, automating the presentation of test targets to test users and recording test data, was developed.

5.3 Framework Overview

The framework consists of two components, the gaze library and the gaze selection library (Figure 5.1). Access to gaze point data from the eye tracker is provided by the gaze library (Section 5.6). The gaze selection library (Section 5.7) provides support for implementing gaze selection techniques, automating the presentation of test targets to users and recording test data. Requirements for each of these components are discussed in the next section.
Figure 5.1: Framework Overview
Chapter 5: A Framework for Gaze Selection Techniques

5.4 Framework Requirements

The requirements for the framework are divided into three sections, namely requirements common to both the gaze library and the gaze selection library (Section 5.4.1), requirements for the gaze library component (Section 5.4.2) and requirements for the gaze selection technique library (Section 5.4.3).

5.4.1 General Requirements

The general requirements are both related to compatibility, namely with tracking equipment and with different operating systems.

The gaze library and gaze selection library should be independent of any particular eye tracker. This requirement enables applications to be developed for a common interface, shielding them from having to link directly to any tracker-specific routines. The gaze selection framework should satisfy this requirement by relying on the gaze library for gaze input. The implemented algorithms should be able to handle the different sampling frequencies of different eye trackers.

Although all evaluations for this research are to be conducted using the Windows operating system, supporting different operating systems ensures maximum flexibility in terms of future research. Operating systems used in eye gaze experiments vary. Examples include (but are not limited to) Windows (Fono and Vertegaal 2005, Kumar et al. 2007), Mac OS (Fono and Vertegaal 2005), Linux (Ashmore et al. 2005) and Solaris (Sibert and Jacob 2000). The gaze library and gaze selection framework should support as many platforms as possible. This requirement is not critical to the success of this research, but a reasonable effort should be made to ensure that the system works on other platforms, or can be ported with minimal effort.

5.4.2 Gaze Library Requirements

The purpose of the gaze library is to provide the framework with portable access to eye tracking data (Figure 5.1). This library is not required to interface with the tracking equipment directly, but rather to provide a common interface for accessing this type of data.
The gaze library should provide a **simple interface** for accessing gaze data. Accessing gaze data should require minimal effort on the part of developers.

Fixation jitter is inherent in raw gaze data provided by an eye tracker (Section 2.4.5). This data can be smoothed using a **filtering (smoothing) algorithm**. The gaze library should support the implementation of gaze data filters and include an implementation of at least one such filter for use by applications.

Currently, eye tracking equipment is expensive (Section 2.4.3). For this reason it is not always possible to have convenient access to an eye tracker at all times. It is therefore not always possible to perform all developmental work in the direct presence of a tracker. A simple solution to this issue is to **simulate tracker input using the mouse cursor**. This type of simulated input may not accurately reflect actual eye movements (Section 2.2), but is nevertheless useful for basic debugging and development purposes. Once a technique has been developed and debugged using the simulated input, it can be refined using an actual eye tracker.

### 5.4.3 Gaze Selection Library Requirements

The gaze selection library builds on the gaze library to provide a common foundation for the implementation of different gaze selection techniques (Figure 5.1). This library performs three main functions, namely simplifying the development of gaze selection techniques, recording of test data, and automated presentation of test targets to users.

The framework should **simplify the implementation of gaze selection techniques** by providing implementations of functionality common to these techniques and clearly defined extension points. By simplifying the implementation of such techniques, the implementation time should be reduced.

**Support for data collection** is perhaps the most important requirement. Reliable collection of sufficiently detailed data is critical for successful data analysis. The framework should record test data that is relevant to all gaze selection techniques. This data includes gaze point data, timing data and target data. A screen capture, taken at the point when each selection is made, should also be stored. Screen capture images are useful for creating customised visualizations based on gaze point data (Section
5.7.4) Storage of additional developer defined data that is specific to a particular gaze selection technique should also be supported. This information is essential for reconstructing the configuration of a technique for analysis or simulation.

The **presentation of target items to the user should be automated**. Automating tests makes it easier for the system to record which settings were used for which selection tasks and what the intended targets were. If the system does not know which target item the user intended to select, there is no way for it to determine the number of correct or incorrect selections. An automated test script also ensures consistency in the presentation of targets to test users.

### 5.5 Design and Implementation

The gaze library and the gaze selection library are implemented in Java as this is supported on a wide range of platforms, including all of those listed in Section 5.4.1. Operating systems that the framework have been tested on include Windows, Linux and Solaris. Testing on Windows has been more thorough as this is the operating system which was used for all experiments during this research. Basic compatibility testing with other operating systems was performed using simulated input due to the system attached to the Tobii eye tracker in the Nelson Mandela Metropolitan University (NMMU) usability lab only running Windows. This platform is also a more convenient choice as test participants are usually more familiar with it.

Demonstrating tracker independence is somewhat more difficult. Although the gaze library has only been tested with a Tobii T60 eye tracker, the interfaces on which the library depends are very simple and rely on basic information which is common to virtually all eye tracker models (Figure 5.2). In the unlikely event of a tracker not supporting validity data, a default value can be used. Timestamp data can also be provided by the framework. The same classes used for data from the Tobii tracker work just as well with data simulated using the mouse, demonstrating the extent to which the interfaces used shield applications from any particular tracker implementation. Developers wishing to access tracker-specific gaze point data can cast `GazePoint` instances (Figure 5.2) to a specific subclass at the point of use. This practice is, however, strongly discouraged, as it will tie an application to a particular
tracker model. All of the algorithms implemented in the framework are designed to adapt to the sampling frequencies of different trackers. Tracker frequency is used as a parameter.

5.6 Gaze Library

One of the most important interfaces in the gaze library is the GazePoint interface (Figure 5.2). This interface provides access to gaze point data such as the location of the gaze point on the screen, the time when the gaze point occurred, and the validity of the gaze point. Validity data is represented by an enumeration value representing one of three possible values, namely VALID, PARTIALLY_VALID, and INVALID. By defining a common representation for gaze point data, this interface shields applications using the library from tracker-specific data representations that would tie them to a particular tracker. All of the gaze library and gaze selection library functionality is built using this interface. As this library is tracker independent, those using the framework need to map gaze point data from their particular tracker to this interface. An example of such a mapping is a TobiiGazePoint class implementing the GazePoint interface for data provided by a Tobii T60 eye tracker (Figure 5.2). The SimpleGazePoint class represents a basic implementation of the GazePoint interface that is used for creating points for simulated or filtered gaze data.

![Figure 5.2: The GazePoint Interface](image)
Any class which requires gaze input from the eye tracker has to implement the \texttt{GazePointListener} interface (Figure 5.3). This method receives a gaze point event for each gaze point generated by the tracker. In order to receive these events, the \texttt{GazePointListener} needs to be registered with a \texttt{GazeDataProvider}.

![Figure 5.3: The GazePointListener Interface](image)

All gaze data provided by the library is exposed via the \texttt{GazeDataProvider} interface (Figure 5.4). The \texttt{DummyGazeProvider} class is used to simulate gaze point data using the mouse cursor. This data source is useful to developers when debugging applications. Invalid gaze points can also be generated to test that the applications can handle such points. The \texttt{TobiiGazeProvider} is an example of an implementation of the \texttt{GazeDataProvider} interface. Other providers can be implemented for different trackers.

The requirement for gaze data filtering support (Section 5.4.2) is satisfied by the \texttt{DefaultFilteredProvider} class (Figure 5.4). This class implements the
FilteredDataProvider interface using the smoothing algorithm of Kumar (2007, Section 2.4.5). The FilteredDataProvider interface acts as a marker, identifying a GazeDataProvider as being filtered. It does not provide any additional methods, as the interface is the same as that of the GazeDataProvider.

![GazeDataProvider Interface Diagram](image)

5.7 Gaze Selection Library

The gaze selection library provides a foundation for implementing gaze selection techniques (Sections 5.7.1 and 5.7.2), automating testing (Section 5.7.3) and recording test data (Section 5.7.4). Using the functionality provided by this library, the framework has been used to prototype new techniques in a matter of hours.
5.7.1 Implementing Selection Techniques

Selection techniques are implemented using the Swing GUI toolkit. Every selection technique is implemented by extending the `GazeTarget` class, either directly or indirectly. This class extends `JComponent` and represents a panel for displaying target items on the screen.

A new technique is typically implemented by extending the `BasicGazeTarget` class (Figure 5.5), which provides default implementations of the abstract methods of the `GazeTarget` class. Gaze selection techniques that involve the use of pop-ups (e.g., `TargetPoint`, `EyePoint`) can be implemented by extending the `GazeTargetWithPopup` class. This class provides a `setPopup` method which enables developers to register another `GazeTarget` as the pop-up for this class. In this manner, a `GazeTarget` can be used either as a standalone technique, or as a pop-up.

![Diagram of GazeTarget class and related subclasses](image)

Figure 5.5: The GazeTarget Class and Related Subclasses

5.7.2 The GazeTarget Class

A gaze selection technique is represented by a `GazeTarget`. An understanding of the interface of the abstract `GazeTarget` class (Figure 5.6) is essential for understanding how gaze selection techniques are represented using the framework.
A gaze selection technique implementation makes a selection by calling the `selectionTriggered` method. This call triggers the selection sequence (Figure 5.7). Researchers are free to call this method based on the use of a selection trigger mechanism of their choice. The default `GazeTarget` implementations use a key press.

The `resolveSelection` method is called whenever a selection is triggered, for example, the user pressing the trigger key (Figure 5.7). It is the responsibility of this method to determine the target item to select when the trigger is pressed. Data passed into this method includes the location of the most recent filtered gaze point, a list of the raw (unfiltered) `GazePoint` instances preceding the trigger point, a list of the filtered `GazePoint` instances preceding the trigger point, and two `GazeDataFuture` objects.

The `GazeDataFuture` objects provide access to the raw and filtered gaze points that will occur after the trigger has been pressed. This data is intended for use in cases where researchers would like to reduce the number of early trigger errors (Section 3.3.6). The number of early trigger errors can be reducing by resolving selections using gaze points which occur after the selection trigger is pressed. Early trigger errors
can occur due to various reasons including sensor lag, selection using peripheral vision, and latencies introduced by a filtering algorithm (Kumar et al. 2008).

Gathering gaze points after a selection is triggered presented a challenge in the implementation of the framework. It is impractical to wait for all of the relevant gaze points to occur each time before calling the resolve selection method. This approach would result in a performance tax being levied on all interaction techniques, even those which do not require future gaze data. The problem was solved by creating a `GazeDataFuture` interface and a default implementation class called `FutureGazeReader`. Instances of this class collect gaze points that occur after the trigger is pressed using a background thread. Using this approach, the `resolveSelection` method does not block unless future gaze points are requested from instances of this class.

Once the target selection has been resolved by the `resolveSelection` method, the actual selection needs to be performed by the `doSelection` method (Figure 5.7). This action typically results in the item selected being shown as selected. This method is separate from the `resolveSelection` method to allow the system to
capture a screen shot and other test data between the target resolution step and the selection being carried out. The time taken to record the screen capture and provide feedback to the user, for example the selected button being depressed, is not included in any timing results.

The constructor of `GazeTarget` takes two parameters, a raw `GazeDataProvider`, and a `FilteredDataProvider`. These data sources provide all gaze point data for calls to `resolveSelection`. Subclasses may include additional parameters. Client code is free to use a raw data source and filtering algorithm of their choice as the constructor relies on interfaces rather than any specific implementation.

In order to record the configuration of a selection technique during a test, and to automate the presentation of test targets, all `GazeTarget` instances must return an XML representation from the `toXML` method. An additional constructor which takes this representation as input is also required.

When an automated test is being performed, the user needs to be provided with some indication of which target item they should select. The purpose of the `setTargetIndex` method is to notify `GazeTargets` that a particular target item should be highlighted.

`GazeTarget` implementations are required to implement the `targetRect` method. It returns a rectangle representing the bounds of the currently highlighted target.

`GazeTargets` can be used as pop-ups for other `GazeTargets` (usually a `GazeTargetWithPopup` instance). In this manner a gaze selection technique is implemented as a combination of two `GazeTargets`. One acts as the owner, and the other the pop-up. The `setPopupOwner` method is called to register a `GazeTarget` as an owner of a pop-up. When the user presses the pop-up trigger, the pop-up appears. The selection is made when the user releases the pop-up trigger (which may be represented by any type of selection trigger). This method sets the owner value of a `GazeTarget`, preparing it for use as a pop-up. EyePoint (Section 3.5.2.3),
TargetPoint (Section 4.3.1) and ScanPoint (Section 4.3.2) are examples of techniques which can be implemented using pop-ups.

A pop-up must appear in the correct location when triggered. This is typically based on the location last filtered point (fixation). The `initLocation` method sets the location of the pop-up on the screen based on this location and performs any other initialisation tasks required before the pop-up can be displayed.

Screen shots, along with gaze data are required for later analysis of important events. This occurs by default for selections and gaze pop-ups. Some techniques may require data and screen captures for other important events. A screen shot is captured when the `takeScreenshot` method is called.

## 5.7.3 Automated Testing

The `TestManager` class (Figure 5.8) provides support for automated presentation of targets to test participants. This class makes use of a `GazeTargetPresenter`, which is typically a window, for displaying a `GazeTarget` panel. A `TestPlanItem` represents a set of trials for a particular interaction technique. The `GazeTarget` for these trials is constructed from its XML representation. Each test plan item can be stored to disk by recording its own XML representation as returned by the `toXML` method of this class. The XML representation of the `GazeTarget` is embedded within this string.

A `RepetitionType` is associated with every `TestPlanItem`. This represents the ordering in which individual target items are presented to participants during a test. If the `RANDOM_NO_SUCCESSIVE_REPEATS` option is specified, individual target items are randomly selected, with the restriction that no target item will be selected twice in a row. This is to avoid a user having to pick the same item twice in succession. The `RANDOM_NO_REPEATS` option also randomises the targets, with the provision that no target item is selected more than once, unless all the target items provided by a given `GazeTarget` have already been selected. When this occurs, the list of items already selected is cleared and the process starts again. This ensures that each of the target items is selected.

102
This property is useful as eye tracking for certain areas of the screen may be more accurate than for others. If some target items are selected more often than others, test results could be biased. Other options include RANDOM, SEQUENTIAL and REVERSE_SEQUENTIAL.

The framework classes can be used to construct testing applications which enable researchers to plan and execute experimental trials. An example of such an application
is the test planner system (Figure 5.9). This application, used in the experiment described in Chapter 6, demonstrates the practical usefulness of the testing system.

![Test Planner](image)

*Figure 5.9: A Test Planner Built Using the Framework*
5.7.4 Data Recording

The data recording functionality of the framework uses XML to store test data for analysis. Data is gathered from various sources, including the test plan, the selection technique implementations and the gaze data providers. The most important information is gathered during the selection resolution phase of a selection. This data includes:

- raw gaze data before a selection
- raw gaze data after a selection
- filtered gaze data before a selection
- filtered gaze data after a selection
- index of the selected item
- index of the intended target item
- time taken by the user to perform selection
- configuration of the test panel and targets

This type of information is also gathered whenever a pop-up is activated using a pop-up based technique.

A screen capture at the moment the selection trigger (or pop-up trigger) is pressed is particularly useful for analysis. The gaze point data, both raw and filtered, can be painted over this image to help understand how a particular target was selected based on the input data. This type of information can provide researchers with a very clear and accurate picture of the data that was used to perform a particular selection. An example of such a visualisation constructed from pilot test data is depicted in Figure 5.10. Gaze points are painted in different colours to represent different types of gaze point data (note the scattered nature of the raw data points compared to the filtered points). A green rectangle is used to indicate the intended target item.
Currently, commercial eye tracking analysis software is poorly suited to the evaluation of gaze selection techniques. These systems visualise either raw gaze points or fixations identified using their own undocumented algorithms. Researchers who use filtered gaze points to resolve selections need to be able to visualise these filtered points (which do not correspond to the filtered points displayed by the commercial analysis software). Customised analysis software developed to visualise the data gathered by the framework (Figure 5.11) has generally proven to be more useful for analysis purposes. This application is intended for data analysis rather than as a tool for general consumption, hence the (comparatively) simple interface. The combo boxes at the bottom of the interface enable the user to pick a user, session and particular selection to analyse. Each selection is represented by a screen shot (centre of the screen) and gaze point data captured by the framework. The gaze point data is visualised using coloured dots. Slider controls above the display area enable users to select intervals of gaze point data (raw or filtered) to display.
5.8 Implementing TargetPoint using the Framework

The framework has been used to implement at least seven gaze selection techniques (Chapter 4) in addition to EyePoint (Section 3.5.2.3). In order to demonstrate the practical usefulness of the framework, the implementation of TargetPoint (Section 4.3.1) using the framework is outlined as a representative example. The implementation of EyePoint and Tagged Pop-up selection (Section 4.3.4) using the framework has also been documented (van Tonder et al. 2008). The implementation of these techniques demonstrates how the framework classes can be extended to prototype new techniques, including an existing technique from documented manuscripts.

The design of TargetPoint is described in Section 4.3.1. This technique involves the use of a key trigger to display a pop-up displaying targets expanded in motor space.

Figure 5.11: Customised Analysis Software Example
(and spread visually). The location of the pop-up is based on the area where the user is looking when the trigger key was pressed (Figure 5.12). The user selects a target item by looking at it and releasing the trigger key.

Figure 5.12: TargetPoint Implementation Screen Capture

TargetPoint is an example of a pop-up based technique. It was implemented by creating an instance of the GazeTargetWithPopup class (Figure 5.5) and associating a pop-up with it. The pop-up in this case is an instance of a new class representing a TargetPoint pop-up. This class extends GazeTarget to create a spread out view of the initial target area.

The TargetPoint pop-up implements and overrides various methods of the GazeTarget class (Section 5.7.2). A summary of the important methods implemented follows.

In the initLocation method, the pop-up bounds are set based on the filtered gaze point data. The pop-up is then configured to display a spread out view of the original target area.
In the **resolveSelection** method the filtered gaze data is used to identify which item the user was looking at. If one of the invisible motor space rectangles around a target contains the most recent gaze point, it is selected.

The **targetRect** method returns the rectangle associated with the motor space rectangle that corresponds to the current target.

For the TargetPoint class the **doSelection** method fires an event which notifies the owner of this pop-up that a target at a particular index should be selected.

In the **toXML** method, a string is constructed using XML tags to store a representation of the current TargetPoint instance.

The current implementation of the TargetPoint class has approximately 650 lines of source code. Most of this code represents functionality unique to this interaction technique rather functionality that could have been handled by the framework. In order to determine the extent of code reuse in this implementation, consider the classes which the TargetPoint implementation extends. The **GazeTarget**, **BasicGazeTarget** and **GazeTargetWithPopup** classes (Figure 5.5) are all used to implement TargetPoint. These three classes alone consist of over 1200 lines of additional source code. Based on this figure, a very conservative estimate for the reduction of code due to the use of the framework when implementing TargetPoint is 65%. This figure excludes the code from the gaze data library (including the smoothing algorithm implementation), utility classes provided by the framework, significant code related to data collection and storage, and most of the automated testing functionality. Including a subset of the code related to these items yields a further 2000 lines of code. Thus, a conservative estimate is that had TargetPoint been implemented without the framework, the number of lines of code required to implement the necessary functionality (including data recording) would have been in excess of 3850. This reduction is equivalent to a code reuse percentage of over 83%.

### 5.9 Conclusions

The framework serves three main purposes, namely providing a common testing platform for different gaze selection techniques, ensuring consistent collection of
relevant data, and maximising code reuse when implementing new selection techniques. Data collection is simplified using the framework, as the data needed for analysis is collected automatically during the selection sequence (Figure 5.7). In Chapters 6 and 7 the use of the framework to successfully conduct an experiment and collect relevant data is demonstrated.

New gaze selection techniques can be rapidly developed and tested using the framework. The framework has been used to prototype new techniques in a matter of hours. Due to the common functionality provided by the framework, the quantity of code required to implement a new technique is significantly reduced, enhancing code reusability (Section 5.8). Implementation effort is restricted to functionality which is unique to each technique. The components are designed in a manner which ensures that they are not specific to any particular eye tracker. A common eye tracking data interface, which is provided by the gaze library, shields implementations from tracker-specific APIs. This interface has proved to be particularly useful for debugging purposes as simulated input can be provided through the same interface.

Seven novel gaze selection techniques have been implemented using the framework (Chapter 4). An existing technique, EyePoint (Section 3.5.2.3), has also been implemented. A common testing and data collection framework ensures that these techniques can be compared directly in an evaluation.

The framework also simplifies the collection of data for analysis. Screen shots captured automatically at critical points during a selection can be used together with the test data to visualise gaze points during selections (Figure 5.11). Data points from custom gaze data smoothing algorithms (which are often used in eye gaze interaction systems) can also be visualised accurately, a feature that is not generally supported by existing commercial systems.
Chapter 6: Experimental Design

6.1 Introduction

In order to evaluate the gaze selection techniques proposed in Chapter 4 a usability evaluation was conducted. The purpose of this evaluation was to compare the usability of three novel gaze selection techniques, namely TargetPoint, StaggerPoint, and ScanPoint to an implementation of EyePoint and the mouse. EyePoint was selected for this comparative evaluation as it arguably represents the current state of the art in eye gaze interaction. In a published evaluation (Kumar et al. 2007), EyePoint was found to be popular with test participants, and provided excellent accuracy and efficiency results (Section 3.5.2.3). The mouse is used as a basis for comparison as it represents the current standard for interaction with graphical user interfaces. The use of existing techniques as benchmarks for comparative evaluation is a proven method for evaluating gaze selection techniques (Zhai et al. 1999, Ashmore et al. 2005, Kumar et al. 2007). The design of the experiment is based on similar evaluations (Miniotas et al. 2004, Kumar et al. 2007, Kumar et al. 2008).

Using the framework (Chapter 5) to implement and evaluate all of the techniques ensures that the same test targets and conditions are used throughout, thereby facilitating a direct comparison. It also ensures consistency in the methods by which test data is collected and stored.

The research hypotheses tested are presented in Section 6.2. The pilot evaluation phase of the research (Section 6.3) contributed to the development of the final experimental design. This pilot evaluation description is followed by a discussion of the usability evaluation design (Section 6.4). The test venue and equipment, naming conventions, test procedure, collection and storage of test data are presented. Data
analysis tools and methods used to evaluated the test data are elaborated on (Section 6.4.5).

6.2 Hypotheses

The research hypotheses tested all relate to usability. $H_0$ refers to usability in general, whereas $H_{0.1}$, $H_{0.2}$ and $H_{0.3}$ refer to the three components of usability, namely efficiency, effectiveness and satisfaction (ISO 9241 1997). Each of the three novel techniques is compared to an implementation of EyePoint and the mouse. The components of usability and the metrics used to measure them are discussed in detail in Section 6.4.4. Each of the three research hypotheses represents one of these components.

$H_0$ (i, j) : Technique i is not more usable than technique j, $i \in \{\text{TargetPoint, StaggerPoint, ScanPoint}\}, j \in \{\text{EyePoint, mouse}\}$

$H_{0.1}$ (i, j) : Technique i is not more efficient than technique j,
\[ i \in \{\text{TargetPoint, StaggerPoint, ScanPoint}\}, j \in \{\text{EyePoint, mouse}\} \]

$H_{0.2}$ (i, j) : Technique i is not more effective than technique j,
\[ i \in \{\text{TargetPoint, StaggerPoint, ScanPoint}\}, j \in \{\text{EyePoint, mouse}\} \]

$H_{0.3}$ (i, j) : Technique i is not more satisfying than technique j,
\[ i \in \{\text{TargetPoint, StaggerPoint, ScanPoint}\}, j \in \{\text{EyePoint, mouse}\} \]

6.3 Pilot Evaluation

Two small pilot evaluations were conducted. Five participants participated in the pilot evaluations. As one of the purposes of these studies was to refine the techniques, it was sometimes the case that changes to the techniques would be made between different participants being tested. Refinements made to the different techniques are described in Chapter 4. The purpose of the pilot evaluations was also to highlight any issues with the test procedure (Heim 2008).
Chapter 6: Experimental Design

6.4 Usability Evaluation Design

Once the pilot evaluation phase was completed, an experiment to evaluate the usability of each technique involving 30 test participants (Section 7.2) was conducted. This experiment was conducted in the Nelson Mandela Metropolitan University (NMMU) usability laboratory. A description of the test venue and equipment (Section 6.4.1), naming of the techniques for the test (Section 6.4.2), test procedure (Section 6.4.3) and collection and storage of system and self reported metrics (Section 6.4.4) follows.

6.4.1 Test Venue and Equipment

The experiment was conducted in the NMMU usability laboratory, which is described in more detail by van Greunen (2002). This usability lab provides the standard equipment commonly found in usability labs as described by Barnum (2002). Each test subject was seated in the participant room, which is separated from the observer room by one-way glass. The observer room provides a remote desktop viewer and facilities for monitoring and recording video data. Communication between the observer and participant rooms is supported via microphone and speakers.

All tests were conducted using a Tobii T60 eye tracker (2008b). This system incorporates a binocular remote eye tracker into a 17” LCD monitor (Figure 6.1). The tracker samples at 60 Hz and is attached to the test system using a network cable. For all tests, the screen resolution was set to 1280 x 1024. Participants were seated approximately 60 cm from the tracker. A 2.9 GHz Core 2 Duo Extreme system with 2 GB RAM running Windows XP was used as a test platform.

Gaze point data is obtained from the tracker using a C# application developed using the Tobii SDK (Tobii 2008c). This application transfers the data over a TCP/IP socket to the testing framework which written in Java. A GazeDataProvider implementation (Section 5.6) for the Tobii tracker received the data, enabling its use within the framework (Chapter 5).
6.4.2 Naming of Techniques Evaluated

Five techniques were evaluated, namely TargetPoint, StaggerPoint, ScanPoint, EyePoint and the mouse. In order to aid test participants in remembering the names of the different techniques, each eye gaze interaction technique was given a descriptive, yet neutral, name for the test. Throughout the test TargetPoint was referred to as the “spaced pop-up”, due to the spacing between the components in the pop-up. The name “magnified pop-up” was given to an implementation of EyePoint which was developed based on the published descriptions of this technique (Kumar 2007, Kumar et al. 2007, Kumar et al. 2008). StaggerPoint was named the “staggered selection” technique, and ScanPoint was named the “timed highlight” technique.

The purpose of the evaluation was to compare the different approaches to gaze selection taken by different techniques, so even if there may have been differences between the “magnified pop-up” implementation and the original EyePoint implementation (Kumar et al. 2007), the comparison between the high level approaches of these techniques remains valid.
6.4.3 Test Procedure

Each test participant used each of the five techniques during the evaluation. The order in which the techniques were used was varied between participants in order to counterbalance any learning or fatigue effects. This ordering was stored as a unique test script for each participant. The test scripts were stored in XML using the framework (Chapter 5). The relevant test plan for each test participant was loaded from the disk and run using the TestManager class.

Test participants were calibrated using a 5 point calibration procedure provided by Tobii Studio (Tobii 2008d) at the start of the test. Participants were then presented with a grid of target buttons appearing in the centre of the screen (Figure 6.2). A button labelled “Start” was positioned so that its left edge was 250 pixels to the left of the edge of the first button, and its top edge was aligned with the top edge of the first button. Before each selection operation this “Start” button appears. When the participant fixated on this button for 100 ms it would disappear, and the text of one of the buttons in the target grid on the right would turn red, indicating the item which the participant should select.

Selection operations were timed from the point when the “Start” button vanished to the point when a selection (incorrect or not) was completed. The button on the left then reappeared, enabling participants to start the next selection. Participants were informed that the selection of this button was merely a way of triggering the start of each test, and that this selection did not form any part of the actual evaluation. For tests involving the mouse, participants would click on the “Start” button instead of looking at it. In order to ensure that all participants were able to select this button using gaze (and therefore start the test), it was expanded (invisibly) in motor space. The expansion extended 100 pixels beyond the left edge of the button and 100 pixels beyond the top edge. This expansion is represented by the shaded area (Figure 6.2). Participants were informed that if they had any difficulty selecting the start button they could look slightly above or to the left of it. Initially the plan was to have the “Start” button in the centre of a region extending 50 pixels in all directions around the button. This would effectively have meant that the right edge of the start button would be
closer to the target grid for the gaze interaction techniques than for the mouse, potentially giving the gaze techniques an advantage in the experiment. For this reason the expanded region was shifted to its current location (Figure 6.2).

The idea of triggering the start of each selection test by fixating upon an invisibly expanded home button is derived from the experimental design of Miniotas et al. (2004), who also expanded the home button invisibly in motor space. Triggering the start of a selection with this button ensured that participants always begin at the same point before a selection. In this manner, the distance that the eye (or mouse) has to travel to the target grid was the same for all participants. Without the home button, the participant's initial gaze position would have been left to chance.

The grid of target buttons was placed in the centre of the screen as this is the area where the tracker is most accurate. It was observed during preliminary studies that the Tobii T60 tracker is less accurate away from the centre of the screen with wide
variations in tracking accuracy towards the edges of the display. In order to avoid the
effects of these variations dwarfing the effect of the differences between the
techniques, it was decided to place all targets in the central portion of the screen. This
ensured the most consistent results.

As previously noted (Section 5.7.3), the selection of targets for gaze selection should
not be entirely random. If button 1 was randomly selected more often than button 8,
the test data may be skewed as the eye tracker may be more accurate in one part of the
screen than another. The number of test selections should be a multiple of the number
of targets in the grid. In the case of this experiment there are ten buttons. Each of the
ten buttons should be selected as a target at least once for every ten targets presented.
Targets are selected randomly by the system from the remaining pool of targets which
have yet to be presented as a target. The pool is repopulated after every ten selections.

For each of the five techniques being evaluated there were five test batches, each
consisting of 30 selections. The first batch was always a practice run, enabling
participants to become comfortable with each technique before using it. The four
batches which followed are the test batches. For each test batch the participant was
given a goal. An instruction to “try to select the targets quickly” (speed condition) or
to “try to select the targets accurately” (accuracy condition) was given. This idea is
based on an existing experimental design (Kumar et al. 2008).

Two target sizes were provided. For some tests the small buttons were used and for
others the large buttons. The size of the large buttons was 79 x 23 pixels. The small
buttons measured 79 x 15 pixels. According to Bates (1999) the sizes of common
components in the Windows operating system range from 10 to 32 pixels. The height
of both the small and large target sizes used fall within this range. As the Swing GUI
toolkit for Java is used to implement the framework used for the test, the button width
(and height in the case of the larger target size) was based on the preferred size for a
button with the text “Button 10” in the Windows XP look and feel.

Combining the two test conditions (size and goal) yields four combinations, hence the
four test batches for each technique (excluding the practice batch). The four
combinations multiplied by the number of techniques yields twenty non-practice test
batches, each consisting of thirty selections per test participant. Each test participant would therefore perform 600 test selections during the experiment.

6.4.4 Collection of Usability Metrics

The purpose of the evaluation is to compare the usability of TargetPoint, StaggerPoint and ScanPoint to that of EyePoint and the mouse. Usability consists of three components, namely efficiency, effectiveness and satisfaction. The International Standards Organisation (ISO) defines usability:

“Usability is the **effectiveness**, **efficiency** and **satisfaction** with which specified users achieve specified goals in particular environments; where **effectiveness** is the accuracy and completeness with which specified users achieve specified goals in particular environments; **efficiency** is the resources expended in relation to the accuracy and completeness of goals achieved; and **satisfaction** is the comfort and acceptability of the work system to its users and other people affected by its use” (ISO 9241 1997).

In order to measure usability, a combination of system recorded and self reported metrics were collected. The framework (Chapter 5) provides the means to gather detailed quantitative information regarding each selection with each technique. Tobii Studio (Tobii 2008d) was used to create screen recordings of the tests and log keystrokes.

Self reported data was collected by means of questionnaires. Before the test, participants completed a pre-test questionnaire (Appendix A). Participants were asked to fill in their age, how many years they had been using computers and the mouse, and other biographical information. Participants were also asked to fill in whether they were wearing glasses or contact lenses for the test.

Upon completion of the test tasks, participants completed a post-test questionnaire (Appendix B). A five point Likert scale was used to rate various aspects of the techniques. A value of 1 corresponded to **strongly disagree** and 5 corresponded to **strongly agree**. At the end of the post-test questionnaire participants were asked to rank the techniques in order of preference. For each technique there were sections for
participants to comment on both positive and negative aspects, as well as for providing general comments and feedback. Comments made during the test were also noted.

Data collected during the test was used to compare the usability of the techniques. Specific metrics were used for each of the three aspects of usability, namely effectiveness, efficiency and satisfaction. For each variable the mean was used for comparison purposes.

**Efficiency** ($H_{0.1}$) was measured as the number of correct selections per minute. This value included the time participants took to perform both correct and incorrect selections, but not any idle time between selections. Efficiency was expressed in terms of (correct) selections per minute. This measure of efficiency was based on the recommendation of Tullis and Albert (2008), which is based on the definition of efficiency provided by the Common Industry Format for Usability Test Results (NIST 2001). The following efficiency metrics were used:

- Selection times (average time per selection in milliseconds)
- Selection rate (number of correct selections per minute)
- Perceived efficiency (5 point Likert scale)

**Effectiveness** ($H_{0.2}$) in the context of a gaze selection technique refers to the accuracy and correctness of selections. The following effectiveness metrics were used:

- Selection accuracy (percentage of correct selections and errors)
- Perceived accuracy (5 point Likert scale)
- Perceived accuracy decline for smaller targets (5 point Likert scale)
- Correct and incorrect selections were identified from the data collected by the framework.

**Satisfaction** ($H_{0.3}$) was measured using five metrics. General satisfaction, ease of use, user preference rankings, perceived mental effort and perceiving physical effort. Perceived mental and physical effort are components of satisfaction (Douglas et al., 2011).
1999). Satisfaction and ease of use are also commonly used satisfaction metrics (NIST 2001). With the exception of the preference ranking, all values were reported using a 5 point Likert scale. The use of a small set of questions with a 5 point Likert scale is consistent with a similar evaluation used to compare techniques for selecting windows using gaze (Fono and Vertegaal 2005). The satisfaction metrics used included:

- Preference ranking (ranking of techniques in order of preference from 1 to 5)
- Ease of use
- Overall satisfaction
- Perceived mental effort
- Perceived physical effort

For TargetPoint, participants were also asked to indicate whether they found the target highlight useful using a 5 point Likert scale (1: strongly disagree, 5: strongly agree).

### 6.4.5 Data Analysis Tools and Methods

T-tests\(^1\) were used to analyse the Likert scale data. This technique is appropriate for interval data of this type (Tullis and Albert 2008). Performance metrics, such as average selection times, average selection success, and average efficiency were also analysed using t-tests. T-tests are suitable for analysing data of this type (Tullis and Albert 2008). When a statistically significant difference was found, a Cohen's D value was calculated. This value is a measure of practical significance. The user preference rankings for the different techniques were analysed using the Wilcoxon rank sum test. An alpha value of 0.05 was used. In order to compare each technique within the context of the four other techniques being evaluated, Bonferroni correction was applied. This involves dividing the threshold of 0.05 by the total number of techniques evaluated minus one, resulting in an alpha value of 0.0125 for comparisons between techniques.

The data was analysed using Statistica (StatSoft 2007), Microsoft Excel, Tobii Studio (Tobii 2008d) and customised analysis software developed by the author to read in

---

\(^1\) T-tests were selected for this evaluation in consultation with a statistician
data recorded by the testing framework (Chapter 5). Simulations to analyse the effects of trigger point time shifting (Section 3.5.2.3) and selection disambiguation mechanisms were also used. The use of simulations has already proven useful for simulating adjustments to parameters of gaze selection techniques and measuring the effects (Kumar et al. 2008). Thematic analysis (Ely et al. 1995) was used to analyse qualitative feedback.

6.5 Conclusions

The hypotheses presented in this chapter were designed to evaluate the usability of the selection techniques designed in Chapter 4 and compare them to EyePoint and the mouse. EyePoint was selected for comparison purposes as this technique currently represents the state of the art in gaze selection techniques (Chapter 3). The mouse was also evaluated, as this is the input modality most users currently use for selecting items in a graphical user interface. A comparison including the mouse is useful for identifying the strengths and weaknesses of gaze selection techniques versus the mouse, specifically for target selection.

Appropriate data analysis techniques were selected to determine statistical significance for the hypotheses. In Chapter 7 the data analysis techniques selected are used to analyse the data and compare the different interaction techniques. This data is used to test each of the hypotheses identified in Section 6.2. The hypotheses are:

\[ H_0 \ (i, j) : \text{Technique } i \text{ is not more usable than technique } j, \ i \in \{\text{TargetPoint,}\ \text{StaggerPoint,}\ \text{ScanPoint}\}, \ j \in \{\text{EyePoint,}\ \text{mouse}\} \]

\[ H_{0.1} \ (i, j) : \text{Technique } i \text{ is not more efficient than technique } j, \]
\[ i \in \{\text{TargetPoint,}\ \text{StaggerPoint,}\ \text{ScanPoint}\}, \ j \in \{\text{EyePoint,}\ \text{mouse}\} \]

\[ H_{0.2} \ (i, j) : \text{Technique } i \text{ is not more effective than technique } j, \]
\[ i \in \{\text{TargetPoint,}\ \text{StaggerPoint,}\ \text{ScanPoint}\}, \ j \in \{\text{EyePoint,}\ \text{mouse}\} \]

\[ H_{0.3} \ (i, j) : \text{Technique } i \text{ is not more satisfying than technique } j, \]
\[ i \in \{\text{TargetPoint,}\ \text{StaggerPoint,}\ \text{ScanPoint}\}, \ j \in \{\text{EyePoint,}\ \text{mouse}\} \]
The test procedure is based on existing procedures used for evaluating gaze selection techniques. All tests were conducted in a fully equipped usability laboratory. The framework described in Chapter 5 was used to implement all of the techniques which were evaluated, automate the presentation of test targets to the participants, and record relevant data. As the framework has been used to successfully implement gaze selection techniques for experimental purposes and capture the data necessary for analysis, it can be concluded that the practical usefulness of the framework has been proven.
Chapter 7: Experimental Results

7.1 Introduction

In order to test the research hypotheses, the data collected during the experiment was analysed. Although most of the hypotheses can be tested using purely quantitative data, a significant portion of this chapter is dedicated to discussing qualitative feedback from participants. Qualitative feedback is discussed in detail as it provides unique insights into participants' opinions regarding the different techniques evaluated. This type of feedback provides a clearer picture of how users perceive different design elements of techniques, and aids in understanding why some techniques are more popular than others. Thematic analysis was used to analyse the qualitative data.

Quantitative results are reported first. System recorded metrics are discussed in Section 7.3, followed by a discussion of self reported metrics in Section 7.4. A discussion of the qualitative feedback provided by test participants is presented in Section 7.5. The research hypotheses are examined based on the evaluation results (Section 7.6). The design principles (Section 4.2) are revisited in Section 7.7, considering the results of the experiment.

7.2 Test Participants

Thirty users participated in the experiment (16 male, 14 female). The test population consisted of a convenience sample of unpaid volunteers drawn from senior students and faculty of the NMMU Department of Computer Science and Information Systems. All test participants had been using computers and the mouse for at least five years. None of the participants had used eye gaze as a form of input before. Seven

---

1 Chapters 4, 6 and 7 contain content from a paper which was submitted to the International Journal of Human-Computer Studies. TargetPoint : Accurate Gaze Selection Combining Visual Feedback with Motor Space Expansion, van Tonder, M., Cilliers, C., Greyling, J., submitted December 2008
participants wore glasses and two wore contact lenses during the experiment. A summary of test participants' computer experience and age is provided in Table 7.1.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer Experience</td>
<td>12.9</td>
<td>23</td>
<td>5</td>
</tr>
<tr>
<td>(in years)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mouse Experience</td>
<td>11.6</td>
<td>10.5</td>
<td>3.9</td>
</tr>
<tr>
<td>(in years)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td>24.7</td>
<td>23</td>
<td>5.4</td>
</tr>
</tbody>
</table>

Table 7.1: Test Participant Summary

As five techniques were evaluated with four different test conditions, the total number of test selections performed by the thirty test participants using the five techniques was 18,000 (30 selections x 5 techniques x 4 conditions x 30 test participants).

7.3 System Recorded Metrics

System recorded metrics include error rates, selection times and efficiency. This data was recorded using the framework (Chapter 5). System recorded effectiveness results are reported and discussed (Section 7.3.1). Efficiency results are presented in Section 7.3.2.

7.3.1 Effectiveness

Effectiveness is measured in terms of accuracy and selection errors. The error percentages recorded during the accuracy tests are depicted in Figure 7.1. All of the gaze selection techniques were significantly less accurate than the mouse for the accuracy task (Table 7.2). An error percentage of 0.6% (11 errors) was recorded for the mouse. The most accurate gaze selection technique, TargetPoint, had an error percentage of 3.28% (59 errors). T-tests reveal the difference between the mouse and TargetPoint to be statistically significant (p < 0.0125). A Cohen's D^2 value of 0.77 implies a moderate practical significance. The differences between the mouse and all of the other techniques are also significant (p < 0.0125).

1 Significance threshold is discussed in Section 6.4.5
2 If D < 0.20 then not significant
   0.20 <= D < 0.50 implies small difference
   0.50 <= D < 0.80 implies moderate difference
   D >= 0.80 implies large difference
Chapter 7: Experimental Results

For EyePoint the error percentage is 6.06% (110 errors). The least accurate technique was StaggerPoint (15.89% incorrect, 286 errors), followed by ScanPoint (6.72% incorrect, 121 errors).

Without Bonferroni adjustment, t-tests reveal a statistically significant difference between the error percentage for TargetPoint and EyePoint with $p < 0.05$ ($p = 0.03$). With Bonferroni adjustment the required $p$ threshold is $p < 0.0125$ which means that the difference is *not* statistically significant when considered within the context of the other techniques evaluated.

The accuracy percentages for each technique are thus 99.39% for the mouse (SD\(^1\) 1.67), 96.72% (SD 2.92) for TargetPoint, 93.94% (SD 7.5) for EyePoint, 93.28% (SD 4.8) for ScanPoint, and 84.11% (SD 16.3) for StaggerPoint.

---

\(^1\) SD: Standard Deviation
Chapter 7: Experimental Results

The number of early and late trigger errors in TargetPoint and EyePoint were of particular interest. TargetPoint was designed to use a visual feedback highlight as a method of combating early trigger errors whereas EyePoint used trigger point time shifting.

In order to compare the two approaches, the sources of errors in TargetPoint and EyePoint were analysed. Selection errors were classified into four categories: early trigger errors (the participant releases the trigger key before looking at the target), late trigger errors (trigger key released after looking away from the target), popup triggering errors (pop-up activated when the user isn't looking close enough to the target) and accuracy errors (reported gaze point does not fall within the target area).

The number of early trigger errors obtained using EyePoint can be reduced by shifting the selection trigger point to 80 milliseconds after a selection is triggered (Figure 7.2: Mean Error Percentage (Accuracy Condition)).
3.13. As the “magnified technique” is an implementation of EyePoint, it was decided to use this optimisation for the implementation used in the experiment, the actual shift was 83 milliseconds due to the tracker sampling rate of 60Hz. The effect of removing this optimisation was measured through simulation using the data captured by the framework. It was determined that shifting the trigger point resulted in a decrease in the number of early trigger errors (Figure 7.2 and Figure 7.3) for EyePoint. T-tests reveal this difference to be statistically significant with a Cohen's D value of 0.83 implying a large difference. This shift decreased the number of early trigger errors, but also increased the number of late trigger errors by a smaller margin. The net result is a decrease in the total number of errors.

![Selection Errors by Type (All Conditions)](image)

*Figure 7.2: Selection Errors by Type (all test conditions)*

TargetPoint has a smaller number of early, late and accuracy errors than EyePoint (Figure 7.2). During the design of TargetPoint it was hypothesised that the feedback provided by the visual highlight would result in a decrease in the number of early
trigger errors (Section 4.3.1). T-tests reveal that this decrease is statistically significant when compared to the number of early trigger errors for EyePoint ($p < 0.0125^1$). A Cohen's D value of 0.71 implies a moderate difference. The increase in late trigger errors for EyePoint resulting from the trigger point time shift is also statistically significant ($p < 0.125$). A Cohen's D value of 0.5 implies a moderate difference. Given this information, it is reasonable to argue that providing a feedback highlight is a more effective method of reducing early trigger errors, as it does not result in an increase in late trigger errors (unlike trigger point time shifting). Note that a trigger point time shift of 5 samples for TargetPoint would have resulted in an increase in selection errors (Figure 7.3), vindicating the design decision not to perform trigger point time shifting for TargetPoint. This decision was based on the expectation that visual feedback would reduce the number of early trigger errors.

![Selection Errors versus Trigger Time Shift](image)

Figure 7.3: Trigger Errors versus Trigger Time Shift

One of the advantages of target expansion in motor space only is that it draws the users' gaze closer to the centre of the target (Section 4.3.1). This benefit was confirmed by the data. The average distance (in pixels) from the centre of the target for correct selections was measured (Figure 7.4).

On average, the user's gaze was 40 pixels from the target centre with EyePoint, compared to 16 pixels with TargetPoint. T-tests reveal that this difference is

---

1 Actual value: $p = 0.000521$
statistically significant \( (p < 0.0125) \), with a Cohen's D value of 2.28 implying a large difference. With visual magnification the users' gaze is therefore closer to the boundaries of the target. This increases the chances that inaccurate tracking data will place the reported gaze location outside of the intended target area. The number of accuracy errors was 31\% lower for TargetPoint compared to EyePoint (Figure 7.2), although t-tests do not indicate any statistical significance in this difference \( (p < 0.0125) \).

In order to measure the effectiveness of the target disambiguation mechanism provided by StaggerPoint, a simulation was performed. The number of selection errors obtained using the target resolution logic described in Section 4.3.3 was compared to the number of errors obtained when interpreting the gaze points normally (Figure 7.5). The error reducing effect of the staggered gaze interpretation is clear. The average percentage of incorrect selections was more than halved from 43\% to 20\%. T-tests

\[ \text{Actual value: } p < 0.000001 \]
indicate that this difference is statistically significant ($p < 0.0125$). A Cohen's D value of 1.48 implies a large difference.
The reduction in selection errors due to ScanPoint was also measured using simulation (Figure 7.6). Without the timed highlight correction, the item which would have been selected was the item the user was looking at when the trigger key was pressed. The effect of the timed highlight correction on the percentage of incorrect selections is significant. With timed highlight correction, the percentage of incorrect selections across all conditions drops from 46% to 9%. This reduction is more significant than the reduction which was achieved using staggered gaze interpretation. T-tests reveal that the reduction in errors due to timed highlight correction is significant ($p < 0.0125$). A Cohen's D value of 2.21 implies a large difference.

### 7.3.2 Efficiency

The average time to perform a selection (incorrect or not) using the different techniques is depicted in Figure 7.7. The values reported are for the speed condition (users were instructed to “*try to select the targets quickly*”, Section 6.4.3). On average, the mouse and StaggerPoint were the fastest, averaging around 840 milliseconds per selection. It should be noted that selection times include a search task – locating the target which has been marked for selection.

![Mean Selection Times (Speed Condition)](image-url)
T-tests do not reveal any statistically significant differences between StaggerPoint and the mouse \((p < 0.0125)\) (Table 7.3). The magnified, spaced and timed techniques all took around 1480 milliseconds per selection. T-tests do not reveal any statistically significant differences between these three techniques \((p < 0.0125)\). The differences between all of the combinations involving the mouse and the three slower gaze techniques are statistically significant \((p < 0.0125)\), with Cohen's D values indicating a large difference in all cases. Similarly, t-tests also reveal the speed advantage of StaggerPoint to be statistically significant when paired with each of the other gaze selection techniques \((p < 0.0125)\). Cohen's D values exceeding 2.1 indicate a large difference for all combinations.

![Table 7.3: Mean Selection Times in Milliseconds (Speed Condition)](image)

Underlining indicates result in favour of item on vertical axis.

Bold print Denotes statistical significance \((p < 0.0125)\)

Mean values reported correspond to items on vertical axis.

D : Cohen's D, a measure of practical significance (only reported for statistically significant results).

Efficiency is measured as the number of correct selections per minute (Section 6.4.4). The selection rate (efficiency) of each technique is depicted in Figure 7.8. Selection
using the mouse was the most efficient technique, with an average of 70.1 selections per minute (Table 7.4).

T-tests indicate that the gaze selection techniques were all significantly less efficient than the mouse (p < 0.0125). Cohen's D values indicate that the differences are large, except in the case of StaggerPoint versus the mouse where a Cohen's D value of 0.58 indicates a moderate difference (Table 7.4). T-tests also indicate that StaggerPoint was significantly more efficient than all of the other gaze selection techniques (p < 0.0125). Cohen's D values exceeding 1.16 in all cases indicate large differences. StaggerPoint was 48% more efficient than EyePoint (the next most efficient gaze selection technique). T-tests do not reveal any statistically significant differences between the three non-staggered gaze selection techniques.
### Table 7.4: Mean Efficiency (Correct Selections per Minute) (Speed Condition)

<table>
<thead>
<tr>
<th>Target Point (38.8)</th>
<th>EyePoint (39.7)</th>
<th>Stagger Point (58.6)</th>
<th>ScanPoint (38)</th>
<th>Mouse (70.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPoint (38.8)</td>
<td>N/A</td>
<td>p=0.57</td>
<td>p=0.000000</td>
<td>p=0.000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D=1.3</td>
<td>D=3.2</td>
</tr>
<tr>
<td>EyePoint (39.7)</td>
<td></td>
<td>p=0.57</td>
<td>N/A</td>
<td>p=0.000001</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>p=0.33</td>
<td>p=0.000001</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D=1.16</td>
<td>D=3.09</td>
</tr>
<tr>
<td>StaggerPoint (58.6)</td>
<td></td>
<td>p=0.000000</td>
<td>N/A</td>
<td>p=0.000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D=1.3</td>
<td>p=0.003682</td>
<td>p=0.000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D=1.16</td>
<td>D=0.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>N/A</td>
<td>D=3.3</td>
</tr>
<tr>
<td>ScanPoint (38)</td>
<td></td>
<td>p=0.38</td>
<td>p=0.000000</td>
<td>p=0.000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D=3.2</td>
<td>p=0.003682</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>D=3.09</td>
<td></td>
</tr>
<tr>
<td>Mouse (70.1)</td>
<td></td>
<td>p=0.000000</td>
<td>p=0.000000</td>
<td>p=0.000000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>D=3.2</td>
<td>p=0.003682</td>
<td>D=3.3</td>
</tr>
</tbody>
</table>

Underlining indicates result in favour of item on vertical axis.

Bold print Denotes statistical significance (p < 0.0125)

Mean values reported correspond to items on vertical axis.

D : Cohen's D, a measure of practical significance (only reported for statistically significant results).

---

### 7.4 Self Reported Metrics

Data provided by test participants in the post-test questionnaire (Appendix B) is presented in this section. A summary of the self reported evaluation results is provided (Table 7.5). Perceived effectiveness (accuracy in the context of a gaze selection technique, Section 7.4.1), perceived efficiency (Section 7.4.2) and satisfaction (Section 7.4.3) results are presented. Results which relate to the system recorded metrics are briefly discussed, although most of the discussion is deferred to Sections 7.5 and 7.6.
Table 7.5: Self Reported Metrics (5 point Likert scale) (Standard Deviation Values in Parentheses)

<table>
<thead>
<tr>
<th></th>
<th>EyePoint</th>
<th>Target Point</th>
<th>Mouse</th>
<th>Stagger Point</th>
<th>Scan Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceived Accuracy</td>
<td>4.43</td>
<td>4.53</td>
<td>4.83</td>
<td>3.37</td>
<td>3.43</td>
</tr>
<tr>
<td>(higher is preferred)</td>
<td>(0.73)</td>
<td>(0.73)</td>
<td>(0.38)</td>
<td>(1.16)</td>
<td>(0.97)</td>
</tr>
<tr>
<td>Perceived Accuracy</td>
<td>2.90</td>
<td>2.23</td>
<td>2.87</td>
<td>3.93</td>
<td>3.67</td>
</tr>
<tr>
<td>Decline for Smaller</td>
<td>(1.16)</td>
<td>(1.01)</td>
<td>(1.28)</td>
<td>(1.11)</td>
<td>(1.06)</td>
</tr>
<tr>
<td>Targets (lower is preferred)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perceived Efficiency</td>
<td>4.00</td>
<td>4.17</td>
<td>4.30</td>
<td>3.27</td>
<td>3.00</td>
</tr>
<tr>
<td>(higher is preferred)</td>
<td>(1.05)</td>
<td>(0.91)</td>
<td>(0.92)</td>
<td>(1.2)</td>
<td>(1.08)</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>4.43</td>
<td>4.43</td>
<td>4.73</td>
<td>3.63</td>
<td>3.37</td>
</tr>
<tr>
<td>(higher is preferred)</td>
<td>(0.94)</td>
<td>(0.77)</td>
<td>(0.52)</td>
<td>(1.03)</td>
<td>(1.07)</td>
</tr>
<tr>
<td>Satisfaction</td>
<td>4.23</td>
<td>4.33</td>
<td>4.47</td>
<td>3.10</td>
<td>3.07</td>
</tr>
<tr>
<td>(higher is preferred)</td>
<td>(0.94)</td>
<td>(0.8)</td>
<td>(0.63)</td>
<td>(1.21)</td>
<td>(1.08)</td>
</tr>
<tr>
<td>Perceived Mental Effort</td>
<td>2.53</td>
<td>2.37</td>
<td>1.90</td>
<td>2.97</td>
<td>2.97</td>
</tr>
<tr>
<td>(lower is preferred)</td>
<td>(1.01)</td>
<td>(1.1)</td>
<td>(1.03)</td>
<td>(1.35)</td>
<td>(1.19)</td>
</tr>
<tr>
<td>Perceived Physical Effort</td>
<td>2.40</td>
<td>2.37</td>
<td>2.93</td>
<td>2.77</td>
<td>2.80</td>
</tr>
<tr>
<td>(lower is preferred)</td>
<td>(1.1)</td>
<td>(1.16)</td>
<td>(1.28)</td>
<td>(1.3)</td>
<td>(1.24)</td>
</tr>
</tbody>
</table>

7.4.1 Perceived Effectiveness

Perceived effectiveness relates to accuracy. The mouse was perceived to be the most accurate technique (Table 7.5, Figure 7.9) with an average rating of 4.83 out of 5 (5 point Likert scale), followed by TargetPoint (4.53) and then EyePoint (4.43). ScanPoint and StaggerPoint were perceived as the least accurate. The ordering of the perceived accuracy values corresponds to the ordering of the actual accuracy values recorded during testing (Figure 7.1). It is interesting to note that the ScanPoint technique was perceived to be significantly less accurate than EyePoint, despite the fact that the actual accuracy difference in testing was small (Table 7.2).

T-tests indicate that the differences between the perceived accuracy values for ScanPoint and each of the three techniques perceived as the most accurate (mouse, TargetPoint, EyePoint) are statistically significant (p < 0.0125) (Table 7.6). The same result is obtained for t-tests on StaggerPoint when compared to the three techniques.
perceived as the most accurate. Differences between TargetPoint and EyePoint, and TargetPoint and mouse were not found to be statistically significant using t-tests (p < 0.0125). T-tests reveal the difference between the mouse and EyePoint to be statistically significant (p < 0.0125). A Cohen's D value of 0.64 indicates a moderate difference.

<table>
<thead>
<tr>
<th></th>
<th>Target Point (4.53)</th>
<th>EyePoint (4.43)</th>
<th>Stagger Point (3.37)</th>
<th>ScanPoint (3.43)</th>
<th>Mouse (4.83)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPoint (4.53)</td>
<td>N/A</td>
<td>p=0.59</td>
<td>p=0.00008</td>
<td>p=0.00001</td>
<td>p=0.026</td>
</tr>
<tr>
<td>EyePoint (4.43)</td>
<td>p=0.59</td>
<td>N/A</td>
<td>p=0.00076</td>
<td>p=0.00015</td>
<td>p=0.00143</td>
</tr>
<tr>
<td>StaggerPoint (3.37)</td>
<td>p=0.00008</td>
<td>p=0.00076</td>
<td>N/A</td>
<td>p=0.00001</td>
<td>p=0.000001</td>
</tr>
<tr>
<td>ScanPoint (3.43)</td>
<td>p=0.00001</td>
<td>p=0.00015</td>
<td>p=0.8</td>
<td>N/A</td>
<td>p=0.000001</td>
</tr>
<tr>
<td>Mouse (4.83)</td>
<td>p=0.026</td>
<td>p=0.00143</td>
<td>p=0.00001</td>
<td>p=0.000001</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Underlining indicates result in favour of item on vertical axis.
Bold print Denotes statistical significance (p < 0.0125)
Mean values reported correspond to items on vertical axis.

D : Cohen's D, a measure of practical significance (only reported for statistically significant results).

Participants were also required to indicate their perceptions regarding the decline in accuracy for the smaller target size for each technique. TargetPoint was perceived to have been the least affected by the decline in target size (2.23), while StaggerPoint (2.9) was regarded as having been the most affected by reducing the target size (Table 7.5). T-tests only indicate statistically significant differences in pairs containing either StaggerPoint or ScanPoint (p < 0.0125), but not both (Table 7.7).
Chapter 7: Experimental Results

**Table 7.7: Average Ratings for Perceived Accuracy Decline for Smaller Targets**
*(5 Point Likert Scale, Lower Values Preferred)*

<table>
<thead>
<tr>
<th>Technique</th>
<th>Target Point (2.23)</th>
<th>EyePoint (2.9)</th>
<th>Stagger Point (3.93)</th>
<th>ScanPoint (3.67)</th>
<th>Mouse (2.87)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPoint</td>
<td>N/A</td>
<td>p=0.017</td>
<td>p=0.000000 D=1.27</td>
<td>p=0.000001 D=1.17</td>
<td>p=0.037</td>
</tr>
<tr>
<td>EyePoint</td>
<td>p=0.017</td>
<td>N/A</td>
<td>p=0.00084 D=0.68</td>
<td>p=0.00181 D=0.63</td>
<td>p=0.897</td>
</tr>
<tr>
<td>StaggerPoint</td>
<td>p=0.000000 D=1.27</td>
<td>p=0.00084 D=0.68</td>
<td>N/A</td>
<td>p=0.23</td>
<td>p=0.00087 D=0.68</td>
</tr>
<tr>
<td>ScanPoint</td>
<td>p=0.000001 D=1.17</td>
<td>p=0.00181 D=0.63</td>
<td>p=0.23</td>
<td>N/A</td>
<td>p=0.00727 D=0.53</td>
</tr>
<tr>
<td>Mouse</td>
<td>p=0.037</td>
<td>p=0.897</td>
<td>p=0.00087 D=0.68</td>
<td>p=0.00727 D=0.53</td>
<td>N/A</td>
</tr>
</tbody>
</table>

_Underlining_ indicates result in favour of item on vertical axis.

_Bold print_ Denotes statistical significance (p < 0.0125)

Mean values reported correspond to items on vertical axis.

D : Cohen's D, a measure of practical significance (only reported for statistically significant results).
7.4.2 Perceived Efficiency

The mouse was perceived to be the most efficient technique (4.3) (Figure 7.10, Table 7.5), followed by TargetPoint (4.17) and EyePoint (4.0). ScanPoint was perceived to be the least efficient technique (3.0). It is interesting to note that StaggerPoint was perceived to be the second least efficient technique (3.27), despite being significantly more efficient than all of the other gaze selection techniques (Figure 7.8).

As with the perceived accuracy, t-tests indicate that the differences between ScanPoint and the three techniques rated highest (mouse, TargetPoint, EyePoint) are all statistically significant (p < 0.0125) (Table 7.8). Differences between StaggerPoint and TargetPoint, and StaggerPoint and the mouse were found to be statistically significant with t-tests (p < 0.0125). None of the differences between the other combinations were found to be statistically significant. Cohen's D values for all cases where statistical significance was found indicate moderate differences, except for the differences between the mouse and ScanPoint, and TargetPoint and ScanPoint, where Cohen's D values indicate large differences.

![Perceived Efficiency](image)
One interesting observation regarding the mean ratings for perceived efficiency of the techniques, is that the differences between the mouse and the two pop-up techniques (TargetPoint, EyePoint) are small. This is particularly noteworthy as the actual efficiency data reveals a large difference, with the mouse significantly outperforming TargetPoint and EyePoint by at least 75% (Figure 7.8). The small differences between the ratings may be attributed to users perceiving techniques involving gaze to be more efficient than they actually are. This result confirms similar findings (Zhai et al. 1999).

Eight participants commented on the mouse being slow or less efficient than the gaze techniques, while only three participants commented on it being quicker. One example of a comment regarding perceived efficiency is: “These [gaze techniques] were a lot faster. Mouse was too much effort and monotonous.”. It is interesting to note that this comment came from a test participant for whom the mouse outperformed the nearest gaze technique by a factor of 2.4.

<table>
<thead>
<tr>
<th></th>
<th>TargetPoint (4.16)</th>
<th>EyePoint (4)</th>
<th>StaggerPoint (3.27)</th>
<th>ScanPoint (3)</th>
<th>Mouse (4.3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPoint (4.16)</td>
<td>N/A</td>
<td>p=0.47</td>
<td><strong>p=0.00257</strong> D=0.6</td>
<td><strong>p=0.000002</strong> D=1.07</td>
<td>p=0.57</td>
</tr>
<tr>
<td>EyePoint (4)</td>
<td>p=0.47</td>
<td>N/A</td>
<td>p=0.035</td>
<td><strong>p=0.002</strong> D=0.62</td>
<td>p=0.24</td>
</tr>
<tr>
<td>StaggerPoint (3.27)</td>
<td><strong>p=0.00257</strong> D=0.6</td>
<td>p=0.035</td>
<td>N/A</td>
<td><strong>p=0.00279</strong> D=0.6</td>
<td>p=0.31</td>
</tr>
<tr>
<td>ScanPoint (3)</td>
<td><strong>p=0.000002</strong> D=1.07</td>
<td><strong>p=0.002</strong> D=0.62</td>
<td>p=0.31</td>
<td>N/A</td>
<td><strong>p=0.00003</strong> D=0.9</td>
</tr>
<tr>
<td>Mouse (4.3)</td>
<td>p=0.57</td>
<td>p=0.24</td>
<td><strong>p=0.00279</strong> D=0.6</td>
<td><strong>p=0.00003</strong> D=0.9</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Underlining indicates result in favour of item on vertical axis.

Bold print Denotes statistical significance (p < 0.0125)

Mean values reported correspond to items on vertical axis.

D : Cohen's D, a measure of practical significance (only reported for statistically significant results).

Table 7.8: Average Perceived Efficiency Ratings
(5 Point Likert Scale, Higher Values Preferred)
Chapter 7: Experimental Results

A second observation is that StaggerPoint was perceived to be the second slowest technique, despite the fact that the data reveals that this technique was the most efficient gaze selection technique evaluated by a significant margin (Figure 7.8). Possible explanations for this discrepancy include users' perceptions that this technique required the most mental effort (Table 7.5, Section 7.4.3), and the fact that this technique was (correctly) perceived to be the least accurate of the gaze selection techniques (assuming users considered correctness of selections to be an element of efficiency).

7.4.3 Satisfaction
The mouse was perceived as the easiest technique to use (4.73) (Figure 7.11, Table 7.5). TargetPoint and EyePoint were rated equally (4.43). ScanPoint was considered the most difficult to use (3.37), followed by StaggerPoint (3.63).

T-tests do not indicate any statistically significant differences between the mouse, TargetPoint and EyePoint for ease of use (p < 0.0125) (Table 7.9). For convenience of reporting, these three techniques will be named group A. The difference between StaggerPoint and ScanPoint was not found to be significant. These two techniques
will be named group B. For each pair combining any element from group A with any element from group B, the difference is statistically significant according to t-tests ($p < 0.0125$). Cohen's D values indicate that the differences involving the mouse are all large. Differences involving EyePoint are moderate. The difference between TargetPoint and ScanPoint is large, while the difference between TargetPoint and StaggerPoint is moderate.

<table>
<thead>
<tr>
<th></th>
<th>Target Point (4.43)</th>
<th>EyePoint (4.43)</th>
<th>Stagger Point (3.63)</th>
<th>ScanPoint (3.37)</th>
<th>Mouse (4.73)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPoint</td>
<td>N/A</td>
<td>p=1.00</td>
<td><strong>p=0.00143</strong> D=0.64</td>
<td><strong>p=0.000008</strong> D=0.99</td>
<td>p=0.037</td>
</tr>
<tr>
<td>EyePoint</td>
<td>p=1.00</td>
<td>N/A</td>
<td><strong>p=0.00727</strong> D=0.53</td>
<td><strong>p=0.00113</strong> D=0.66</td>
<td>p=0.12</td>
</tr>
<tr>
<td>StaggerPoint</td>
<td><strong>p=0.00143</strong> D=0.64</td>
<td><strong>p=0.00727</strong> D=0.53</td>
<td>N/A</td>
<td><strong>p=0.00001</strong> D=0.95</td>
<td></td>
</tr>
<tr>
<td>ScanPoint</td>
<td><strong>p=0.000008</strong> D=0.99</td>
<td><strong>p=0.00113</strong> D=0.66</td>
<td>p=0.28</td>
<td><strong>p=0.000002</strong> D=1.10</td>
<td></td>
</tr>
<tr>
<td>Mouse</td>
<td>p=0.037</td>
<td>p=0.12</td>
<td><strong>p=0.00001</strong> D=0.95</td>
<td><strong>p=0.000002</strong> D=1.10</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Underlining** indicates result in favour of item on vertical axis.

**Bold print** Denotes statistical significance ($p < 0.0125$)

Mean values reported correspond to items on vertical axis.

D : Cohen's D, a measure of practical significance (only reported for statistically significant results).

Table 7.9: Average Ease of Use Ratings (5 Point Likert Scale, Higher Values Preferred)

T-tests indicate that the differences between pairs including an element from group A (EyePoint, TargetPoint, mouse) and an element from group B (StaggerPoint, ScanPoint) are all statistically significant for satisfaction ($p < 0.0125$) (Table 7.10). Cohen's D values indicate that the differences are large for all pairs, except those which include EyePoint.

The mouse was rated highest for overall satisfaction (4.47), followed by TargetPoint (4.33) and EyePoint (4.23) (Figure 7.12, Table 7.5). Satisfaction was lowest for ScanPoint (3.07), followed by StaggerPoint (3.1).
Chapter 7: Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Target Point (4.33)</th>
<th>EyePoint (4.23)</th>
<th>Stagger Point (3.1)</th>
<th>ScanPoint (3.07)</th>
<th>Mouse (4.47)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPoint</td>
<td>N/A</td>
<td>p=0.6</td>
<td>p=0.00015</td>
<td>p=0.000002</td>
<td>p=0.35</td>
</tr>
<tr>
<td>(4.33)</td>
<td></td>
<td></td>
<td>D=0.8</td>
<td>D=1.08</td>
<td></td>
</tr>
<tr>
<td>EyePoint</td>
<td>p=0.6</td>
<td>N/A</td>
<td>p=0.00288</td>
<td>p=0.00047</td>
<td>p=0.2</td>
</tr>
<tr>
<td>(4.23)</td>
<td></td>
<td></td>
<td>D=0.59</td>
<td>D=0.72</td>
<td></td>
</tr>
<tr>
<td>StaggerPoint</td>
<td>p=0.00015</td>
<td>p=0.00288</td>
<td>N/A</td>
<td></td>
<td>p=0.00004</td>
</tr>
<tr>
<td>(3.1)</td>
<td>D=0.8</td>
<td>D=0.59</td>
<td></td>
<td></td>
<td>D=0.89</td>
</tr>
<tr>
<td>ScanPoint</td>
<td>p=0.000002</td>
<td>p=0.00047</td>
<td>p=0.89</td>
<td>N/A</td>
<td>p=0.000000</td>
</tr>
<tr>
<td>(3.07)</td>
<td>D=1.08</td>
<td>D=0.72</td>
<td></td>
<td></td>
<td>D=1.24</td>
</tr>
<tr>
<td>Mouse</td>
<td>p=0.35</td>
<td>p=0.2</td>
<td>p=0.00004</td>
<td>p=0.000000</td>
<td>N/A</td>
</tr>
<tr>
<td>(4.47)</td>
<td></td>
<td></td>
<td>D=0.89</td>
<td>D=1.24</td>
<td></td>
</tr>
</tbody>
</table>

**Underlining** indicates result in favour of item on vertical axis.

**Bold print** Denotes statistical significance (p < 0.0125)

Mean values reported correspond to items on vertical axis.

**D**: Cohen's D, a measure of practical significance (only reported for statistically significant results).

Table 7.10: Average Satisfaction Ratings (5 Point Likert Scale, Higher Values Preferred)

![Satisfaction](image)

**Figure 7.12**: Satisfaction
(95% Confidence Intervals Indicated)

142
The comparatively low satisfaction ratings for StaggerPoint and ScanPoint can be attributed to issues raised by users in the qualitative feedback section (Section 7.5). These issues included mental fatigue, accuracy and aesthetic issues on the part of StaggerPoint, and general frustration with the timing issues associated with the timed highlight of ScanPoint.

When asked to rank the techniques in order of preference, 70% \((n=21)\) of the participants ranked TargetPoint ahead of EyePoint. A Wilcoxon rank sum test indicates that this difference is statistically significant \((p < 0.0125)\) (Table 7.11). A Cohen's D value of 0.53 implies that the difference is moderate. Of the ten possible paired combinations, the differences in all but three pairs are statistically significant according to a Wilcoxon rank sum test \((p < 0.0125)\). The three pairs for which the differences were not found to be statistically significant are the mouse and TargetPoint, the mouse and EyePoint, and StaggerPoint and ScanPoint.

Exactly half of the test participants ranked the mouse as their favourite technique \((n=15)\) (Figure 7.13). The remaining first place rankings were divided between TargetPoint which was ranked first eight times, EyePoint (five times), and StaggerPoint (twice). The mouse was ranked ahead of EyePoint by 77% of the participants \((n=23)\). It was also ranked ahead of TargetPoint by 57% \((n=17)\) of the test participants.
TargetPoint was ranked in second place most often (15 times), followed by first place (8 times). The frequency distribution for EyePoint peaks in 3rd place \( (n=12) \). StaggerPoint and ScanPoint were the least popular. StaggerPoint was ranked last most often \( (n=13) \), and ScanPoint was ranked second last most often \( (n=14) \).

<table>
<thead>
<tr>
<th></th>
<th>TargetPoint ( (2.03) )</th>
<th>EyePoint ( (2.83) )</th>
<th>StaggerPoint ( (4) )</th>
<th>ScanPoint ( (4.13) )</th>
<th>Mouse ( (2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPoint ( (2.03) )</td>
<td>N/A</td>
<td>( p=0.008 ) D=0.56</td>
<td>( p=0.000 ) D=1.11</td>
<td>( p=0.000 ) D=1.77</td>
<td>( p=0.75 )</td>
</tr>
<tr>
<td>EyePoint ( (2.83) )</td>
<td>( p=0.008 ) D=0.56</td>
<td>N/A</td>
<td>( p=0.007 ) D=0.56</td>
<td>( p=0.001 ) D=0.74</td>
<td>( p=0.064 )</td>
</tr>
<tr>
<td>StaggerPoint ( (4) )</td>
<td>( p=0.000 ) D=1.11</td>
<td>( p=0.007 ) D=0.56</td>
<td>N/A</td>
<td>( p=0.67 )</td>
<td>( p=0.000 ) D=1.07</td>
</tr>
<tr>
<td>ScanPoint ( (4.13) )</td>
<td>( p=0.000 ) D=1.77</td>
<td>( p=0.001 ) D=0.74</td>
<td>( p=0.67 ) N/A</td>
<td>( p=0.000 ) D=1.26</td>
<td>( 0.000 ) D=1.26</td>
</tr>
<tr>
<td>Mouse ( (2) )</td>
<td>( p=0.75 )</td>
<td>( p=0.064 )</td>
<td>( p=0.000 ) D=1.07</td>
<td>( p=0.000 ) D=1.26</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*Underlining* indicates result in favour of item on vertical axis.

*Bold print* Denotes statistical significance \( (p < 0.0125) \)

Mean values reported correspond to items on vertical axis.

\( D \) : Cohen's D, a measure of practical significance (only reported for statistically significant results).

Table 7.11: Average Preference Rankings (Rankings from 1 to 5, lower values preferred)

The popularity of the mouse is surprising in light of previous research. In a previous study EyePoint was preferred over the mouse by three quarters of the test participants (Kumar et al. 2007). In the current evaluation this result is reversed, *with more than three quarters (77%, \( n=23 \)) of the test participants preferring the mouse to EyePoint*. User preference for the mouse over TargetPoint is not as clear, with seventeen test participants (57%) favouring the mouse. This result is consistent with the fact that the mouse was rated the best in five of the seven self reported metrics (accuracy, efficiency, ease of use, satisfaction, mental fatigue) (Table 7.5). As noted in Sections 7.3.1 and 7.3.2, the mouse was the most accurate and efficient technique tested.
Chapter 7: Experimental Results

The fact that the majority of test participants (70%) preferred TargetPoint to EyePoint is consistent with the mean values for the self reported metrics (Table 7.5), and the qualitative feedback (Section 7.5). On average, TargetPoint was perceived to be more accurate, more efficient, more satisfying, require less mental and physical effort, and suffer a smaller accuracy decline for smaller targets than EyePoint. Both techniques were rated equally for ease of use. In most cases these perceived values are consistent with the actual data. TargetPoint was more accurate than EyePoint (Figure 7.1, Figure 7.2), and was not affected as much by reducing the size of the targets. The only exception is the perceived efficiency means, in which the ordering of the techniques differs from the actual efficiency results as measured in the speed test (Figure 7.8). These values do however correspond to the efficiency means when the data from all the tests (not just the speed tests) are taken into consideration, in which case TargetPoint was more efficient than EyePoint. The discrepancy between perceived and actual efficiency may also be related to the selection times for TargetPoint being lower than those of EyePoint (Figure 7.7).

![Perceived Mental Effort](image)

*Figure 7.14: Perceived Mental Effort (lower is preferred)
(95% confidence interval indicated)*

Participants perceived the mouse to require the least mental effort (1.9) (Figure 7.14, Table 7.5), followed by TargetPoint (2.37), and EyePoint (2.53). ScanPoint (2.97) and
StaggerPoint (2.97) were perceived to require the most mental effort. Qualitative feedback regarding these two techniques provides some insight into this rating (Sections 7.5.4 and 7.5.5).

T-tests do not reveal a statistically significant difference between TargetPoint and EyePoint ($p < 0.0125$) (Table 7.12). Most of the statistically significant differences are between the mouse and the other techniques. The differences between the mouse and all of the other techniques (except TargetPoint) are statistically significant according to t-tests ($p < 0.0125$).

<table>
<thead>
<tr>
<th></th>
<th>TargetPoint (2.37)</th>
<th>EyePoint (2.53)</th>
<th>StaggerPoint (2.97)</th>
<th>ScanPoint (2.97)</th>
<th>Mouse (1.9)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPoint (2.37)</td>
<td>N/A</td>
<td>$p=0.46$</td>
<td>$p=0.05$</td>
<td>$p=0.00998$</td>
<td>$D=0.5$</td>
</tr>
<tr>
<td>EyePoint (2.53)</td>
<td>$p=0.46$</td>
<td>N/A</td>
<td>$p=0.16$</td>
<td>$p=0.11$</td>
<td>$p=0.00931$</td>
</tr>
<tr>
<td>StaggerPoint (2.97)</td>
<td>$p=0.05$</td>
<td>$p=0.16$</td>
<td>N/A</td>
<td>$p=1$</td>
<td>$p=0.00048$</td>
</tr>
<tr>
<td>ScanPoint (2.97)</td>
<td>$p=0.00998$</td>
<td>$p=0.11$</td>
<td>$p=1$</td>
<td>N/A</td>
<td>$p=0.00048$</td>
</tr>
<tr>
<td>Mouse (1.9)</td>
<td>$p=0.06$</td>
<td>$p=0.00931$</td>
<td>$p=0.00048$</td>
<td>$p=0.00048$</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Underlining indicates result in favour of item on vertical axis.

Bold print Denotes statistical significance ($p < 0.0125$)

Mean values reported correspond to items on vertical axis.

$D$ : Cohen's D, a measure of practical significance (only reported for statistically significant results).

TargetPoint was perceived to require the least physical effort (2.37), followed closely by EyePoint (Figure 7.15, Table 7.5). A mean rating of 2.93 indicates that users perceived the mouse to require the most physical effort. T-tests, however, do not reveal any statistically significant difference between any of the combinations ($p < 0.0125$) (Table 7.13).
### Chapter 7: Experimental Results

#### Table 7.13: Average Perceived Physical Effort Ratings

(5 Point Likert Scale, Lower Values Preferred)

<table>
<thead>
<tr>
<th>Technique</th>
<th>Target Point (2.37)</th>
<th>EyePoint (2.4)</th>
<th>Stagger Point (2.77)</th>
<th>ScanPoint (2.8)</th>
<th>Mouse (2.93)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetPoint</td>
<td>N/A</td>
<td>p=0.83</td>
<td>p=0.11</td>
<td>p=0.03</td>
<td>p=0.06</td>
</tr>
<tr>
<td>EyePoint</td>
<td>p=0.83</td>
<td>N/A</td>
<td>p=0.09</td>
<td>p=0.016</td>
<td>p=0.1</td>
</tr>
<tr>
<td>StaggerPoint</td>
<td>p=0.11</td>
<td>p=0.09</td>
<td>N/A</td>
<td>p=0.86</td>
<td>p=0.63</td>
</tr>
<tr>
<td>ScanPoint</td>
<td>p=0.03</td>
<td>p=0.016</td>
<td>p=0.86</td>
<td>N/A</td>
<td>p=0.68</td>
</tr>
<tr>
<td>Mouse</td>
<td>p=0.06</td>
<td>p=0.1</td>
<td>p=0.63</td>
<td>p=0.68</td>
<td>N/A</td>
</tr>
</tbody>
</table>

*Underlining* indicates result in favour of item on vertical axis.

**Bold print** Denotes statistical significance (p < 0.0125)

Mean values reported correspond to items on vertical axis.

**D**: Cohen's D, a measure of practical significance (only reported for statistically significant results).

![Perceived Physical Effort](image)

*Figure 7.15: Perceived Physical Effort (lower is preferred) (95% confidence interval indicated)*
Chapter 7: Experimental Results

7.4.4 Usefulness of Visual Feedback
Twenty nine (97%) of the thirty test participants found the feedback highlight provided by TargetPoint to be useful, with nineteen of them providing a rating of five ("strongly agree"), and ten of them a rating of four. Only one participant did not feel that the highlight was useful and provided a rating of two. This participant felt that the highlighting was superfluous as the manner in which the targets were shifted\(^1\) by the TargetPoint pop-up was sufficient to determine which button was being looked at.

7.5 Qualitative Data
As noted in the introduction to this chapter, qualitative data is particularly important as it helps to provide an understanding of how users perceive the different techniques. It is also useful for identifying specific issues that participants had with different techniques. Aspects of techniques which proved popular with participants can also be identified. Thematic analysis (Ely et al. 1995) was used to analyse the data.

7.5.1 TargetPoint
The feedback for TargetPoint was largely very positive. The spacing of the buttons proved popular with the test participants. Eighteen of the test participants made positive comments about the spacing of the buttons in the pop-up. Examples include:

- “Spacing the choices out, made choosing the right one easier.”
- “I preferred the spaced one because it separated the things”
- “It spreads the buttons out so it is easier for you to concentrate on a particular button”
- “I prefer spaced because it spreads them out so it doesn't confuse the area, because this one [magnified] didn't have any area between the actual buttons”

Only one participant made a negative comment regarding the spacing: “I hated the space”.

In total, seventeen (57%) test participants expressed either positive views regarding the feedback highlight and/or complained that EyePoint lacked this feature. Only one

\(^1\) Figure 4.3
test participant made a negative comment regarding the feedback highlight (that it jumps to the wrong button too easily). This majority view goes against one of the design principles of EyePoint “...our primary design principle of not slaving any visual feedback to eye movements” (Kumar et al. 2007). The latter view has also been expressed by the creators of MAGIC pointing (Zhai et al. 1999).

It should be noted that some of the participants in the pilot study for EyePoint expressed a desire for feedback, but when provided with feedback in the form of a gaze point marker changed their minds (Kumar et al. 2007). The type of feedback highlight used by TargetPoint is different from a gaze point marker, as its design (Sections 4.2.2 and 4.3.1) greatly reduces the risk of a positive feedback loop. With TargetPoint, users only need to be looking within the general area of a target for it to be highlighted correctly, so the highlight is more likely to appear where users expect it to be.

Twelve participants (of 13 who commented) were positive about the feedback provided by the highlight. Examples of such comments include the following:

- “It's the fact that it [TargetPoint] did give you feedback of where you were looking at a particular point, which is the reason the magnified popup was ranked last”
- “Shows selection highlight - gives a chance to correct selection mistakes”, “...you got immediate feedback on the spaced one and not with the magnified one”
- “I like this one. It's good. The blue highlight shows what your focus of attention is on”
- “The highlight was very helpful...”
- “I found the highlight very useful. With the highlight I felt like I could scroll through the options. That was cool. It was like a metaphor for the mouse scroll wheel, that's how I experienced it.”
Chapter 7: Experimental Results

- “This one had very nice feedback, you could see exactly which one would be selected...”.

One benefit noted for the highlight was the ability to make a correction using the feedback provided by the visual highlight.

Seventeen participants commented that TargetPoint was “easy” to use, or made selection of targets “easier”. Comments included:

- “It was just easier”
- “Very easy and friendly”
- “Easy to use and understand”
- “Very easy to select buttons”

Eighteen of the test subjects commented positively on the accuracy of this technique or the ease with which it enabled them to select the correct target. Quotes include:

- “The accuracy rate of the spaced popup was incredibly high. I just couldn't get a selection wrong.”
- “Because of the spaces, it is easier to use and select items regardless of how small the buttons were”.

Overall, there were comparatively few negative comments regarding TargetPoint. The most common issue commented on was that the wrong part of the screen is displayed in the pop-up if the trigger key is pressed when looking at the wrong part of the screen (an issue it shares with EyePoint). Four participants commented on this issue. Three participants commented that TargetPoint required more eye movement compared to the other techniques. Two participants complained that it made the buttons too small, while two others commented that it was more straining on the eyes (twice as many participants claimed that this technique was less straining on the eyes). It is interesting to note that not one participant ranked TargetPoint last (Figure 7.13). EyePoint, however was ranked last four times.
7.5.2 EyePoint

The majority of the feedback regarding EyePoint was also positive. Seventeen test participants made positive comments about the magnification aspect of this technique. Examples included:

- “The bigger buttons are nice”
- “Large buttons easy to focus on”
- “Magnifying the choices made it clearer to see and made the choice easier to make”

Twenty one of the test subjects commented positively on the accuracy of EyePoint or the ease of with which it enabled them to correctly select targets. Examples of comment included:

- “Zooms in on area to help you select buttons more accurately”
- “Easy to select correct button”
- “Very accurate, easy”

Some negative issues were also raised regarding EyePoint. The lack of feedback in particular was viewed in a negative light, with this issue being cited by 13 test participants. Comments included:

- “This one would have been a lot nicer if it had feedback”
- “The highlight provides a confirmation of what you are looking at [for TargetPoint]. You feel like you are in control. With the magnified popup you kind of had to guess.”
- “I’m not sure which button I’m looking at”
- “You cannot view where your eyes are looking, when it pops up, so you are unsure if what you select is accurate”
- “This technique was very vague and the feedback given to the user was lacking”.

151
Participants' feedback regarding EyePoint also included some unexpected negative comments regarding the focus points (grid of dots) overlaid in the pop-up and the semi-transparent background of the pop-up. Previous research has indicated that users prefer having the grid present (Kumar et al. 2007). Five test participants noted a dislike for the grid. Examples of negative comments include:

- “grid noisy and frustrating”
- “didn't like the dots”
- “the dotted patterns could perhaps be somewhat distracting”.

Three participants noted a dislike for the semi-transparent background of the magnified pop-up, as it was distracting. These two groups had one participant in common (one participant who disliked both the focus points and the semi-transparent background). No positive comments were made regarding either the grid or the transparency.

7.5.3 Mouse

Participants' comments regarding the mouse were largely positive. Nineteen participants (63%) made positive comments regarding the mouse in terms of familiarity and ease of use. Six participants cited the feedback provided by the cursor as a reason for liking the mouse. Sample quotes include the following:

- “I know where the mouse cursor is. (always clear what the PC thinks I am looking at.)”
- “Immediate feedback with mouse cursor.”
- “You can be 100% sure the pointer is on the button.”.

The accuracy of the mouse was also a source of positive comments, with twelve participants commenting on the accuracy.

The mouse was, however, rated as the technique which requires the most physical effort (Table 7.5). Fifteen (50%) of the test participants commented on the mouse requiring physical effort. Examples of these comments include:
• “Too physically tiring.”

• “You use double the effort because you use both the eyes and the mouse.” and “Have to use the mouse and 'click' so it takes effort and is not as quick”.

Despite the mouse being ranked first more often than the other techniques, two participants ranked it in last place and one in second last place. The two participants who ranked the mouse last commented on the mouse being slower and requiring more effort to move around:

• “taking time to focus the pointer on the button and moving it around is time-consuming.”

• “compared to all the others all the effort required is irritating and slow”.

7.5.4 StaggerPoint

Overall, participants' feedback for StaggerPoint was mixed. Twelve participants noted a dislike for the staggering of the buttons, primarily for aesthetic and comfort reasons. Examples of comments include:

• “Felt odd having to look to and fro”

• “Button positioning looks a bit messy”

• “The buttons aren't placed in order - not aesthetic. Uncomfortable to work with”

• “Alignment disconcerting at times”

Eight test participants commented that StaggerPoint was simple or easy to use. Examples of such comments include:

• “Easiest technique of them all”

• “Was simple and easy to use”

• “Simple technique to use and very quick”

Opinions about the accuracy of this technique were mixed. Ten participants made negative comments regarding the accuracy of StaggerPoint. Nine participants
commented positively about the accuracy of this technique. One issue, which is consistent with the actual accuracy data recorded (Figure 7.1), is the difficulty of selecting the smaller buttons using this technique. This issue was noted by six participants.

Comments from some participants are consistent with the higher level of mental effort reported for this technique (Figure 7.14). A total of six participants noted that this technique required more concentration or effort, or was tiring. Participants seemed to feel that they had to concentrate harder when having difficulty selecting items correctly, especially for smaller targets.

Despite StaggerPoint being significantly more efficient than the other gaze selection techniques (Figure 7.8), only four test participants (13%) commented that this technique was efficient. Negative issues raised by a small number of participants included the lack of feedback (3 participants), and the lack of an opportunity to correct errors (2 participants). One participant noted that this technique combined speed and accuracy: “It was great using this technique as it allows for speed and accuracy”. Another participant liked this technique “because there wasn't too much stuff jumping around on the screen”.

### 7.5.5 ScanPoint

The majority of the feedback regarding ScanPoint was negative. Half of the test participants (n=15) noted that the delay used for the scanning process (visual highlight shifting from one target to another), was too short. The delay of 600 milliseconds meant that participants had to react very quickly to avoid making a mistake. This issue is the most likely explanation for the higher level of mental effort reported by participants for this technique (Figure 7.14). Examples of comments by participants include:

- “Sometimes not able to react fast enough to highlight”
- “You have to concentrate on releasing CTRL at the exact moment - puts pressure on you”
- “I must be alert and respond quickly”
A second issue related to timing noted by 14 of the test participants, was that having to wait for the highlight to reach the correct button took too long. This issue is the most likely explanation for the lower perceived efficiency for this technique (Figure 7.10). Half of this group (n=7) also commented that the highlight was too quick. Examples of negative comments regarding the time it took the highlight to reach the correct button include:

- "I don't like the fact that you have to wait for an item to be highlighted"
- “Sometimes it took too long to jump to the next option available when the wrong choice was made”
- “That one took too long to do because you had to wait for the highlight”

The difficulties of balancing the highlighting delay so that it isn't too quick, yet does not take too long is perhaps best summed up by a comment by one participant: “If the delay is set longer you have to wait a long time when it is incorrect. If it's too short you may not release the key in time if the first one is correct.”.

Six test participants made positive comments regarding the ability to make a correction with this technique. One participant felt that this technique provided a sense of security as the tracker wasn't that accurate. Negative issues raised included participants disliking the lack of control over the timed highlight (2 participants), and the inconsistent highlight shift sequence (1 participant).

### 7.6 Testing of Hypotheses

The research hypotheses (Section 6.2) were tested using the data gathered during the experiment. The hypotheses are presented as null hypothesis in the following form:

$$H_0 \ (i, \ j) : \text{Technique } i \text{ is not more } \text{usable} \text{ than technique } j, \ i \in \{\text{TargetPoint, StaggerPoint, ScanPoint}\}, \ j \in \{\text{EyePoint, mouse}\}$$

Each hypothesis consists of three sub-components:

$$H_{0.1} \ (i, \ j) : \text{Technique } i \text{ is not more } \text{efficient} \text{ than technique } j,$$

$$i \in \{\text{TargetPoint, StaggerPoint, ScanPoint}\}, \ j \in \{\text{EyePoint, mouse}\}$$
\( H_{0.2}(i, j) : \) Technique \( i \) is not more \textit{effective} than technique \( j \),
\( i \in \{ \text{TargetPoint, StaggerPoint, ScanPoint} \}, j \in \{ \text{EyePoint, mouse} \} \)

\( H_{0.3}(i, j) : \) Technique \( i \) is not more \textit{satisfying} than technique \( j \),
\( i \in \{ \text{TargetPoint, StaggerPoint, ScanPoint} \}, j \in \{ \text{EyePoint, mouse} \} \)

The first set of hypotheses to be considered is \( H_0 \) (i, EyePoint). For each \( i \in \{ \text{TargetPoint, StaggerPoint, ScanPoint} \} \), the lower level hypotheses need to be considered first.

\textbf{7.6.1 TargetPoint versus EyePoint}

None of the differences in any of the \textit{efficiency} metrics between TargetPoint and EyePoint are statistically significant (Table 7.4, Table 7.8). This is \textbf{insufficient to reject the null hypothesis,} \( H_{0.1} \) (TargetPoint, EyePoint), that TargetPoint is not more \textit{efficient} than EyePoint.

When considered within the context of all of the techniques evaluated (Bonferroni adjustment), none of the differences in any of the \textit{effectiveness} metrics between TargetPoint and EyePoint are statistically significant (Table 7.2, Table 7.6, Table 7.7). This evidence is \textbf{insufficient to reject the null hypothesis,} \( H_{0.2} \) (TargetPoint, EyePoint), that TargetPoint is not more \textit{effective} than EyePoint. Note that without Bonferroni adjustment the difference in accuracy between TargetPoint and EyePoint is statistically significant.

The only statistically significant difference between TargetPoint and EyePoint in terms of \textit{satisfaction} is the preference rankings. Overall preference for TargetPoint over EyePoint is statistically significant (Table 7.11). None of the other differences between TargetPoint and EyePoint for any of the other satisfaction metrics is statistically significant (Table 7.9, Table 7.10, Table 7.12, Table 7.13). \textbf{This evidence is sufficient to reject the null hypothesis,} \( H_{0.3} \) (TargetPoint, EyePoint), that TargetPoint is not more \textit{satisfying} than EyePoint.

Given the fact that there are no statistically significant differences which favour EyePoint over TargetPoint and that \( H_{0.3} \) (TargetPoint, EyePoint) was rejected, \textbf{there is}
sufficient evidence to reject the null hypothesis, $H_0 (\text{TargetPoint, EyePoint})$, that TargetPoint is not more usable than EyePoint.

### 7.6.2 StaggerPoint versus EyePoint

The difference in actual efficiency between StaggerPoint and EyePoint is statistically significant in favour of StaggerPoint (Table 7.4). The difference in perceived efficiency between the two techniques is not statistically significant (Table 7.8). This evidence is sufficient to reject the null hypothesis, $H_{0.1} (\text{StaggerPoint, EyePoint})$, that StaggerPoint is not more efficient than EyePoint.

All of the differences in the effectiveness metrics between StaggerPoint and EyePoint are statistically significant in favour of EyePoint (Table 7.2, Table 7.6, Table 7.7). This evidence is insufficient to reject the null hypothesis, $H_{0.2} (\text{StaggerPoint, EyePoint})$, that StaggerPoint is not more effective than EyePoint.

The fact that the only statistically significant differences in the effectiveness metrics between StaggerPoint and EyePoint are in favour of EyePoint is sufficient evidence that EyePoint is more effective than StaggerPoint.

The differences between StaggerPoint and EyePoint are statistically significant for three of the five satisfaction metrics in favour of EyePoint. These metrics include overall preference (Table 7.11), satisfaction (Table 7.10) and ease of use (Table 7.9). The differences between the two techniques for perceived mental effort (Table 7.12) and perceived physical effort (Table 7.13) are not statistically significant. This evidence is insufficient to reject the null hypothesis, $H_{0.3} (\text{StaggerPoint, EyePoint})$, that StaggerPoint is not more satisfying than EyePoint.

The fact that the only statistically significant differences in the satisfaction metrics between StaggerPoint and EyePoint are in favour of EyePoint is sufficient evidence that EyePoint is more satisfying than StaggerPoint.

Although StaggerPoint was found to be more efficient than EyePoint (rejection of $H_{0.1}$ (StaggerPoint, EyePoint)), EyePoint was found to be more satisfying and effective. These results are mixed. There is therefore insufficient evidence to reject the null
**hypothesis**, $H_0$ (*StaggerPoint, EyePoint*), that StaggerPoint is not more *usable* than EyePoint.

### 7.6.3 ScanPoint versus EyePoint

The difference in perceived *efficiency* between ScanPoint and EyePoint is statistically significant in favour of EyePoint (Table 7.8). The difference in actual efficiency between the two techniques is not statistically significant (Table 7.4). **This evidence is insufficient to reject the null hypothesis, $H_{0.1}$ (*ScanPoint, EyePoint*), that ScanPoint is not more *efficient* than EyePoint.**

The fact that the only statistically significant difference in the efficiency metrics between ScanPoint and EyePoint is in favour of EyePoint is sufficient evidence that EyePoint is more efficient than ScanPoint.

Two of the differences in the *effectiveness* metrics between ScanPoint and EyePoint are statistically significant in favour of EyePoint. These values include perceived accuracy (Table 7.6) and perceived accuracy decline for smaller targets (Table 7.7). The difference between the error percentages of the two techniques is not statistically significant (Table 7.2). **This evidence is insufficient to reject the null hypothesis, $H_{0.2}$ (*ScanPoint, EyePoint*), that ScanPoint is not more *effective* than EyePoint.**

The fact that the only statistically significant differences in the effectiveness metrics between ScanPoint and EyePoint are in favour of EyePoint is sufficient evidence that EyePoint is more effective than ScanPoint.

The differences between ScanPoint and EyePoint are statistically significant for three of the five *satisfaction* metrics in favour of EyePoint. These metrics include overall preference (Table 7.11), satisfaction (Table 7.10) and ease of use (Table 7.9). The differences between the two techniques for perceived mental effort (Table 7.12) and perceived physical effort (Table 7.13) are not statistically significant. **This evidence is insufficient to reject the null hypothesis, $H_{0.3}$ (*ScanPoint, EyePoint*), that ScanPoint is not more *satisfying* than EyePoint.**
The fact that the only statistically significant differences in the satisfaction metrics between ScanPoint and EyePoint are in favour of EyePoint is sufficient evidence that EyePoint is more satisfying than ScanPoint.

Given the fact that there are no statistically significant differences which favour ScanPoint over EyePoint and that none of \( H_{0.1} (\text{ScanPoint, EyePoint}), H_{0.2} (\text{ScanPoint, EyePoint}), H_{0.3} (\text{ScanPoint, EyePoint}) \) were rejected, there is insufficient evidence to reject the null hypothesis, \( H_0 (\text{ScanPoint, EyePoint}) \), that ScanPoint is not more usable than EyePoint.

The results for the comparisons involving EyePoint are summarised in Table 7.14.

<table>
<thead>
<tr>
<th>( H_{0}(i, \text{EyePoint}) )</th>
<th>i</th>
<th>TargetPoint</th>
<th>StaggerPoint</th>
<th>ScanPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{efficiency} ( H_{0.1} )</td>
<td>( i \in { \text{TargetPoint, StaggerPoint, ScanPoint} } )</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>\textit{effectiveness} ( H_{0.2} )</td>
<td></td>
<td>No*</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>\textit{satisfaction} ( H_{0.3} )</td>
<td></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>\textit{usability} ( H_0 )</td>
<td></td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

\textbf{Yes} : Sufficient evidence to reject null hypothesis  
\textbf{No} : Insufficient evidence to reject null hypothesis  
\* Sufficient evidence for rejection without Bonferroni adjustment

\begin{table}[h]
\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
\( H_{0}(i, \text{EyePoint}) \) & \textbf{i} & TargetPoint & StaggerPoint & ScanPoint \\
\hline
\textit{efficiency} \( H_{0.1} \) & \( i \in \{ \text{TargetPoint, StaggerPoint, ScanPoint} \} \) & No & Yes & No \\
\hline
\textit{effectiveness} \( H_{0.2} \) & & No* & No & No \\
\hline
\textit{satisfaction} \( H_{0.3} \) & & Yes & No & No \\
\hline
\textit{usability} \( H_0 \) & & Yes & No & No \\
\hline
\end{tabular}
\end{center}
\caption{Results for Hypotheses \( H_{0}(i, \text{EyePoint}) \)  
\( i \in \{ \text{TargetPoint, StaggerPoint, ScanPoint} \} \)}
\end{table}

The next set of hypotheses to be considered is \( H_0 \ (i, \text{Mouse}) \). For each \( i \in \{ \text{TargetPoint, StaggerPoint, ScanPoint} \} \) the lower level hypotheses need to be considered first.

### 7.6.4 TargetPoint versus Mouse

The difference in actual \textit{efficiency} between TargetPoint and the mouse is statistically significant in favour of the mouse (Table 7.4). The difference in perceived \textit{efficiency}
between the two techniques is not statistically significant (Table 7.8). **This evidence is insufficient to reject the null hypothesis, \( H_{0.1} (\text{TargetPoint, Mouse}) \), that TargetPoint is not more efficient than the mouse.**

The fact that the only statistically significant difference in the efficiency metrics between TargetPoint and the mouse is in favour of the mouse is sufficient evidence that the mouse is more efficient than TargetPoint.

Two of the differences in the effectiveness metrics between TargetPoint and the mouse are not statistically significant. These values include perceived accuracy (Table 7.6) and perceived accuracy decline for smaller targets (Table 7.7). The difference between the error percentages of the two techniques is statistically significant in favour of the mouse (Table 7.2). This evidence in **insufficient to reject the null hypotheses, \( H_{0.2} (\text{TargetPoint, Mouse}) \), that TargetPoint is not more effective than the mouse.**

The fact that the only statistically significant difference in the effectiveness metrics between TargetPoint and the mouse is in favour of the mouse is sufficient evidence that the mouse is more effective than TargetPoint.

None of the differences between TargetPoint and the mouse are statistically significant for the five satisfaction metrics (Table 7.9, Table 7.10, Table 7.11, Table 7.12, Table 7.13). **This evidence is insufficient to reject the null hypothesis, \( H_{0.3} (\text{TargetPoint, Mouse}) \), that TargetPoint is not more satisfying than the mouse.**

Given the fact that there are no statistically significant differences which favour TargetPoint over the mouse and that none of \( \{ H_{0.1} (\text{TargetPoint, Mouse}), H_{0.2} (\text{TargetPoint, Mouse}), H_{0.3} (\text{TargetPoint, Mouse}) \} \) were rejected, **there is insufficient evidence to reject the null hypothesis, \( H_0 (\text{TargetPoint, Mouse}) \), that TargetPoint is not more usable than the mouse.**

### 7.6.5 StaggerPoint versus Mouse

The difference in actual efficiency between StaggerPoint and the mouse is statistically significant in favour of the mouse (Table 7.4). The difference in perceived efficiency between the two techniques is also statistically significant in favour of the mouse
Chapter 7: Experimental Results

This evidence is insufficient to reject the null hypothesis, \( H_{0.1} \) (StaggerPoint, Mouse), that StaggerPoint is not more efficient than the mouse.

The fact that the only statistically significant differences in the efficiency metrics between StaggerPoint and the mouse are in favour of the mouse is sufficient evidence that the mouse is more efficient than TargetPoint.

The differences in all three of the effectiveness metrics between StaggerPoint and the mouse are statistically significant in favour of the mouse. These values include perceived accuracy (Table 7.6), perceived accuracy decline for smaller targets (Table 7.7) and error percentage (Table 7.2). This evidence is insufficient to reject the null hypotheses, \( H_{0.2} \) (StaggerPoint, Mouse), that StaggerPoint is not more effective than the mouse.

The fact that the only statistically significant differences in the effectiveness metrics between StaggerPoint and the mouse are in favour of the mouse is sufficient evidence that the mouse is more effective than StaggerPoint.

The differences between StaggerPoint and the mouse are statistically significant for four of the five satisfaction metrics in favour of the mouse. These metrics include overall preference (Table 7.11), satisfaction (Table 7.10), ease of use (Table 7.9) and perceived mental effort (Table 7.12). The difference between the two techniques for perceived physical effort (Table 7.13) is not statistically significant. This evidence is insufficient to reject the null hypothesis, \( H_{0.3} \) (StaggerPoint, Mouse), that StaggerPoint is not more satisfying than the mouse.

The fact that the only statistically significant differences in the satisfaction metrics between StaggerPoint and the mouse are in favour of the mouse is sufficient evidence that the mouse is more satisfying than StaggerPoint.

Given the fact that there are no statistically significant differences which favour StaggerPoint over the mouse and that none of \( \{ H_{0.1} \) (StaggerPoint, Mouse), \( H_{0.2} \) (StaggerPoint, Mouse), \( H_{0.3} \) (StaggerPoint, Mouse) \} were rejected, there is insufficient evidence to reject the null hypothesis, \( H_0 \) (StaggerPoint, Mouse), that StaggerPoint is not more usable than the mouse.
Chapter 7: Experimental Results

7.6.6 ScanPoint versus Mouse

The difference in actual efficiency between ScanPoint and the mouse is statistically significant in favour of the mouse (Table 7.4). The difference in perceived efficiency between the two techniques is also statistically significant (Table 7.8). This evidence is insufficient to reject the null hypothesis, \( H_{0.1} \) (ScanPoint, Mouse), that ScanPoint is not more efficient than the mouse.

The fact that the only statistically significant differences in the efficiency metrics between ScanPoint and the mouse are in favour of the mouse is sufficient evidence that the mouse is more efficient than ScanPoint.

Three of the differences in the effectiveness metrics between ScanPoint and the mouse are statistically significant in favour of the mouse. These values include perceived accuracy (Table 7.6), perceived accuracy decline for smaller targets (Table 7.7) and error percentage (Table 7.2). This evidence is insufficient to reject the null hypotheses, \( H_{0.2} \) (ScanPoint, Mouse), that ScanPoint is not more effective than the mouse.

The fact that the only statistically significant differences in the effectiveness metrics between ScanPoint and the mouse are in favour of the mouse is sufficient evidence that the mouse is more effective than ScanPoint.

The differences between ScanPoint and the mouse are statistically significant for four of the five satisfaction metrics in favour of the mouse. These metrics include overall preference (Table 7.11), satisfaction (Table 7.10), ease of use (Table 7.9) and perceived mental effort (Table 7.12). The difference between the two techniques for perceived physical effort (Table 7.13) is not statistically significant. This evidence is insufficient to reject the null hypothesis, \( H_{0.3} \) (ScanPoint, Mouse), that ScanPoint is not more satisfying than the mouse.

The fact that the only statistically significant differences in the satisfaction metrics between ScanPoint and the mouse are in favour of the mouse is sufficient evidence that the mouse is more satisfying than ScanPoint.
Chapter 7: Experimental Results

Given the fact that there are no statistically significant differences which favour ScanPoint over the mouse and that none of \( H_{0.1} \) (ScanPoint, Mouse), \( H_{0.2} \) (ScanPoint, Mouse), \( H_{0.3} \) (ScanPoint, Mouse) were rejected, there is insufficient evidence to reject the null hypothesis, \( H_0 \) (ScanPoint, Mouse), that ScanPoint is not more usable than the mouse.

The results for the comparisons involving the mouse are summarised in Table 7.15.

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>TargetPoint</th>
<th>StaggerPoint</th>
<th>ScanPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H_{0.1} ) (i, Mouse) efficiency</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>( H_{0.2} ) (i, Mouse) effectiveness</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>( H_{0.3} ) (i, Mouse) satisfaction</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>( H_0 ) (i, Mouse) usability</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Yes</th>
<th>Sufficient evidence to reject null hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Insufficient evidence to reject null hypothesis</td>
</tr>
</tbody>
</table>

Table 7.15: Results for Hypotheses \( H_{0.x} \) (i, Mouse)

i ∈ \{TargetPoint, StaggerPoint, ScanPoint\}

7.7 Design Principles Revisited

Seven design guidelines were proposed during the design phase (Section 4.2). A subset of these guidelines was based purely on existing theory (Section 4.2.1). Another subset was proposed for guiding the implementation of a visual feedback highlight (Section 4.2.2). TargetPoint was the only technique to implement visual feedback in the manner prescribed by guidelines 5, 6 and 7 (Section 4.2.2). The successful application of these design principles was demonstrated by TargetPoint. Feedback from test participants regarding the feedback highlight of TargetPoint was overwhelmingly positive. TargetPoint was also more popular among test participants and found to be more usable than an existing technique, EyePoint.
StaggerPoint and ScanPoint did not conform to all of the design principles (in order to explore alternative approaches to target selection disambiguation). Negative comments from test participants regarding certain aspects of these techniques suggest that additional design principles may be required. Based on negative comments regarding the staggering of the targets for StaggerPoint the following (8th) guideline is proposed:

8. Gaze selection techniques should limit the impact of disambiguation mechanisms on the aesthetics of a user interface

An additional principle is also proposed based on negative feedback from test participants regarding ScanPoint:

9. A visual feedback highlight should be under the control of the user

An exception to this principle may be for the traditional use of scanning, namely for users with severe motor disabilities (Section 3.4.1.1). Design principle 9 also addresses the issue of users having to wait for the timed highlight to loop around through all the options.

The nine resulting design principles for gaze selection techniques are:

1. Support target selection disambiguation to improve selection accuracy
2. Favour gaze added interfaces for able bodied users
3. Favour a hardware button trigger for able bodied users
4. Magnification and target expansion should only be applied when performing a selection
5. Visual feedback is useful, but should only be applied when performing a selection
6. Visual feedback should be in the form of a target highlight and not a gaze cursor
7. Visual feedback should only be applied to sufficiently spaced/enlarged targets
8. Gaze selection techniques should limit the impact of disambiguation mechanisms on the aesthetics of a user interface
9. **A visual feedback highlight should be under the control of the user**

**7.8 Conclusions**

Of the six high level research hypotheses $H_0 (i, j)$ (Section 6.2) there was only sufficient evidence to reject one null hypothesis $H_0$ (TargetPoint, EyePoint) (Section 7.6.1). This indicates that TargetPoint provides an improvement in usability over EyePoint. TargetPoint was found to provide a higher level of satisfaction than EyePoint (rejection of $H_{0.3}$ (TargetPoint, EyePoint)) (Table 7.14). The lower error rate of TargetPoint compared to EyePoint is only statistically significant if not considered within the context of the other techniques which were evaluated (Section 7.6.1). TargetPoint was the only gaze selection technique which was found to provide a higher level of usability than EyePoint (Table 7.14). StaggerPoint was found to be more efficient than EyePoint (Section 7.6.2), but was less satisfying and less effective. None of the novel gaze selection techniques was found to provide any statistically significant usability advantages over the mouse (Sections 7.6.4, 7.6.5, 7.6.6 and Table 7.15).

One of the most significant findings was the strong positive response of participants towards the visual feedback highlight provided by TargetPoint (Sections 7.4.4 and 7.5.1). The lack of visual feedback in EyePoint drew significant criticism from participants, with 43% ($n=13$) of participants complaining about the lack of feedback in this technique (Section 7.5.2). This evidence supports design principles 5, 6 and 7 (Section 4.2.2). Unlike the other design principles, these three could not be supported by existing research alone. It was therefore important to test participants' responses to a visual feedback highlight that conformed to these principles. Two additional design principles were proposed (Section 7.7) based on feedback from test participants.

Visual feedback was also demonstrated to be a more effective method of reducing early trigger errors than trigger point time shifting (Section 7.3.1). Unlike trigger point time shifting, a visual feedback highlight does not result in an increase in late trigger errors.
The popularity of the feedback highlight among test participants is also noteworthy, as other researchers have previously argued (Section 3.7) that visual feedback for this type of interaction would not be viewed favourably by users. This result has implications for designers of future gaze selection techniques. The data suggests that, unlike previously argued, a lack of visual feedback is not necessarily desirable for this type of interaction technique.

The popularity of TargetPoint over EyePoint is significant. TargetPoint was ranked ahead of EyePoint by the majority of test participants in this experimental evaluation. On average, test participants rated TargetPoint above or equal to EyePoint for every self reported metric. StaggerPoint and ScanPoint were the least popular overall. The designs of these techniques explored novel approaches to target selection disambiguation. Both disambiguation techniques (staggering and scanning highlight) proved effective in reducing the number of selection errors. The short scanning delay used for ScanPoint proved problematic for participants. A longer delay would have likely resulted in fewer selection errors and less frustration. This consideration needs to be weighed against the other problem of participants not wanting to wait a long time for the highlight to reach the correct button.

Results suggest that user preference for gaze over the mouse for selection tasks may not be as strong as previously reported. In fact more than half of the test participants (57%, \( n = 17 \)) ranked the mouse ahead of the most popular of the gaze selection techniques evaluated (TargetPoint). Preference for the mouse over EyePoint is particularly strong, with 70% of participants \( (n = 21) \) favouring the mouse. The mouse was also significantly faster and more accurate than all of the gaze selection techniques evaluated.

The absence of visual magnification (to accompany motor space expansion) in TargetPoint was found to draw the users’ gaze closer to the centre of the targets (Figure 7.4), confirming earlier work (Section 3.5.1.1). Overall, TargetPoint was more accurate, and had fewer early trigger errors, late trigger errors and accuracy errors than EyePoint, although the effect of drawing users' gaze towards the centre of targets did not appear to have a statistically significant effect on accuracy errors.
Chapter 7: Experimental Results

The practical utility of the framework for implementing gaze selection techniques was demonstrated by the successful implementation of eight gaze selection techniques. This number consists of seven novel gaze selection techniques and EyePoint (Chapters 4 and 5). The framework has also been used to successfully conduct a usability evaluation. The data gathering capabilities of the framework have also been demonstrated by the data analysis presented. The data gathered using the framework has proven useful for analysis purposes (for simulations in particular), demonstrating the practical usefulness of the framework for recording relevant test data for gaze selection techniques. Implementing all of the techniques which were evaluated using the framework also ensured that tests were automated (and data collected) in a consistent manner, facilitating a valid comparison of the techniques.
Chapter 8: Conclusion

8.1 Introduction
The main goal of this research was to improve the usability of gaze selection techniques. Novel techniques which were developed were designed to address issues such as tracking inaccuracy and early trigger errors. Design principles were also proposed for guiding the development of gaze selection techniques. A framework for implementing and evaluating gaze selection techniques was developed in order to implement the novel techniques. A comparative evaluation against an existing technique, EyePoint, and the mouse was conducted using the framework in order to measure the usability of the novel techniques relative to existing approaches.

In this chapter the findings of this research are summarised (Section 8.2). Research objectives identified in Chapter 1 are revisited in order to ascertain whether they were accomplished or not. Hypotheses and design principles are also revisited. The theoretical and practical contributions of this research are discussed (Section 8.3), followed by a discussion of the limitations of this research (Section 8.4). A number of suggestions for future research are also provided (Section 8.5).

8.2 Summary of Findings
The discussion of the findings is divided into three sections. The hypotheses are discussed first (Section 8.2.1). In the second section (Section 8.2.2) the overall goal and objectives of this research are revisited and relevant findings presented. A brief discussion of the design principles is provided in Section 8.2.3.
8.2.1 Hypotheses

The hypotheses all relate to usability (Section 6.2). Each of the six high level hypotheses was represented as a null hypothesis to be tested, and if warranted, rejected in favour of the corresponding alternative hypothesis.

\[ H_0 (i, j) : \text{Technique } i \text{ is not more usable than technique } j, \ i \in \{ \text{TargetPoint, StaggerPoint, ScanPoint}\}, \ j \in \{ \text{EyePoint, mouse}\} \]

There was only sufficient evidence to reject one of the high level hypotheses: \( H_0 (\text{TargetPoint, EyePoint}) : \text{TargetPoint is not more usable than EyePoint} \) (Section 7.6.1). This result indicates that TargetPoint is more usable than EyePoint. At a lower level, TargetPoint was found to be more satisfying than EyePoint as \( H_{0.3} (\text{TargetPoint, EyePoint}) \) was rejected. StaggerPoint was found to be more efficient than EyePoint as \( H_{0.1} (\text{StaggerPoint, EyePoint}) \) was rejected. StaggerPoint was not found to be more usable than EyePoint due to inferior effectiveness and satisfaction (Section 7.6.2). These results are summarised in Table 7.14. ScanPoint was not found to provide any usability advantages over EyePoint (Section 7.6.3). The error percentage for TargetPoint was 3.3% compared to 6% for EyePoint (Section 7.3.1). Without Bonferroni adjustment, the lower error percentage of TargetPoint when compared to EyePoint is also statistically significant. Invisible motor space expansion was found to draw participants' gaze significantly closer to the centre of targets (Section 7.3.1), confirming existing research. The influence of this effect on accuracy errors was unclear. The number of errors which could be attributed to inaccuracy was lower for TargetPoint than EyePoint, but the difference was not found to be statistically significant.

None of the three techniques being considered (TargetPoint, StaggerPoint, ScanPoint) was found to provide any usability advantages over the mouse (Section 7.6.4, 7.6.5 and 7.6.6). The results for the hypotheses involving the mouse are summarised in Table 7.15.

The number of selection errors using the mouse was significantly lower compared to any of the gaze selection techniques evaluated, including the most accurate gaze
selection technique evaluated (TargetPoint) (Section 7.3.1). The accuracy advantage of the mouse is consistent with the findings of existing research (Sections 3.5.2.3 and 3.5.2.4). The mouse was also found to be the most efficient technique (Section 7.3.2) and preferred by the majority of the test participants (Section 7.4.3). Seventy percent of participants preferred the mouse to EyePoint, which differs from an existing study where three quarters of participants preferred EyePoint (Section 3.5.2.3).

8.2.2 Goal and Objectives

The overall goal was to improve the usability of gaze selection techniques (Section 1.3). This goal has been satisfied as TargetPoint, a novel technique developed during this research, was found to be more usable than EyePoint (Section 7.6.1).

The primary objectives of this research were to:

1. Develop novel gaze selection techniques which minimise selection errors and maximise usability
2. Explore novel approaches to target selection disambiguation
3. Compare the usability of the proposed techniques to that of existing techniques
4. Investigate the effect of visual feedback on selection errors and user satisfaction

In addressing the first objective, seven novel gaze selection techniques were developed (Chapter 4). Three of these techniques were evaluated in a usability evaluation. TargetPoint, in particular, was found to minimise selection errors (Section 7.3.1) and was preferred by 70% of test participants over the existing benchmark, EyePoint (Section 7.4.3). StaggerPoint and ScanPoint reduced the number of selection errors when compared to selection without any disambiguation, although these two techniques were less popular with test participants (Section 7.4.3). Only two test participants ranked StaggerPoint as their favourite technique.

StaggerPoint and ScanPoint represent novel approaches to target selection disambiguation (objective 2). StaggerPoint (Section 4.3.3) used a staggered arrangement of targets to significantly reduce the percentage of incorrect selections
when compared to selection without staggering (from 43% to 20%) (Section 7.3.1). ScanPoint represents a novel variation to existing scanning techniques (Section 4.3.2). This technique was also found to significantly reduce the percentage of incorrect selections when compared to gaze selection without any disambiguation mechanism (from 46% to 9%). The usability of three novel gaze selection techniques was compared to that of an existing technique (EyePoint) and the mouse (objective 3). EyePoint was selected as a basis for comparison as it was argued that this technique represents the state of the art in gaze selection techniques. TargetPoint was found to be more usable than EyePoint (Section 7.6.1). StaggerPoint was found to be 48% more efficient than EyePoint (Section 7.6.2) but less satisfying and effective. ScanPoint was not found to provide any usability advantages over EyePoint (Section 7.6.3). None of the novel gaze selection techniques evaluated were found to provide any usability advantages over the mouse (Table 7.15).

TargetPoint implemented a visual feedback highlight in support of objective 4. The highlight was found to significantly reduce the number of early trigger errors (Section 7.3.1). A visual feedback highlight was also found to be a more effective method of reducing early trigger errors than trigger point time shifting. The latter approach was found to lead to an increase in late trigger errors (Section 7.3.1). The form of visual feedback provided by TargetPoint also proved popular with test participants (Sections 7.4.3, 7.4.4 and 7.5.1). The positive findings regarding visual feedback contradict existing views regarding this type of feedback (Section 3.7).

The secondary objectives of this research were to:

1. Propose a set of design principles for designing gaze selection techniques
2. Develop a framework for the purpose of implementing and evaluating\(^1\) the proposed techniques

---

\(^1\) Required evaluation functionality limited to automated presentation of test targets to test participants and data collection
The design principles proposed (secondary objective 1) to guide the development of gaze selection techniques are derived from documented research, prototyping of techniques and usability evaluation (Section 7.7).

The framework discussed in Chapter 5 (secondary objective 2) was found to reduce the effort required to implement new gaze selection techniques. It also proved effective in automating testing and recording the data required for evaluation purposes (Chapters 6 and 7). The data gathered proved particularly useful for simulation and analysis purposes. As the framework has been used to implement eight gaze selection techniques (including EyePoint), its practical usefulness has been demonstrated.

### 8.2.3 Design Principles

Three of the seven design principles identified in Chapter 4 were based on an argument that visual feedback was beneficial, rather than motivated purely by existing theory and results (Section 4.2.2). These principles therefore required confirmation from experimental data. The positive responses from participants regarding the feedback highlight of TargetPoint in particular supported these proposed design principles. TargetPoint was the only technique to implement visual feedback in accordance with these principles. Two additional design principles were proposed based on negative feedback from participants regarding certain aspects of StaggerPoint and ScanPoint. Important issues were identified in Chapter 7 regarding StaggerPoint and the timed highlight of ScanPoint. These design principles were proposed (Section 7.7), in order to address the shortcomings identified.

### 8.3 Summary of Contributions

The contributions of this research have been divided into two categories, namely practical contributions and theoretical contributions. The practical contributions primarily relate to the development of novel gaze selection techniques and the framework, while the theoretical contributions primarily relate to different aspects of the techniques and how the evaluation results impact on the existing body of knowledge on gaze selection techniques.
8.3.1 Practical Contributions
The gaze selection techniques developed during this research represent a practical contribution. Three novel gaze selection techniques (TargetPoint, StaggerPoint and ScanPoint) were designed and implemented (Section 4.3). TargetPoint was found to provide superior usability to EyePoint, the current state of the art in gaze selection techniques (Section 7.6.1). StaggerPoint was found to be more efficient than EyePoint, but like ScanPoint, was not preferred by most test participants (Section 7.4.3). Four additional gaze selection techniques were also proposed and implemented, but not evaluated due to scope constraints (Section 4.3.4).

Two novel gaze selection disambiguation mechanisms were also developed, namely the staggered gaze interpretation of StaggerPoint (Section 4.3.3) and scanning highlight of ScanPoint (Section 4.3.2). ScanPoint uses the existing concept of scanning, but bases the starting point of the scanning process on the user's gaze. Staggered gaze interpretation and the timed highlight were found to significantly reduce the number of selection errors compared to gaze selection without any disambiguation mechanism (Section 7.3.1).

The results of the evaluation (Chapter 7) represent a practical contribution as other researchers may wish to compare the evaluation results obtained to their own. The framework which was designed and implemented provides support for implementing and evaluating gaze selection techniques (Chapter 5). This framework was used to implement all eight gaze selection techniques (including EyePoint), and evaluate four of them (and the mouse) (Chapter 6 and 7). The practical usefulness of this framework was demonstrated by using it to successfully conduct this research.

8.3.2 Theoretical Contributions
The results obtained confirm the theory of Miniotas et al. (2004) that motor space expansion, when not accompanied by visual expansion draws the user's gaze closer to the centre of targets (Section 7.3.1).
The statistically significant reduction of early trigger errors by trigger point time shifting (Section 7.3.1) in the EyePoint implementation (“the magnified technique”) provides support for existing early trigger correction research (Kumar et al. 2008).

Trigger point time shifting was found to result in a statistically significant increase in late trigger errors, indicating a possible drawback to trigger point time shifting (Section 7.3.1). Existing research (Kumar et al. 2008) only considers the effect of early trigger correction in speed tasks, but does not consider a mixture of speed and accuracy tasks which is more likely to occur in practice.

Results obtained in the evaluation of TargetPoint indicate that the use of a visual feedback highlight may result in a reduction in early trigger errors (Section 7.3.1). When considered along with the issue of late trigger errors associated with trigger point time shifting, it was concluded that a visual feedback highlight is a more effective method of combating early trigger errors.

The positive findings regarding visual feedback contradicts existing research (Section 3.7). Positive comments from test participants regarding the visual feedback highlight of TargetPoint (Section 7.5.1) and negative comments regarding the lack of this feature in EyePoint (Section 7.5.2) are particularly noteworthy. These comments contradict one of the design principles of EyePoint, namely to “avoid slaving any of the interaction directly to eye movements (i.e. not overload the visual channel for pointing)” (Kumar et al. 2007). The majority of test participants also rated the TargetPoint highlight as useful (Section 7.4.4). This result has implications for designers of future gaze selection techniques. The data suggests that, unlike previously argued, a lack of visual feedback is not necessarily desirable for this type of interaction technique.

The design principles related to visual feedback (Section 4.2.2) encapsulate the approach to visual feedback of TargetPoint. As the vast majority of test participants' opinions regarding visual feedback in TargetPoint were positive (Section 7.5.1, Section 7.4.4), it stands to reason that these design principles may be useful to developers of future gaze selection techniques. Additional design principles based on the experimental results are also proposed (Section 7.7).
Chapter 8: Conclusion

8.4 Limitations of Investigation

Test participants were drawn from students and faculty of the NMMU Department of Computer Science and Information Systems (convenience sample). This group has more computer knowledge than the average user, so it is difficult to generalise the results obtained. Further research may be necessary in order to determine whether this positive view of visual feedback extends to a wider and more general range of test users.

The evaluation consisted of a short term study. A long term study would be needed to ensure that the novelty value of gaze selection did not affect users' perceptions. Participants had years of experience with the mouse, but were inexperienced with gaze selection.

Test targets consisted of a stack of rectangular buttons. One reason for this choice was to facilitate the evaluation of StaggerPoint alongside the other gaze selection techniques. Although the use of artificial test targets is the norm rather than the exception for the evaluation of gaze selection techniques, a more realistic scenario would involve more complex applications.

The implementation of EyePoint which was evaluated, was developed using the framework based on published descriptions of EyePoint. It may be the case that there are minor differences between this implementation and the original EyePoint implementation. If such differences exist, the two implementations are sufficiently similar that the high level comparison between the two techniques remains valid. EyePoint represents a magnified pop-up approach whereas TargetPoint represents a motor space expansion with visual feedback approach.

8.5 Future Research

The four techniques described in Chapter 4 which have not yet been formally evaluated also provide opportunities for future research. Gaze proximity typing and the gaze-joystick technique are particularly promising as alternative techniques. The gaze-joystick technique is currently implemented using a full-sized joystick (Section 4.3.4.3). As noted in Chapter 4, a low-profile thumb joystick which supports a press
and release action would suit this technique. An evaluation of this technique could be conducted with a modified keyboard with a low-profile joystick. This joystick would preferably be located below the space bar for easy access without users having to move their hands between the keyboard and another input device.

One possibility is to evaluate a version of EyePoint modified to provide visual feedback in a manner similar to TargetPoint (in conformance with the proposed design principles). Alternatively, TargetPoint could be evaluated with varying degrees of visual target expansion to accompany the motor space expansion of targets. These options could be evaluated in order to determine whether users prefer having no visual expansion, full visual expansion or a limited degree of visual expansion. The effects of different degrees of visual expansion on selection errors could also be evaluated, although a large test population would likely be required in order to prove statistical significance. Another option to consider is evaluating the effect of a fixation threshold on the visual feedback highlight of TargetPoint.

An alternative method of drawing users' gaze close to the centre of targets would be to provide the feedback highlight in the form of a small marker in the centre of the target. In this manner it would not matter whether the target was expanded visually or only in motor space.

One of the design principles put forward in Chapter 4 (based on existing literature and published experiments) was that a gaze added interface should be favoured over a gaze only interface for able-bodied users. In order to compare the gaze interaction aspects of the techniques evaluated in Chapter 7, users were not given an option of freely switching between the gaze and mouse modalities. One possibility for future research would be to provide users with the option of switching between TargetPoint, EyePoint and the mouse for different tasks in order to get a better idea of the techniques users prefer. A long term study where users would use the techniques for everyday use would provide the most realistic results. This type of study is currently impractical due to the high cost of eye tracking equipment and the fact that current eye gaze interaction techniques have yet to prove practical for a wide variety of everyday computing tasks (beyond systems designed purely for disabled users).
One opportunity for future research is to generalise TargetPoint so that it can fulfill the role of a gaze selection technique and an eye pointing technique. For cases involving discrete targets, the design of TargetPoint could be used. In cases where there are no discrete targets (for example drawing a dot on a paint canvas) the magnified approach of EyePoint with focus points could be adopted. Thus within a single pop-up there could be a mixture of discrete targets and general pointing areas. An alternative scheme would be to involve the use of different trigger keys – one key for gaze selections with TargetPoint and another for eye pointing using EyePoint. Users could interact with applications providing the semantic information required by TargetPoint using TargetPoint. EyePoint could be used as a fallback for applications which do not provide this information, or cases where a more general pointing solution is required.

Results from this research, particularly qualitative feedback from test participants, confirm the importance of accuracy and ease of use in gaze selection techniques. Despite the recent development of improved gaze selection techniques such as EyePoint and TargetPoint, the mouse is still significantly more accurate than gaze selection. Accuracy is therefore the primary weakness of gaze as an input modality. The challenge therefore remains to develop a gaze selection technique able to match the effectiveness of the mouse without compromising on other aspects of usability.
References


Barnum C.M., Usability Testing and Research, Longman, 2002

Bates, R., Multimodal Eye-Based Interaction for Zoomed Target Selection on a Standard Graphical User Interface, Human-Computer Interaction - INTERACT'99 (Volume II), S. Brewster, A. Cawsey and G. Cockton (Editors), pages 7 – 8, 1999


References


Ely, M., Anzul, M., Friedman, T., Garner, D., MacCormack Steinmetz, A., Doing Qualitative Research: Circles Within Circles, Falmer Press, 1995


International Standards Organisation (ISO), ISO 9241 – 1, Ergonomic requirements for office work with visual display terminals, 1997

Isokoski, P., Text Input Methods for Eye Trackers Using Off-Screen Targets, Proceedings of the 2000 ACM SIGCHI Eye Tracking Research and Applications Symposium (ETRA '00), Palm Beach Gardens, United States, pages 15 – 21, 2000


Jacob, R.J.K., Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces in Advances in Human-Computer Interaction, Vol. 4, Edited by H.R. Hartson and D. Hix, Ablex Publishing Co., pages 151-190, 1993


References


Kohlbecher, S., Bardinsy, S., Bartlz, K., Schneider, E., Poitschke, T., Ablassmeierk, M., Calibration-free eye tracking by reconstruction of the pupil ellipse in 3D space, Proceedings of the 2008 ACM SIGCHI Eye Tracking Research and Applications Symposium (ETRA '08), Savannah, Georgia, United States, pages 135 - 138, 2008

Kumar, M., GUIDe Saccade Detection and Smoothing Algorithm. Technical Report CSTR 2007-03, Stanford University, Stanford


Kumar, M., Klingner, J., Puranik, R., Winograd, T., Paepcke, Improving the Accuracy of Gaze Input for Interaction, A., Proceedings of the 2008 ACM SIGCHI Eye Tracking
References

Research and Applications Symposium (ETRA '08), Savannah, Georgia, United States, pages 65 - 68, 2008


Li, D., Babcock, J., Parkhurst, D.J., openEyes: A Low-cost Head-mounted Eye-tracking Solution, Proceedings of the 2006 ACM SIGCHI Eye Tracking Research and Applications Symposium (ETRA '06), San Diego, United States, pages 95-100, 2006


References


Miniotas, D and Špakov, O., An Algorithm to Counteract Eye Jitter in Gaze-Controlled Interfaces, Information Technology and Control 1 (30), pages 65–68, 2004


184


Patmore, D.W., Knapp, R.B., Towards an EOG-Based Eye Tracker for Computer Control, Devices, Proceedings of the First International ACM SIGCAPH Conference
on Assistive Technologies (ASSETS '98), Marina del Rey, United States, pages 197 – 203, 1998

Pelz, J.B., Canosa, R., Babcock, J, Extended Tasks Elicit Complex Eye Movement Patterns, Proceedings of the 2000 ACM SIGCHI Eye Tracking Research and Applications Symposium (ETRA '00), Palm Beach Gardens, United States, pages 37 - 44, 2000


Smith, B.A., Ho, J., Ark, W., Zhai, S., Hand Eye Coordination Patterns in Target Selection, Proceedings of the 2000 ACM SIGCHI Eye Tracking Research and Applications Symposium (ETRA ’00), Palm Beach Gardens, United States, pages 117 – 122, 2000

Špakov, O., Miniotas, D., Gaze-Based Selection of Standard-Size Menu Items, Proceedings of ICMI’05, Trento, Italy, pages 124 – 128, 2005


ten Kate, J.H., Frietman, E.E.E., Willems, W., ter Haar Romeny, B.M., Tenkink, E., Eye-Switch Controlled Communication Aids, Proceedings of the 12th International Conference on Medical and Biological Engineering, Jerusalem, Israel, 1979


188
Tullis, T., Albert, B., Measuring The User Experience, Morgan Kaufmann, 2008


References


Appendix A: Pre-test Questionnaire

Biographical Details:
Computer experience: _______ (years)

Experience using the mouse: _______ (years)

| How would you describe your level of computer literacy? |
|----------------|----------------|----------------|
| Novice         | Intermediate   | Advanced       |

Age: ______

<table>
<thead>
<tr>
<th>Eye Colour (mark with an 'X')</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Your gender (mark with an 'X')</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Do you wear glasses?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Do you wear contact lenses?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
</tr>
</tbody>
</table>

Do you have any other eye conditions e.g. colour blindness? (If yes, which condition(s))
Appendix B: Post-test Questionnaire

Evaluation of Interaction Techniques:

<table>
<thead>
<tr>
<th>Technique: Magnified Pop-up</th>
<th>Description:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The 'control' key is held down and a pop-up appears on the screen displaying a magnified view of the targets. A selection is made by releasing the 'control' key.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>This technique was accurate</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>This technique was less accurate when the buttons were smaller</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>With this technique I could select items quickly</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>This technique was easy to use</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overall I was satisfied with this technique</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Using this technique required a lot of mental effort</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Using this technique required a lot of physical effort</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1  2  3  4  5</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B: Post-test Questionnaire

### Technique: Timed Highlight

**Description:**
The 'control' key is held down and the item that the user is looking at is highlighted. Alternative selections are presented after a time interval. A selection is made by releasing the 'control' key.

<table>
<thead>
<tr>
<th>Question</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>This technique was accurate</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>This technique was less accurate when the buttons were smaller</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>With this technique I could select items quickly</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>This technique was easy to use</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Overall I was satisfied with this technique</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Using this technique required a lot of mental effort</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Using this technique required a lot of physical effort</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>
## Technique: Spaced Pop-up

**Description:**
The 'control' key is held down and a pop-up appears on the screen displaying the targets spaced more widely. A selection is made by releasing the 'control' key. Targets are highlighted in blue when looked at.

<table>
<thead>
<tr>
<th></th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>This technique was</td>
<td></td>
<td></td>
</tr>
<tr>
<td>accurate</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>This technique was</td>
<td></td>
<td></td>
</tr>
<tr>
<td>less accurate when</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>the buttons were</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>smaller</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>With this technique</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I could select items</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>quickly</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>This technique was</td>
<td></td>
<td></td>
</tr>
<tr>
<td>easy to use</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Overall I was</td>
<td></td>
<td></td>
</tr>
<tr>
<td>satisfied with this</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>technique</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Using this technique</td>
<td></td>
<td></td>
</tr>
<tr>
<td>required a lot of</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>mental effort</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Using this technique</td>
<td></td>
<td></td>
</tr>
<tr>
<td>required a lot of</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>physical effort</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>I found the target</td>
<td></td>
<td></td>
</tr>
<tr>
<td>highlight useful</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
### Technique: Staggered Selection

**Description:**

The target item is selected by looking at it and pressing the 'control' key.

<table>
<thead>
<tr>
<th>This technique was accurate</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>This technique was less accurate when the buttons were smaller</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>With this technique I could select items quickly</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>This technique was easy to use</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overall I was satisfied with this technique</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Using this technique required a lot of mental effort</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Using this technique required a lot of physical effort</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
### Technique: Mouse Selection

**Description:**

The target item is selected by clicking on it with the mouse.

<table>
<thead>
<tr>
<th>Question</th>
<th>Strongly Disagree</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>This technique was accurate</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>This technique was less accurate when the buttons were smaller</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>With this technique I could select items quickly</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>This technique was easy to use</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Overall I was satisfied with this technique</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Using this technique required a lot of mental effort</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Using this technique required a lot of physical effort</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
## Comments

<table>
<thead>
<tr>
<th>Mouse Selection:</th>
<th>Describe negative aspects of this technique:</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Mouse Selection Image" /></td>
<td><img src="image" alt="Description" /></td>
</tr>
<tr>
<td>Describe positive aspects of this technique:</td>
<td></td>
</tr>
<tr>
<td>General comments:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Timed Highlight:</th>
<th>Describe negative aspects of this technique:</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Timed Highlight Image" /></td>
<td><img src="image" alt="Description" /></td>
</tr>
<tr>
<td>Describe positive aspects of this technique:</td>
<td></td>
</tr>
<tr>
<td>General comments:</td>
<td></td>
</tr>
</tbody>
</table>
### Appendix B: Post-test Questionnaire

#### Spaced Pop-up:

<table>
<thead>
<tr>
<th>Describe negative aspects of this technique:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Describe positive aspects of this technique:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General comments:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

#### Magnified Pop-up:

<table>
<thead>
<tr>
<th>Describe negative aspects of this technique:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Describe positive aspects of this technique:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>General comments:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
### Appendix B: Post-test Questionnaire

#### Staggered Selection:

**Describe negative aspects of this technique:**

- 
- 
- 
- 

**Describe positive aspects of this technique:**

- 
- 
- 
- 

**General comments:**

- 
- 
- 
- 

---

- Button 1
- Button 2
- Button 3
- Button 4
- Button 5
- Button 6
- Button 7
**Preference**

If given a choice of which technique to use, how would you rank the different techniques? Fill in a ranking next to each technique in the table below. A ranking of 1 is to be given to your favourite technique, and a ranking of 5 to the technique you like the **least**. All the techniques must be ranked. **You may not give two techniques the same ranking.**

<table>
<thead>
<tr>
<th>Technique</th>
<th>Ranking (fill in a number)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mouse Selection:</strong></td>
<td></td>
</tr>
<tr>
<td>Button 3</td>
<td></td>
</tr>
<tr>
<td>Button 4</td>
<td></td>
</tr>
<tr>
<td>Button 5</td>
<td></td>
</tr>
<tr>
<td>Button 6</td>
<td></td>
</tr>
<tr>
<td>Button 7</td>
<td></td>
</tr>
<tr>
<td><strong>Timed Highlight:</strong></td>
<td></td>
</tr>
<tr>
<td>Button 4</td>
<td></td>
</tr>
<tr>
<td>Button 5</td>
<td></td>
</tr>
<tr>
<td>Button 6</td>
<td></td>
</tr>
<tr>
<td>Button 7</td>
<td></td>
</tr>
<tr>
<td><strong>Spaced Pop-up</strong></td>
<td></td>
</tr>
<tr>
<td>Button 6</td>
<td></td>
</tr>
<tr>
<td>Button 7</td>
<td></td>
</tr>
<tr>
<td>Button 8</td>
<td></td>
</tr>
<tr>
<td><strong>Magnified Pop-up:</strong></td>
<td></td>
</tr>
<tr>
<td>Button 5</td>
<td></td>
</tr>
<tr>
<td>Button 6</td>
<td></td>
</tr>
<tr>
<td>Button 7</td>
<td></td>
</tr>
<tr>
<td><strong>Staggered Selection:</strong></td>
<td></td>
</tr>
<tr>
<td>Button 1</td>
<td></td>
</tr>
<tr>
<td>Button 2</td>
<td></td>
</tr>
<tr>
<td>Button 3</td>
<td></td>
</tr>
<tr>
<td>Button 4</td>
<td></td>
</tr>
<tr>
<td>Button 5</td>
<td></td>
</tr>
<tr>
<td>Button 6</td>
<td></td>
</tr>
<tr>
<td>Button 7</td>
<td></td>
</tr>
</tbody>
</table>