Enhanced Sensor-Based Interaction Techniques for Mobile Map-Based Applications

Bradley Paul 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 2012

Promoter: Prof. J. L. Wesson
Declaration

I, Bradley Paul van Tonder (Student Number 203004434), hereby declare that in accordance with Rule G4.6.3, this thesis is my own work and that it has not previously been submitted for assessment to another University or for another qualification.

Signature: _________________________

2 April 2012
Acknowledgements

Firstly, I would like to thank my promoter, Prof. Janet Wesson, for her support, direction and encouragement during the course of this research. Her input, advice and ideas have made an invaluable contribution to this thesis.

Thank you to Danie Venter, from the NMMU Unit for Statistical Consultation, who provided valuable assistance and advice regarding the statistical analysis performed during this research.

I would like to thank the NMMU Telkom Centre of Excellence for the funding and equipment which made this research possible. The National Research Foundation and the Nelson Mandela Metropolitan University also provided valuable funding which supported this research.

I would also like to thank the staff and postgraduate students of the Department of Computing Sciences for creating a pleasant and enjoyable atmosphere in which to do research.

Finally, I would like to thank my family for their unwavering support throughout the course of this research.
Summary

Mobile phones are increasingly being equipped with a wide range of sensors which enable a variety of interaction techniques. Sensor-based interaction techniques are particularly promising for domains such as map-based applications, where the user is required to interact with a large information space on the small screen of a mobile phone. Traditional interaction techniques have several shortcomings for interacting with mobile map-based applications. Keypad interaction offers limited control over panning speed and direction. Touch-screen interaction is often a two-handed form of interaction and results in the display being occluded during interaction. Sensor-based interaction provides the potential to address many of these shortcomings, but currently suffers from several limitations. The aim of this research was to propose enhancements to address the shortcomings of sensor-based interaction, with a particular focus on tilt interaction.

A comparative study between tilt and keypad interaction was conducted using a prototype mobile map-based application. This user study was conducted in order to identify shortcomings and opportunities for improving tilt interaction techniques in this domain. Several shortcomings, including controllability, mental demand and practicality concerns were highlighted. Several enhanced tilt interaction techniques were proposed to address these shortcomings. These techniques were the use of visual and vibrotactile feedback, attractors, gesture zooming, sensitivity adaptation and dwell-time selection. The results of a comparative user study showed that the proposed techniques achieved several improvements in terms of the problem areas identified earlier.

The use of sensor fusion for tilt interaction was compared to an accelerometer-only approach which has been widely applied in existing research. This evaluation was motivated by advances in mobile sensor technology which have led to the widespread adoption of digital compass and gyroscope sensors. The results of a comparative user study between sensor fusion and accelerometer-only implementations of tilt interaction showed several advantages for the use of sensor fusion, particularly in a walking context of use.

Modifications to sensitivity adaptation and the use of tilt to perform zooming were also investigated. These modifications were designed to address controllability shortcomings identified in earlier experimental work. The results of a comparison between tilt zooming and
gesture zooming indicated that tilt zooming offered better results, both in terms of performance and subjective user ratings. Modifications to the original sensitivity adaptation algorithm were only partly successful. Greater accuracy improvements were achieved for walking tasks, but the use of dynamic dampening factors was found to be confusing.

The results of this research were used to propose a framework for mobile tilt interaction. This framework provides an overview of the tilt interaction process and highlights how the enhanced techniques proposed in this research can be integrated into the design of tilt interaction techniques. The framework also proposes an application architecture which was implemented as an Application Programming Interface (API). This API was successfully used in the development of two prototype mobile applications incorporating tilt interaction.

**Keywords:** Sensor-based interaction, mobile map-based applications, tilt interaction, sensor fusion, framework
Table of Contents

Declaration................................................................................................................................................i
Acknowledgements .......................................................................................................................................ii
Summary....................................................................................................................................................iii
List of Figures.............................................................................................................................................xvi
List of Tables ...............................................................................................................................................xxi

Chapter 1: Introduction ..........................................................................................................................1
  1.1 Background ........................................................................................................................................1
  1.2 Problem Statement .............................................................................................................................3
  1.3 Thesis Statement .................................................................................................................................4
  1.4 Research Questions ............................................................................................................................4
  1.5 Research Objectives ..........................................................................................................................5
  1.6 Scope ................................................................................................................................................6
  1.7 Research Philosophy and Approach .................................................................................................7
  1.8 Research Methodology .......................................................................................................................8
    1.8.1 Literature Review ..........................................................................................................................8
    1.8.2 Prototyping ....................................................................................................................................8
    1.8.3 Usability Studies ............................................................................................................................9
  1.9 Research Contribution .......................................................................................................................9
  1.10 Research Ethics ...............................................................................................................................10
  1.11 Thesis Structure ..............................................................................................................................10

Chapter 2: Mobile Map-based Applications .........................................................................................14
  2.1 Introduction .......................................................................................................................................14
  2.2 Background .......................................................................................................................................14
  2.3 Tasks ..................................................................................................................................................15
    2.3.1 Searching .....................................................................................................................................16
# Table of Contents

2.3.2 Navigating .................................................................................................................. 17
2.3.3 Identifying .................................................................................................................. 18
2.3.4 Locating ...................................................................................................................... 19
2.3.5 Checking ..................................................................................................................... 19
2.3.6 Interacting .................................................................................................................. 20
2.4 Low-Level Operations .................................................................................................. 20
2.5 Existing Interaction Techniques .................................................................................. 21
2.5.1 Panning ...................................................................................................................... 21
2.5.2 Zooming .................................................................................................................... 24
2.5.3 Combined Zooming and Panning .............................................................................. 26
2.5.4 Selection .................................................................................................................... 27
2.6 Problems and Shortcomings ....................................................................................... 27
2.6.1 Panning ...................................................................................................................... 28
2.6.2 Zooming .................................................................................................................... 29
2.6.3 Selection .................................................................................................................... 29
2.7 Conclusions .................................................................................................................. 30

Chapter 3: Sensor-Based Interaction Techniques .......................................................... 32

3.1 Introduction ................................................................................................................... 32
3.2 Sensors ........................................................................................................................ 32
3.3 Processing Sensor Input ............................................................................................... 33
3.3.1 Sensor Measurement ............................................................................................... 34
3.3.2 Pre-Processing .......................................................................................................... 34
3.3.3 Feature Extraction ................................................................................................... 35
3.3.4 Mapping onto Operations ....................................................................................... 36
3.3.5 Performing Operations ......................................................................................... 36
3.4 Existing Techniques ..................................................................................................... 36
3.4.1 Panning ...................................................................................................................... 36
# Table of Contents

3.4.2 Zooming ................................................................. 41
3.4.3 Combined Panning and Zooming ................................. 42
3.4.4 Selection .................................................................. 44
3.5 Benefits ..................................................................... 45
3.6 Problems and Shortcomings ........................................... 46
  3.6.1 Physical Limitations ................................................ 46
  3.6.2 Toggling Sensor-based Control ................................. 47
  3.6.3 Establishing a Neutral State ...................................... 48
  3.6.4 Display Visibility .................................................... 49
  3.6.5 Controllability ........................................................ 49
  3.6.6 Summary .............................................................. 51
3.7 Tilt Sensing ................................................................. 52
  3.7.1 Accelerometers ..................................................... 52
  3.7.2 Digital Compasses ................................................ 53
  3.7.3 Gyroscopes .......................................................... 53
3.8 Existing Frameworks .................................................... 54
3.9 Existing APIs ............................................................... 55
  3.9.1 Java ME ............................................................... 55
  3.9.2 Symbian C++ ....................................................... 56
  3.9.3 Android .............................................................. 56
  3.9.4 iOS ..................................................................... 57
  3.9.5 Windows Phone .................................................... 57
  3.9.6 Summary .............................................................. 57
3.10 Conclusions ............................................................... 58

**Chapter 4: Designing a Basic Tilt Interaction Technique** ................................................. 60
4.1 Introduction ............................................................... 60
4.2 Background ............................................................... 60
Table of Contents

4.3 Tilt Interaction Technique ................................................................................. 61
  4.3.1 Sensor Measurement .................................................................................. 62
  4.3.2 Pre-Processing ............................................................................................ 63
  4.3.3 Feature Extraction ....................................................................................... 63
  4.3.4 Mapping to Operations ............................................................................... 63
  4.3.5 Performing Operations ............................................................................... 64
  4.3.6 Rendering Display ...................................................................................... 67

4.4 Prototype: MapExplorer .................................................................................. 67
  4.4.1 Identifying ................................................................................................. 67
  4.4.2 Locating ...................................................................................................... 68
  4.4.3 Checking ..................................................................................................... 68
  4.4.4 Navigating .................................................................................................. 68

4.5 Keypad Interaction ........................................................................................... 69
  4.5.1 Panning ....................................................................................................... 69
  4.5.2 Zooming ...................................................................................................... 70
  4.5.3 Selection ...................................................................................................... 70

4.6 Tilt Interaction .................................................................................................. 70
  4.6.1 Panning ....................................................................................................... 70
  4.6.2 Zooming ...................................................................................................... 71
  4.6.3 Selection ...................................................................................................... 72

4.7 Experimental Design ......................................................................................... 73
  4.7.1 Objectives ................................................................................................... 73
  4.7.2 Metrics ......................................................................................................... 73
  4.7.3 Participants ................................................................................................. 74
  4.7.4 Tasks ............................................................................................................ 74
  4.7.5 Independent Variables ............................................................................... 75
  4.7.6 Test Environment ....................................................................................... 75
Table of Contents

4.7.7 Materials ................................................................................................................................. 76
4.7.8 Procedure ................................................................................................................................. 76
4.8 Results: Self-Reported Metrics ................................................................................................. 77
  4.8.1 Locating Tasks ......................................................................................................................... 77
  4.8.2 Navigating Tasks ..................................................................................................................... 78
  4.8.3 Checking Tasks ....................................................................................................................... 79
  4.8.4 Overall Satisfaction ............................................................................................................... 80
4.9 Results: Performance Metrics .................................................................................................. 83
  4.9.1 Locating Tasks ......................................................................................................................... 83
  4.9.2 Navigating Tasks ..................................................................................................................... 83
  4.9.3 Checking Tasks ....................................................................................................................... 84
  4.9.4 Overall .................................................................................................................................. 84
4.10 Results: Correlations ............................................................................................................... 84
4.11 Discussion ............................................................................................................................... 86
4.12 Improving Tilt Interaction ....................................................................................................... 89
  4.12.1 Controllability ....................................................................................................................... 90
  4.12.2 Precise Selection ................................................................................................................... 91
  4.12.3 Zooming Technique ............................................................................................................. 91
  4.12.4 Mental Demand .................................................................................................................... 92
  4.12.5 Practicality ........................................................................................................................... 92
4.13 Discussion ............................................................................................................................... 93
4.14 Conclusions ............................................................................................................................. 93

Chapter 5: Proposed Enhanced Tilt Interaction Techniques ......................................................... 96
5.1 Introduction ............................................................................................................................... 96
5.2 Proposed Techniques ............................................................................................................... 96
  5.2.1 Visual Feedback .................................................................................................................... 97
  5.2.2 Vibrotactile Feedback .......................................................................................................... 99
5.2.3 Attractor Mechanisms ................................................................. 100
5.2.4 Sensitivity Adaptation .............................................................. 101
5.2.5 Gesture Zooming ................................................................. 103
5.2.6 Dwell-Time Selection ............................................................. 105
5.2.7 Algorithm Design ................................................................. 106
5.3 Implementation ........................................................................... 107
5.3.1 Visual and Vibrotactile Feedback ............................................. 107
5.3.2 Attractor Mechanisms ............................................................. 108
5.3.3 Sensitivity Adaptation .............................................................. 108
5.3.4 Gesture Zooming ................................................................. 109
5.3.5 Dwell-Time Selection ............................................................. 109
5.4 Experimental Design .................................................................. 109
5.4.1 Objectives .............................................................................. 109
5.4.2 Hypotheses ............................................................................ 110
5.4.3 Metrics .................................................................................. 110
5.4.4 Participants ........................................................................... 111
5.4.5 Tasks ..................................................................................... 111
5.4.6 Independent Variables ............................................................ 112
5.4.7 Test Environment ................................................................. 112
5.4.8 Materials .............................................................................. 112
5.4.9 Procedure ............................................................................. 112
5.5 Results ....................................................................................... 113
5.5.1 Performance Results ............................................................... 113
5.5.2 Perceived Workload Results .................................................... 118
5.5.3 User Satisfaction Results ......................................................... 118
5.5.4 Qualitative Feedback Results .................................................. 120
5.5.5 Discussion .............................................................................. 122
## Table of Contents

5.6 Hypothesis Testing ................................................................. 124
5.7 Design Implications .............................................................. 126
5.8 Discussion ........................................................................... 127
5.9 Conclusions ........................................................................ 128

**Chapter 6: Using Sensor Fusion to Improve Tilt Interaction** .................. 130

6.1 Introduction .......................................................................... 130
6.2 Motivation ........................................................................... 131
6.3 Design of Sensor Fusion Approach ......................................... 131
6.4 Accelerometer-Only Implementation ....................................... 135
6.5 MapExplorer Modifications .................................................. 136
  6.5.1 Android Implementation .................................................. 136
  6.5.2 Tilt Zooming ................................................................. 137
6.6 Experimental Design ............................................................. 139
  6.6.1 Objectives ...................................................................... 139
  6.6.2 Hypotheses .................................................................... 139
  6.6.3 Metrics .......................................................................... 140
  6.6.4 Participants .................................................................... 140
  6.6.5 Tasks ............................................................................ 141
  6.6.6 Independent Variables .................................................... 141
  6.6.7 Test Environment .......................................................... 142
  6.6.8 Materials ....................................................................... 142
  6.6.9 Procedure ...................................................................... 142
6.7 Results .................................................................................. 142
  6.7.1 Performance Results ....................................................... 143
  6.7.2 Perceived Controllability Results ..................................... 148
6.8 Discussion ............................................................................ 149
6.9 Hypothesis Testing .............................................................. 151
<table>
<thead>
<tr>
<th>Chapter 6: Design Implications</th>
<th>152</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.10 Design Implications</td>
<td>152</td>
</tr>
<tr>
<td>6.11 Discussion</td>
<td>152</td>
</tr>
<tr>
<td>6.12 Conclusions</td>
<td>153</td>
</tr>
</tbody>
</table>

**Chapter 7: Improving the Controllability of Tilt Interaction**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1 Introduction</td>
<td>155</td>
</tr>
<tr>
<td>7.2 Zooming Techniques</td>
<td>155</td>
</tr>
<tr>
<td>7.3 Gesture Zooming Modifications</td>
<td>156</td>
</tr>
<tr>
<td>7.4 Tilt Zooming Modifications</td>
<td>158</td>
</tr>
<tr>
<td>7.5 Overview</td>
<td>160</td>
</tr>
<tr>
<td>7.6 Sensitivity Adaptation</td>
<td>161</td>
</tr>
<tr>
<td>7.6.1 Training</td>
<td>161</td>
</tr>
<tr>
<td>7.6.2 Static Sensitivity Adaptation</td>
<td>165</td>
</tr>
<tr>
<td>7.6.3 Dynamic Sensitivity Adaptation</td>
<td>166</td>
</tr>
<tr>
<td>7.7 Experimental Design</td>
<td>167</td>
</tr>
<tr>
<td>7.7.1 Objectives</td>
<td>167</td>
</tr>
<tr>
<td>7.7.2 Hypotheses</td>
<td>167</td>
</tr>
<tr>
<td>7.7.3 Metrics</td>
<td>168</td>
</tr>
<tr>
<td>7.7.4 Participants</td>
<td>169</td>
</tr>
<tr>
<td>7.7.5 Tasks</td>
<td>169</td>
</tr>
<tr>
<td>7.7.6 Independent Variables</td>
<td>171</td>
</tr>
<tr>
<td>7.7.7 Test Environment</td>
<td>172</td>
</tr>
<tr>
<td>7.7.8 Materials</td>
<td>172</td>
</tr>
<tr>
<td>7.7.9 Procedure</td>
<td>172</td>
</tr>
<tr>
<td>7.8 Results: Zooming Techniques</td>
<td>173</td>
</tr>
<tr>
<td>7.8.1 Efficiency</td>
<td>173</td>
</tr>
<tr>
<td>7.8.2 Workload</td>
<td>174</td>
</tr>
<tr>
<td>7.8.3 User Satisfaction</td>
<td>176</td>
</tr>
</tbody>
</table>
# Table of Contents

7.8.4 Preferences ................................................................. 178  
7.8.5 Qualitative Feedback .................................................. 179  
7.9 Results: Sensitivity Adaptation ........................................... 181  
  7.9.1 Accuracy ................................................................. 181  
  7.9.2 Efficiency ............................................................... 183  
  7.9.3 User Satisfaction ...................................................... 184  
  7.9.4 Qualitative Feedback .................................................. 185  
7.10 Objective vs. Subjective Data ............................................ 185  
7.11 Dynamic Adaptation ....................................................... 188  
7.12 Discussion ...................................................................... 189  
  7.12.1 Zooming Techniques .................................................. 189  
  7.12.2 Sensitivity Adaptation .................................................. 191  
7.13 Hypothesis Testing ........................................................... 193  
  7.13.1 Zooming Technique .................................................... 193  
  7.13.2 Sensitivity Adaptation .................................................. 194  
7.14 Design Implications .......................................................... 196  
7.15 Discussion ...................................................................... 197  
7.16 Conclusions ................................................................... 198  

## Chapter 8: A Mobile Tilt Interaction Framework .................................................. 201  
  8.1 Introduction .................................................................... 201  
  8.2 Design Implications Revisited ............................................ 201  
  8.3 MoTIF: A Mobile Tilt Interaction Framework ......................... 202  
    8.3.1 Sensor Measurement, Pre-Processing and Feature Extraction ........ 204  
    8.3.2 Mapping Tilt Input onto Panning and Zooming Operations .......... 205  
    8.3.3 Performing Panning .................................................. 207  
    8.3.4 Performing Zooming .................................................. 208  
    8.3.5 Performing Selection .................................................. 209
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.3.6</td>
<td>Updating the Display</td>
<td>209</td>
</tr>
<tr>
<td>8.3.7</td>
<td>Supporting Evidence</td>
<td>210</td>
</tr>
<tr>
<td>8.4</td>
<td>Framework Architecture</td>
<td>210</td>
</tr>
<tr>
<td>8.4.1</td>
<td>Sensor Data Provider</td>
<td>211</td>
</tr>
<tr>
<td>8.4.2</td>
<td>Context Calculator</td>
<td>212</td>
</tr>
<tr>
<td>8.4.3</td>
<td>Position Listener</td>
<td>212</td>
</tr>
<tr>
<td>8.5</td>
<td>Framework Implementation</td>
<td>213</td>
</tr>
<tr>
<td>8.5.1</td>
<td>API Overview</td>
<td>213</td>
</tr>
<tr>
<td>8.5.2</td>
<td>API Implementations</td>
<td>215</td>
</tr>
<tr>
<td>8.6</td>
<td>Discussion</td>
<td>216</td>
</tr>
<tr>
<td>8.7</td>
<td>Conclusions</td>
<td>216</td>
</tr>
<tr>
<td>9.1</td>
<td>Introduction</td>
<td>218</td>
</tr>
<tr>
<td>9.2</td>
<td>Summary of Findings</td>
<td>218</td>
</tr>
<tr>
<td>9.2.1</td>
<td>Existing Interaction Techniques</td>
<td>218</td>
</tr>
<tr>
<td>9.2.2</td>
<td>Existing Sensor-Based Interaction Techniques</td>
<td>219</td>
</tr>
<tr>
<td>9.2.3</td>
<td>Keypad and Tilt Interaction</td>
<td>221</td>
</tr>
<tr>
<td>9.2.4</td>
<td>Proposed Techniques</td>
<td>222</td>
</tr>
<tr>
<td>9.2.5</td>
<td>Comparison of Existing and Proposed Techniques</td>
<td>224</td>
</tr>
<tr>
<td>9.2.6</td>
<td>Accelerometer-Only and Sensor Fusion Tilt Interaction</td>
<td>226</td>
</tr>
<tr>
<td>9.2.7</td>
<td>Proposed Framework</td>
<td>227</td>
</tr>
<tr>
<td>9.3</td>
<td>Summary of Contributions</td>
<td>229</td>
</tr>
<tr>
<td>9.4</td>
<td>Limitations and Problems Encountered</td>
<td>233</td>
</tr>
<tr>
<td>9.5</td>
<td>Recommendations for Future Work</td>
<td>233</td>
</tr>
<tr>
<td></td>
<td>References</td>
<td>236</td>
</tr>
<tr>
<td></td>
<td>Appendix A: Informed Consent Form</td>
<td>248</td>
</tr>
<tr>
<td></td>
<td>Appendix B: Pre-Test Biographical Questionnaire</td>
<td>250</td>
</tr>
</tbody>
</table>
List of Figures

Figure 1.1. Overview of thesis structure

Figure 2.1. Main menus of Nokia Maps (left) and Google Maps (right)

Figure 2.2. Searching in Nokia Maps (left) and Google Maps (right)

Figure 2.3. Navigating tasks in Google Maps

Figure 2.4. Locating tasks in Nokia Maps (left) and Google Maps (right)

Figure 2.5. Checking POI details in Google Maps

Figure 2.6. Interface for interacting with POIs in Nokia Maps

Figure 2.7. On-screen controls for zooming and scrollbars for panning

Figure 2.8. Dragging panning gesture

Figure 2.9. Flicking panning gesture

Figure 2.10. Touch-screen zooming using gestures

Figure 2.11. Overview + detail mobile map-based visualisation

Figure 2.12. Example of touch-screen input occluding the display

Figure 3.1. Stages in processing sensor data for sensor-based interaction

Figure 3.2. Smoothing accelerometer data with a Savitsky-Golay filter

Figure 3.3. Tilt gestures for two-dimensional panning

Figure 3.4. Position control for menu navigation

Figure 3.5. Mapping between tilt angles and panning speeds

Figure 3.6. Linear, quadratic and sigmoidal discretisation

Figure 3.7. Camera-based interaction using a grid (left) and static map (right)

Figure 3.8. Tilt zooming in the Samsung Galaxy S2

Figure 3.9. Tilt-controlled SDAZ in a mobile document browser
Figure 3.10. Semi-Automatic Zooming using tilt interaction (Kratz and Rohs 2008) ..........44
Figure 3.11. Maximum angle of rotation possible using different wrist-based gestures (Rahman et al. 2009) ........................................................................................................47
Figure 3.12. Using a button as a clutch for tilt interaction (Rath and Rohs 2006) ..........48
Figure 3.13. X, Y and Z axes, relative to which acceleration and orientation are calculated..52
Figure 4.1. The different stages involved in using tilt interaction to perform panning and zooming ........................................................................................................................................62
Figure 4.2. State transition diagram for the implementation of SDAZ .........................66
Figure 4.3. Asymmetric hysteresis..................................................................................66
Figure 4.4. POI icons in MapExplorer ............................................................................68
Figure 4.5. Viewing detailed information for a POI.......................................................68
Figure 4.6. Navigation in MapExplorer ..........................................................................69
Figure 4.7. Performing panning using directional keys in MapExplorer .......................70
Figure 4.8. Panning at low speed ....................................................................................71
Figure 4.9. Panning at medium speed .............................................................................72
Figure 4.10. Panning at high speed ................................................................................72
Figure 4.11. Evaluation participant using tilt interaction in MapExplorer .......................75
Figure 4.12. Task instructions presented to participants...............................................76
Figure 4.13. Confirmation of task completion....................................................................77
Figure 4.14. Distribution of ratings regarding participants' preferred interaction technique \((n=32)\)................................................................................................................80
Figure 4.15. Mean interval scale ratings for post-test questionnaire (95% confidence intervals shown) \((n=32)\)..............................................................................................................81
Figure 4.16. Mean total task times organised by task type (95% confidence intervals shown) \((n=32)\)................................................................................................................84
Figure 5.1. Visual feedback .............................................................................................99
Figure 5.2. Example of the use of attractors to aid selection.........................................101
Figure 5.3. Comparison of acceleration variances when seated (left) and walking (right) ...102
Figure 5.4. Sensitivity settings when the user is (a) stationary and (b) mobile ..............103
List of Figures

Figure 5.5. Zooming out using gesture zooming ................................................................. 104
Figure 5.6. Z-axis acceleration data showing zooming gestures ......................................... 105
Figure 5.7. Z-acceleration plot showing typical zooming out (left) and zooming in (right) z- acceleration patterns ................................................................. 105
Figure 5.8. Enhanced tilt interaction algorithm design .......................................................... 107
Figure 5.9. Visual feedback in MapExplorer ........................................................................... 108
Figure 5.10. Route plots for participants performing route-following without sensitivity adaptation (left) and with sensitivity adaptation (right). The actual route is shown in blue .............................................................................................................................. 115
Figure 5.11. Comparison of selection using tilt without attractors (left) and with attractors (right) .................................................................................................................. 116
Figure 5.12. Example of an overshooting error during icon selection ...................................... 117
Figure 5.13. Mean perceived workload ratings (95% confidence intervals shown) (n = 16) 118
Figure 5.14. Mean user satisfaction interval scale ratings (95% confidence intervals shown) (n = 16) ......................................................................................................................... 119
Figure 6.1. Illustration of gyroscopic drift. The grey line indicates drifting gyroscope-only data, while the black line shows stable fused data ................................................................................. 133
Figure 6.2. Sensor fusion algorithm used in MapExplorer ...................................................... 134
Figure 6.3. Comparison of pitch angle output for accelerometer approach (grey) and sensor fusion approach (black) when using tilt interaction while climbing stairs ........................................................................ 135
Figure 6.4. Android implementation of MapExplorer ................................................................ 137
Figure 6.5. Tilt zooming in MapExplorer. The button used to activate tilt zooming is indicated ................................................................................................................................. 138
Figure 6.6. Example of a navigation task in MapExplorer ....................................................... 141
Figure 6.7. Comparison of accelerometer (left) and sensor fusion (right) route plots of a routing task performed while walking. The actual route is shown as a thin blue line .............................................. 144
Figure 6.8. Comparison of mean variance for pitch and roll angles (where angles were measured in degrees) for sensor fusion and accelerometer implementations. 95% confidence intervals shown (n = 17) ....................................................................................................................... 146
Figure 6.9. Comparison of sensor fusion and accelerometer output for a walking navigation task ........................................................................................................................................... 147
Figure 6.10. Comparison of mean variance for latitude and longitude for sensor fusion and accelerometer implementations ................................................................. 147
Figure 7.1. Correctly identified zooming gesture (left), erroneously identified gesture (right) .............................................................. 157

Figure 7.2. Visual feedback when performing gesture zooming .............................................................. 158

Figure 7.3. Performing tilt zooming in MapExplorer .............................................................. 159

Figure 7.4. Overview in MapExplorer (top right corner) .............................................................. 160

Figure 7.5. Artificial Neural Network architecture used to distinguish walking and seated contexts of use. The input layer is simplified for readability. .............................................................. 163

Figure 7.6. Static sensitivity adaptation .............................................................................................. 165

Figure 7.7. Example zooming task in MapExplorer ............................................................................ 170

Figure 7.8. Route-following tasks in MapExplorer ............................................................................ 171

Figure 7.9. Mean task times for gesture and tilt zooming techniques. Error bars show 95% confidence intervals (n=30). .............................................................................................. 173

Figure 7.10. Mean perceived workload ratings for seated tasks. Errors bars show 95% confidence intervals (n=30). .............................................................................................. 174

Figure 7.11. Perceived workload ratings - walking tasks (n=30) ........................................................ 175

Figure 7.12. Mean user satisfaction ratings for zooming techniques for seated tasks (n=30) .............................................................................................. 176

Figure 7.13. Mean user satisfaction ratings for walking tasks (n=30) ................................................ 177

Figure 7.14. Participant zooming technique preferences (n=30) ....................................................... 179

Figure 7.15. Mean deviation from planned routes (in pixels) (n=30) ................................................ 182

Figure 7.16. Comparison of route plots for all participants for the same route. Plots for participants using the no adaptation setting shown left, static adaptation shown centre and dynamic adaptation shown right. .............................................................................................. 183

Figure 7.17. Mean task times for different sensitivity adaptation settings (n=30) ................................ 183

Figure 7.18. Mean user satisfaction ratings for sensitivity adaptation. 95% confidence intervals shown (n=30). .............................................................................................. 184

Figure 7.19. Correlation between walking speed and accuracy for no adaptation (n=30) .... 186

Figure 7.20. Correlation between walking speed and accuracy for static adaptation (n=30) 187

Figure 7.21. Correlation between walking speed and accuracy for dynamic adaptation (n=30) .............................................................................................. 187

Figure 7.22. Plot of dampening factor vs. time for one participant using dynamic adaptation .............................................................................................. 189
List of Figures

Figure 8.1. Procedural view of the Mobile Tilt Interaction Framework (MoTIF) ...............203
Figure 8.2. Combined sensor measurement, pre-processing and feature extraction stages of MoTIF .............................................................................................................. 204
Figure 8.3. Mapping between tilt angles and panning and zooming speeds .................. 205
Figure 8.4. Performing panning in MoTIF ........................................................................ 207
Figure 8.5. Zooming stage in MoTIF ................................................................................. 208
Figure 8.6. Selection stage in MoTIF ................................................................................ 209
Figure 8.7. Updating and rendering the display in MoTIF ............................................ 210
Figure 8.8. MoTIF architecture ......................................................................................... 211
Figure 8.9. Simplified class diagram of application implementing the MoTIF API (without parameters) ........................................................................................................ 214
Figure 8.10. PhotoWave photo gallery application using tilt interaction (Leonard 2011) .... 215
List of Tables

Table 1.1. Research questions to be addressed in this thesis .......................................... 4
Table 2.1. Breakdown of the low-level operations required by each task type ............... 21
Table 4.1 Discretisation of tilt angles and corresponding panning speeds ................. 64
Table 4.2. Keypad and tilt interaction support for panning, zooming and selection .......... 72
Table 4.3. Results of t-tests comparing keypad and tilt interaction for locating tasks (bold p-values are statistically significant) (n=32) .................................................. 78
Table 4.4. Results of t-tests comparing keypad and tilt interaction for navigating tasks (bold p-values denote statistical significance) (n=32) .......................................... 79
Table 4.5. Results of t-tests comparing keypad and tilt interaction for checking tasks (bold p-values denote statistical significance) (n=32) ....................................... 79
Table 4.6. Results of t-tests comparing overall controllability, efficiency, effectiveness and ease of use for keypad and tilt interaction (bold p-values denote statistical significance) (n=32) ................................................................................................. 81
Table 4.7. Frequencies of top three positive and negative themes for each keypad interaction ................................................................................................................................... 82
Table 4.8. Frequencies of top three positive and negative themes for tilt interaction ...... 83
Table 4.9. Results of t-tests comparing mean total task times for each group of tasks for keypad and tilt interaction (n=32) ................................................................. 83
Table 4.10. Preferred interaction technique vs. previous experience with tilt interaction (1 participant indicated a neutral preference rating and is not included) .............. 85
Table 4.11. Preferred interaction technique vs. previous experience with tilt interaction on a mobile phone ....................................................................................................... 85
Table 4.12. Correlations between task times and pitch and roll angles ......................... 86
Table 4.13. Usability issues regarding tilt interaction ...................................................... 89
Table 5.1. Usability problems of tilt interaction and corresponding proposed techniques ...... 97
Table 5.2. Wilcoxon signed-rank test results comparing mean task times (in seconds) for seated and walking tasks (n = 16) ................................................................. 113
Table 5.3. Wilcoxon signed-rank test results comparing mean task times (in seconds) for different task types (n = 16) ................................................................. 114
Table 5.4. Wilcoxon signed-rank test results comparing mean accuracy for navigating tasks (measured as mean deviation from planned route in pixels) \((n = 16)\) ........................................14

Table 5.5. Number of participants employing each selection method employed for checking tasks \((n = 16)\) .................................................................144

Table 5.6. Wilcoxon results comparing mean distances (in pixels) from POI icons when performing selection operations \((n = 16)\) .................................................................145

Table 5.7. Wilcoxon results comparing mean number of crossings for checking tasks \((n = 16)\) .................................................................146

Table 5.8. Wilcoxon results comparing mean time (in seconds) between entering a POI region and settling on the icon for checking tasks \((n = 16)\) .................................................................147

Table 5.9. Wilcoxon results comparing participant post-test preference ratings to intermediate value of 4 \((1 = \text{Basic tilt}, 7 = \text{Enhanced tilt})\) \((n = 16)\) .................................................................148

Table 5.10. Wilcoxon results comparing feedback usefulness ratings to intermediate value of 4 \((1 = \text{Strongly Disagree}, 7 = \text{Strongly Agree})\) \((n = 16)\) .................................................................149

Table 5.11. Frequencies of positive and negative issues for basic tilt interaction ........................................150

Table 5.12. Frequencies of positive and negative issues for enhanced tilt interaction ........................................151

Table 5.13. Summary of hypothesis testing results ........................................................................152

Table 6.1. Wilcoxon signed-rank test results comparing mean task times (in seconds) for the two implementations \((n = 17)\) ................................................................................163

Table 6.2. Wilcoxon results comparing mean task times (in seconds) organised by task type \((n = 17)\) ................................................................................164

Table 6.3. Wilcoxon results comparing mean accuracy for navigating tasks. Deviation from the actual route reported in pixels \((n = 17)\) ................................................................................165

Table 6.4. Wilcoxon results comparing mean number of crossings per selection \((n = 17)\) ................................................................................166

Table 6.5. Wilcoxon results comparing mean time (in seconds) between entering the region of an icon and performing a selection operation \((n = 17)\) ................................................................................167

Table 6.6. Questionnaire used to measure perceived controllability for different task types and contexts of use ................................................................................168

Table 6.7. Wilcoxon results comparing mean post-test questionnaire perceived controllability ratings. Seven point interval scales were used, where 1 = Accelerometer and 7 = Sensor Fusion \((n = 17)\) ................................................................................169

Table 6.8. Hypothesis testing results for perceived controllability ................................................................170

Table 6.9. Hypothesis testing results for controllability ................................................................................171

Table 6.10. Hypothesis testing results for efficiency ................................................................................172
Table 7.1. Comparison of zooming technique approaches .............................................156
Table 7.2. Example of dampening factor calculation using dynamic adaptation. The largest ratio (x-axis) shown in bold is used as the current dampening factor in this example..167
Table 7.3. Results of t-tests comparing mean task times (in seconds) for gesture and tilt zooming (n=30)..................................................................................................................174
Table 7.4. Results of t-tests comparing mean perceived workload ratings for seated tasks (n=30)..........................................................................................................................175
Table 7.5. Results of t-tests comparing mean perceived workload ratings for walking tasks (n=30).........................................................................................................................176
Table 7.6. Results of t-tests comparing mean user satisfaction ratings for zooming tasks while seated (n=30)................................................................................................................177
Table 7.7. Results of t-tests comparing mean user satisfaction ratings for zooming techniques for walking tasks (n=30)..........................................................................................178
Table 7.8. Results of t-tests comparing mean overview usefulness ratings to intermediate value of 4 (n=30)...........................................................................................................178
Table 7.9. Results of t-tests for statistical significance for user preference ratings (n=30)...179
Table 7.10. Frequency of most common positive themes regarding gesture zooming.........180
Table 7.11. Frequency of most common negative themes regarding gesture zooming.......180
Table 7.12. Frequency of the most common positive themes regarding tilt zooming........181
Table 7.13. Frequency of the most common negative themes regarding tilt zooming.......181
Table 7.14. Results of Tukey HSD post-hoc tests comparing accuracy of individual pairs of sensitivity adaptation settings (n=30) ..................................................................................182
Table 7.15. Results of Tukey post-hoc tests comparing mean effectiveness ratings for different sensitivity adaptation settings (n=30) ...............................................................185
Table 7.16. Results of Tukey post-hoc tests comparing mean ease of use ratings for different sensitivity adaptation settings (n=30) ........................................................................185
Table 7.17. Results of Tukey post-hoc tests comparing mean controllability ratings for different sensitivity adaptation settings (n=30) .................................................................185
Table 7.18. Correlations between perceived and actual sensitivity adaptation efficiency and accuracy (n=30) .............................................................................................................188
Table 7.19. Zooming technique hypothesis testing results ..............................................194
Table 7.20. Sensitivity adaptation hypothesis testing results .........................................196
Table 8.1. Design recommendations for tilt interaction in mobile map-based applications..202
Table 8.2. Proposed/modified aspects of tilt interaction and supporting evidence .............210
Table 9.1. Design recommendations for tilt interaction in mobile map-based applications..228
Chapter 1: Introduction

1.1 Background

Mobile interaction techniques have evolved to keep pace with changes in the functionality provided by mobile phones. Mobile phones started out as tools for making phone calls and later, sending and receiving text messages. Keypad interaction was well-suited to these functions (Hansen, Eriksson and Lykke-Olesen 2005). In the last decade, mobile phones have evolved into devices offering a wide range of functionality. Smart phones today offer not only phone functionality, but also allow the user to browse the Internet, capture and browse photos and videos, listen to music, edit documents, play games and interact with maps. Mobile interaction techniques have evolved in order to meet the interaction needs of these applications (Hughes, Oakley and O’Modhrain 2004, Oakley and Park 2009). Sensor-based interaction techniques, which use physical movement of the mobile phone as a form of input, represent one such evolution (Yim, Lee and Choi 2011).

Many mobile phones are now equipped with a range of sensors which enable the orientation and movement of the phone to be detected. Accelerometers, digital compasses and gyroscopes can all be used to retrieve information about movement and orientation changes of the mobile phone (Ruiz, Li and Lank 2011). This sensing capability has enabled the development of intuitive interaction techniques relying on physical gestures. Sensor-based interaction techniques have mostly been applied in domains where the user is required to browse large information spaces on mobile phones, including menus, documents, maps and photo collections (Oakley and O’Modhrain 2005, Oakley and Park 2009, Eslambolchilar, Williamson and Murray-Smith 2004, Yim et al. 2011, Kratz, Brodien and Rohs 2010). This research will focus on the domain of mobile map-based applications.

Mobile map-based applications are now standard features of almost all smart phones. Users of mobile map-based applications typically have access to a wide range of functionality, including browsing maps, planning routes and viewing information regarding points of interest (Looije, te Brake and Neerinex 2007). These applications take advantage of the portable nature and sensing capabilities of mobile phones to deliver information and services relevant to the user’s current location and context. Historically, keypad and touch-screen interaction techniques have been employed to facilitate interaction with mobile map-based applications (Setlur, Kuo and Mikelsons 2010).
Existing interaction techniques are not well-suited to interacting with mobile map-based applications (Kratz et al. 2010). Keypad interaction, while easy to control and familiar to users, has been described as a frustrating form of input for panning long distances (Winkler, Rangaswamy and Zhou 2007, Cho et al. 2007b). Possible reasons include a lack of precise speed control and constrained freedom of movement (Kratz and Rohs 2008). Touch-screen interaction overcomes some of the limitations of keypad interaction by employing an expressive array of touch-screen gestures to facilitate interaction. Touch-screen interaction, however, frequently requires the use of both hands to perform operations such as zooming and selection, with one hand required to hold the device and the other hand used to perform the required gesture (Cho et al. 2007b). One-handed interaction techniques have previously been identified as being preferable in mobile contexts of use (Pascoe, Ryan and Morse 2000, Karlson, Bederson and Contreras-Vidal 2006). Touch-screen techniques also result in the user’s hand occluding the display during interaction. Furthermore, accurate selection is often difficult using touch-screen interaction without a stylus (Park and Han 2010). Sensor-based interaction techniques provide the possibility of addressing some of these shortcomings.

Sensor-based interaction techniques are often described as intuitive and rely on natural physical gestures to interact with mobile applications (Rohs and Essl 2007, Weberg, Brange and Hansson 2001, Bartlett 2000). Both continuous and discrete forms of interaction are possible using sensor-based techniques, providing many possibilities for the design of expressive and innovative interaction techniques (Crossan et al. 2008). Existing techniques frequently employ tilting gestures to perform panning and zooming operations (Dong, Watters and Duffy 2005). Sensor-based interaction techniques allow for one-handed interaction, as the hand used to hold the mobile phone is also used to perform interaction (Dong et al. 2005). This has the added advantage of leaving the display unobscured during interaction (Cho et al. 2007b). Despite the potential advantages, existing sensor-based interaction techniques have several usability shortcomings which have hindered their widespread adoption.

Usability is defined as (International Standards Organisation (ISO) 1997):

“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
Sensor-based interaction techniques have been described as difficult to control. Overshooting problems, where users struggle to perform tasks that require exact positioning, have previously been observed (Cho et al. 2007b). Designers of sensor-based interaction techniques also have to consider a range of practical concerns, including how to avoid unintentional interaction and contend with varying contexts of use (Rahman et al. 2009). This research will investigate how the usability shortcomings of sensor-based interaction for mobile map-based applications can be addressed.

1.2 Problem Statement

Current mainstream interaction techniques supported by mobile devices are not well suited to interacting with mobile map-based applications. Keypad interaction techniques offer limited control over panning speed and constrained freedom of movement. Touch-screen interaction techniques require the use of both hands, which is not always desirable in a mobile context of use (Karlson et al. 2006). Touch-screen interaction also results in the display being obscured during interaction and makes precise selection difficult (Fitchett and Cockburn 2009, Park and Han 2010).

Sensor-based interaction techniques, particularly those incorporating continuous tilt gestures, provide the potential to improve on existing interaction techniques by facilitating intuitive, one-handed interaction (Oakley and O’Modhrain 2005, Crossan and Murray-Smith 2004). Tilt interaction has been used to perform panning (Dong et al. 2005), zooming (Büring, Gerken and Reiterer 2008) and combined panning and zooming (Eslambolchilar and Murray-Smith 2004, Kratz and Rohs 2008) in mobile map-based applications. Existing approaches, however, suffer from several shortcomings. Sensor-based interaction has been described as providing too coarse a level of control to be used as a general form of input (Karlson et al. 2006). Sensor-based interaction has also been reported to suffer from controllability problems. Users often struggle to exercise precise control over the cursor position, making selection operations prone to overshooting errors (Cho, Murray-Smith and Kim 2007a, Hinckley and Song 2011). Furthermore, several existing sensor-based interaction techniques have been shown to be difficult to control while walking (Wilson et al. 2011, Crossan et al. 2009, Crossan et al. 2008).
1.3 **Thesis Statement**

The thesis statement of this research is:

*Enhanced interaction techniques can be used to improve the usability of sensor-based interaction for mobile map-based applications.*

This research will identify usability shortcomings of existing sensor-based interaction techniques by comparing sensor-based interaction with an established one-handed interaction technique. Appropriate enhanced techniques will then be proposed to address the usability shortcomings identified. Comparative usability studies will then be conducted to identify whether the proposed techniques improve the usability of sensor-based interaction for mobile map-based applications.

1.4 **Research Questions**

Following from the thesis statement, the primary research question to be addressed by this research is: *How can enhanced interaction techniques be used to improve the usability of sensor-based interaction for mobile map-based applications?*

<table>
<thead>
<tr>
<th>#</th>
<th>Research Question</th>
<th>Method(s)</th>
<th>Chapter(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>What are the shortcomings of existing interaction techniques for mobile map-based applications?</td>
<td>Literature Review</td>
<td>2</td>
</tr>
<tr>
<td>2.</td>
<td>What are the benefits and shortcomings of existing sensor-based interaction techniques?</td>
<td>Literature Review</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>What are the shortcomings of existing sensor-based interaction techniques for mobile map-based applications?</td>
<td>Prototyping and Usability Study</td>
<td>4</td>
</tr>
<tr>
<td>4.</td>
<td>What techniques can be used to address the shortcomings of existing sensor-based interaction techniques?</td>
<td>Prototyping</td>
<td>5, 7</td>
</tr>
<tr>
<td>5.</td>
<td>To what extent do the proposed techniques address the shortcomings of sensor-based interaction?</td>
<td>Usability Study</td>
<td>5, 7</td>
</tr>
<tr>
<td>6.</td>
<td>To what extent does sensor fusion address the shortcomings of sensor-based interaction?</td>
<td>Prototyping and Usability Study</td>
<td>6</td>
</tr>
<tr>
<td>7.</td>
<td>How can the proposed enhanced techniques be integrated into the design of sensor-based interaction techniques?</td>
<td>Deductive Reasoning</td>
<td>8</td>
</tr>
</tbody>
</table>

*Table 1.1. Research questions to be addressed in this thesis*
In order to answer these questions, several secondary research questions were identified. These questions, along with the associated research methods and the chapters where they are addressed, are outlined in Table 1.1. The research methods to be used to answer these research questions are discussed in more detail in Section 1.8.

### 1.5 Research Objectives

The primary aim of this research is to identify techniques to address the problems and shortcomings of sensor-based interaction. In order to achieve this aim, the following objectives were identified:

1. *To identify the problems and shortcomings of existing interaction techniques for mobile map-based applications (Chapter 2).*

   Existing interaction techniques need to be analysed in order to explain why sensor-based interaction is worthy of further investigation. This process will help to identify shortcomings of existing techniques which can be addressed by sensor-based interaction.

2. *To identify and analyse existing sensor-based interaction techniques for interacting with mobile map-based applications (Chapter 3).*

   Before any enhanced sensor-based interaction techniques can be proposed, it is necessary to analyse existing research. This will help to identify the state of the art in the design of sensor-based interaction techniques. Existing techniques will also be analysed to determine the shortcomings of existing sensor-based interaction techniques that need to be addressed.

3. *To compare sensor-based interaction to keypad interaction for mobile map-based applications (Chapter 4).*

   In order to gain deeper insight into the differences between sensor-based interaction and an established one-handed interaction technique, a comparative evaluation between keypad and sensor-based interaction will be conducted. This will allow opportunities for improvement of sensor-based interaction techniques for mobile map-based applications to be identified.

4. *To identify enhanced interaction techniques to address the usability shortcomings of sensor-based interaction (Chapters 5 and 7).*

   Once the shortcomings of sensor-based interaction have been identified, enhanced techniques will be proposed to address these shortcomings. The design of these techniques will be motivated and described.
5. To evaluate the proposed techniques against existing techniques (Chapters 5 and 7).
   In order to determine whether the proposed techniques offer any subjective or objective
   benefits over basic sensor-based interaction, comparative user studies will be conducted
   to compare the proposed techniques with a basic sensor-based interaction
   implementation. Participants in these user studies will be required to perform typical
   mobile map-based tasks.

6. To compare sensor-based interaction using sensor fusion with a single-sensor
   implementation (Chapter 6).
   The use of sensor fusion can potentially address many of the shortcomings of sensor-
   based interaction techniques. A sensor-based interaction technique using input from a
   single sensor will be compared to an implementation using sensor fusion by means of a
   comparative user study.

7. To propose a framework for the design of enhanced sensor-based interaction techniques
   (Chapter 8).
   The results of this research will be used to propose a framework for sensor-based
   interaction. This framework will demonstrate how the proposed techniques can be
   incorporated into the design of sensor-based interaction, both from a procedural and an
   application design perspective.

1.6 Scope
This research will be limited to the domain of mobile map-based applications. Map-based
applications require large information spaces to be browsed at multiple zoom levels. The
problems with existing interaction techniques are particularly severe in applications of this
nature. While this domain has much in common with other applications (e.g. web browsers,
photo galleries and document viewers), focusing on map-based applications will allow the
scope of this research to be constrained. Specific tasks which require interacting with a two-
dimensional map interface (which also supports multiple zoom levels), such as locating
points of interest, following a route and finding a route, will be used for evaluation purposes
(Büring et al. 2008). Interaction requiring menus, dialog navigation and text entry will not be
considered, as existing interaction techniques are well suited to such operations.

This research will focus specifically on one-handed interaction techniques. For this reason,
touch-screen interaction is not evaluated as an alternative to sensor-based interaction. Touch-
screen interaction will not be ignored, however, and is analysed along with existing
techniques to determine how the shortcomings of existing techniques can be addressed. Furthermore, the focus of this research is on improving existing sensor-based interaction techniques, rather than attempting to replace touch-screen interaction.

The sensors to be used in the development of sensor-based interaction techniques will be limited to those that are publicly available on mobile phones.

1.7 Research Philosophy and Approach

This research will largely follow a positivist research philosophy. Under this philosophy, certain assumptions are made about the phenomena being investigated and the research approach employed. These assumptions include (Oates 2006, Myers 2009, Saunders, Lewis and Thornhill 2009):

- **Independence**: The world being researched exists independently of the observer and can be studied and measured.
- **Objectivity**: The researcher is objective and independent of what is being observed.
- **Hypothesis testing**: Specific hypotheses will be formulated in order to test proposed theories.
- **Measurement**: The formulated hypotheses will be tested using data collected about specific variables.
- **Analysis**: The collected data will be analysed using statistical analysis to decide whether or not to reject the proposed hypotheses.
- **Generalisation**: Conclusions and inferences will be drawn based on the analysis of the observed sample data and generalisations made.

This philosophy is appropriate in order to evaluate the thesis statement proposed in Section 1.3. Modifications to sensor-based interaction will be proposed and evaluated to determine whether significant usability benefits are achieved. While positivist research largely focuses on the use of quantitative data, qualitative data will also be used to obtain user perceptions (Saunders et al. 2009). This data will be used to gain richer insight into participants’ experiences with the interaction techniques and to help interpret the quantitative data collected.

This research will largely employ a deductive research approach, as is common with a positivist research philosophy (Oates 2006, Saunders et al. 2009). Deductive research
involves formulating hypotheses based on some theory, identifying appropriate metrics to evaluate the hypotheses, measuring these metrics through experimentation, analysing the results and arriving at conclusions based on the results (Saunders et al. 2009). This approach will be employed in the various user studies to be performed in this research. Inductive reasoning will be employed in the analysis of qualitative comments in order to avoid limiting the interpretation of the data to pre-conceived ideas (Oates 2006).

1.8 Research Methodology

The following sections describe and motivate the research methods that will be used to answer the research questions identified in Section 1.4.

1.8.1 Literature Review

Literature reviews are important to place research in context and to form a theoretical basis from which to work (Hofstee 2006, Olivier 2009). Two literature review chapters are presented in this thesis. The first (Chapter 2), will examine the domain of mobile map-based applications. This literature review is important to identify the typical tasks supported by mobile map-based applications. This will allow representative tasks to be identified for use in experimentation. Furthermore, the low-level operations required to execute these tasks need to be identified. Existing interaction techniques which support these operations need to be analysed to identify problems and shortcomings and motivate the need for sensor-based interaction techniques.

The second literature review chapter (Chapter 3) will examine the state of the art in sensor-based interaction techniques. This literature review will be used to identify current sensor-based interaction techniques used to support the low-level operations required by map-based applications identified in Chapter 2. Existing techniques will be analysed in order to identify problems and shortcomings and opportunities for improvement.

1.8.2 Prototyping

Prototypes can serve as useful tools for experimentation (Lazar, Feng and Hochheiser 2010, Olivier 2009). Extensive use of prototyping will be made in this research. A prototype mobile map-based application supporting typical map-based tasks will be developed in order to evaluate the techniques proposed in this research. Versions of the prototype supporting keypad and sensor-based interaction will be developed for comparison purposes. The proposed enhanced sensor-based interaction techniques will be implemented using this
Chapter 1: Introduction

1.8.3 Usability Studies

Several comparative usability studies will be conducted in this research in order to evaluate the proposed research hypotheses regarding different interaction techniques (Lazar et al. 2010, Tullis and Albert 2008). Participants will be required to perform typical mobile map-based tasks using a prototype application. The usability studies will involve measurement of objective performance metrics as well as collection of subjective user ratings and comments. Usability studies will be conducted to compare sensor-based interaction to keypad interaction and to evaluate whether the proposed enhanced sensor-based interaction techniques provide any benefits. The experimental design of the individual user studies will be discussed and motivated in more detail in the relevant chapters.

1.9 Research Contribution

This research will make the following contributions to the study of sensor-based interaction techniques for mobile map-based applications:

- A prototype mobile map-based application incorporating sensor-based interaction;
- Identification of usability shortcomings of existing sensor-based interaction techniques;
- Proposal of enhanced sensor-based interaction techniques;
- Experimental results regarding the evaluation of the proposed techniques;
- Design recommendations regarding the use of sensor-based interaction; and
- A framework for the design of enhanced sensor-based interaction techniques.

A prototype mobile map-based application will be developed to evaluate different sensor-based interaction techniques. This prototype will also be used to perform a comparative user study between sensor-based interaction and keypad interaction. The result of this user study will be used to identify and categorise the usability shortcomings of existing sensor-based interaction techniques. These shortcomings will be used as the starting point for the proposal of enhancements to basic sensor-based interaction. These enhancements will seek to address or minimise the shortcomings of existing sensor-based interaction techniques.
A series of user studies will then be conducted to evaluate the proposed techniques. The results of these user studies will be used to support the proposal of design recommendations for the use of sensor-based interaction for mobile map-based applications.

These recommendations will be incorporated into a framework for the design of mobile sensor-based interaction techniques. This framework will encapsulate common design practice in the domain of sensor-based interaction. This framework will also be used to demonstrate how the individual techniques proposed in this research can be integrated into future sensor-based interaction techniques, both from an architectural and a procedural point of view. The framework will be implemented as an Application Programming Interface (API) which can be used in the implementation of future mobile applications incorporating sensor-based interaction.

1.10 Research Ethics

Ethics approval for all experimental work done as part of this research was obtained from the Faculty of Science Research, Technology and Innovation Committee at the Nelson Mandela Metropolitan University (NMMU) on behalf of the NMMU Research Ethics Committee: Human (Reference number H09-Sci-CSS-007).

1.11 Thesis Structure

This thesis consists of nine chapters designed to address the research questions outlined in Table 1.1. Figure 1.1 illustrates the relationship between the various chapters in this thesis.

Chapter 2 (Mobile Map-Based Applications) presents a literature review of mobile map-based applications. Several mobile map-based applications are analysed to identify the typical tasks supported by these applications and the associated low-level operations. Existing interaction techniques used to support these operations are then discussed. These techniques are analysed to identify problems and shortcomings of existing interaction techniques.

Chapter 3 (Sensor-Based Interaction Techniques) presents a detailed literature review of sensor-based interaction. The chapter begins with a discussion of sensor technology and the major stages involved in sensor-based interaction are described. Existing sensor-based interaction techniques are analysed in terms of the low-level operations required by map-based applications identified in Chapter 2. These sensor-based interaction techniques are then
analysed to identify benefits, problems and shortcomings. Finally, existing frameworks and API support for sensor-based interaction are analysed.

Chapter 4 (Designing a Basic Tilt Interaction Technique) describes the design process of a basic tilt interaction technique employing best practices identified in literature. This tilt interaction technique is then compared to keypad interaction using a prototype mobile map-based application, called MapExplorer. The experimental design of the user study is first presented, followed by the results. The results are then analysed to identify problems and shortcomings related to the use of tilt interaction in mobile map-based applications.
Chapter 5 (Proposed Enhanced Tilt Interaction Techniques) follows on from the previous chapter by proposing several techniques to address the shortcomings of tilt interaction identified in Chapter 4. Multimodal feedback, attractors to aid selection, gesture zooming, dwell-time selection and sensitivity adaptation are proposed. The design of each technique is described and motivated. The experimental design of a user study comparing an enhanced tilt interaction technique (incorporating the proposed techniques) with basic tilt interaction is then described. The results of this user study are presented and analysed, highlighting techniques which exhibited improvements and opportunities for further improvement.

Chapter 6 (Using Sensor Fusion to Improve Tilt Interaction) describes the use of sensor fusion to address the shortcomings of accelerometer-only implementations of tilt interaction. The implementation of a sensor-fusion algorithm combining accelerometer, compass and gyroscope data is presented. The experimental design of a user study comparing accelerometer-only tilt with a sensor fusion implementation is then presented. The results of the user study are presented and analysed to determine whether any benefits were achieved.

Chapter 7 (Improving the Controllability of Tilt Interaction) follows on from Chapters 5 and 6 by investigating ways to improve the controllability of tilt interaction incorporating sensor fusion. Chapter 7 focuses on improving the controllability of two aspects of tilt interaction, namely zooming operations and the use of tilt interaction while mobile. Rate controlled tilt zooming is proposed. Sensitivity adaptation, proposed in Chapter 5, is further refined by employing a different learning technique and a dynamic approach to dampening the sensitivity of tilt interaction. The results of a user study comparing rate controlled tilt zooming and gesture zooming, as well as analysing different approaches to sensitivity adaptation, are presented and analysed.

Chapter 8 (A Mobile Tilt Interaction Framework) presents a framework highlighting the different stages of the tilt interaction process. The individual stages are then examined in more detail, describing how the techniques proposed in Chapters 5 – 7 can be integrated into the tilt interaction process. An application architecture is also proposed, demonstrating how the proposed techniques can be integrated into the design of mobile map-based applications. The implementation of the framework as an Android API is also described.

Chapter 9 (Conclusions) concludes this thesis. A summary of the research findings is presented, highlighting the results of the user studies that were conducted. The contribution
made by this research is then outlined. Finally, problems encountered and opportunities for future research are identified.
Chapter 2: Mobile Map-based Applications

2.1 Introduction

This chapter presents a literature review of mobile map-based applications. The primary objective of this chapter is to identify the problems and shortcomings of existing interaction techniques used by mobile map-based applications. In order to identify the problems and shortcomings of existing techniques, the domain of mobile map-based applications will first be reviewed. Common tasks supported by mobile map-based applications will then be identified. This will allow the identification of representative tasks to be used in experimentation in this research. The typical mobile map-based tasks will then be analysed in order to identify the low-level operations that must be supported by any interaction technique. Existing interaction techniques will then be analysed in terms of the low-level operations identified in order to determine any problems and shortcomings that may exist.

2.2 Background

The use of maps on mobile phones is a rapidly expanding market. A recent study showed that in the United States alone, 48 million people accessed maps on their mobile phone in the three month period ending in May 2011, 39% more than during the same period the year before (Comscore 2011). All major mobile phone platforms now ship with standard mapping software. Maps are accessed on mobile phones either via the browser or via native applications. In some cases, native applications are merely wrappers for easy access to maps via the Internet. In the US, more users make use of dedicated map applications than browser-based maps (Comscore 2011). In April 2011, the Google Maps mobile app was identified as the most popular mobile app in the UK relying on Internet connectivity, with 6.4 million users (Guardian 2011).

Mobile map-based applications take advantage of the sensing capabilities provided by mobile phones to deliver information that is relevant to the user’s context (Setlur et al. 2010). Most mobile map-based applications now provide not only map views but also aerial photograph and satellite terrain views. Street-level views constructed from multiple photographs captured by custom-built vehicles are also becoming more common, blurring the lines between map software and augmented reality. Mobile map-based applications are also increasingly starting
to emerge on medium-range mobile phones as the costs of GPS receivers are reduced (Nokia 2011c).

2.3 Tasks

Mobile map-based applications have been developed to support a wide variety of tasks. Several multi-purpose mobile map-based applications exist and support users in a range of everyday tasks such as navigation and locating places and points of interest. Examples include Nokia Maps and Google Maps (Nokia 2011b, Google 2011). Other applications are restricted to a particular domain, such as the management of natural resources or mobile tourist guides (Burigat and Chittaro 2008, Baus, Chreverst and Kray 2005). This research focuses largely on multi-purpose mobile map-based applications.

Previous research has categorised the tasks commonly supported by mobile map-based applications (Von Hunolstein and Zipf 2003, Reichenbacher 2004). Analysis of previous categorisations yields the following set of common tasks:

- **Searching**: Identifying facilities matching certain criteria (e.g. where is the nearest petrol station?);
- **Navigating**: Finding a route between two points and navigating along the route;
- **Identifying**: Identifying and recognising persons or objects (e.g. who is denoted by the blue icon?);
- **Locating**: Identifying the position of something (e.g. where am I?); and
- **Checking**: Determining the condition of a particular location (e.g. operating hours of a business).

Two contemporary mobile map-based applications, namely Nokia Maps and Google Maps, were analysed in order to identify whether the above tasks are representative of current mobile map-based applications. These two applications were selected as both are available on multiple mobile platforms and are two of the most popular contemporary mobile map-based applications.

Figure 2.1 shows the main menus of Nokia Maps and Google Maps. All the above tasks are supported by both applications (locating, checking and identifying require interacting with the maps). In addition to the above tasks an additional task category can also be identified, namely:
• **Interacting:** Interacting tasks allow users to interact with POIs. Opportunities for interaction include sharing the POI’s location via social networking tools, saving the POI’s location, submitting a review of the POI and, if appropriate, communicating via the provided email address or phone number.

The above tasks can be combined to perform a particular activity. For example, going to a restaurant can combine searching for a restaurant, sharing the restaurant location, checking opening hours and navigating to the restaurant.

The following section discusses the above tasks in more detail. Nokia Maps and Google Maps are used to provide illustrative examples of the tasks in contemporary mobile map-based applications.

### 2.3.1 Searching

Searching tasks require the user to find people, places or objects matching certain criteria. Mobile map-based applications often provide a dialog interface to facilitate searching using different criteria. Searches may be distance related (for example, finding restaurants within 5km). Category searches are also common (for example, users may want to find all Italian restaurants). Different search parameters can also be combined (for example, to search for all hotels within 5km with at least a four star rating). In addition, searching by address is often facilitated to allow the user to easily search for a particular street address.
Chapter 2: Mobile Map-based Applications

Figure 2.2 shows the search interfaces provided by Nokia Maps and Google Maps. Both applications allow the user to search either by selecting a category or by entering a place name or address in the search box. Selecting a particular category retrieves a list of places or POIs matching the selected category near the user’s current location. Once users have identified a particular place or POI, they can then select to view the relevant location on the map. Existing keypad and touch-screen interaction techniques are well suited to searching in this manner as searching tasks largely rely on menu navigation and text entry.

2.3.2 Navigating

Navigating tasks consist of two separate sub-tasks. The first sub-task requires the user to plan a route. Users must select the starting point (which may be their current location) and the destination point. The second sub-task is following the route to navigate from the starting point to the destination. Figure 2.3 shows these two stages being performed in Google Maps. The screenshot on the left shows the interface for selecting the start and end points of the route (can be a search term, a location on the map or a saved location). The desired mode of transport can also be selected. The screenshot on the right shows the calculated route with navigation instructions being displayed at the appropriate points along the route. Devices which incorporate location sensors (such as GPS) often provide an option to automatically update the display to follow the user’s real-world progress along the route.
Identifying tasks involve identifying and recognizing people, places, or objects in a mobile map-based application (Reichenbacher 2004). Most mobile map-based applications offer some sort of detail-on-demand functionality. While users are browsing the map, they are able to select items to view related information about the relevant item. Figure 2.4 illustrates identifying tasks in Nokia Maps and Google Maps. Identification information is presented using a tool-tip style technique (similar to a mouse-over effect on desktop computers) to aid in identifying icons or locations on the map. The tool-tip is activated by moving the cursor over the POI icon (Nokia Maps) or touching the icon on the screen (Google Maps). Selecting the POI icon displays more detailed information.
2.3.4 Locating

A common task in mobile map-based applications is to locate a particular person, place or object (Reichenbacher 2004). Users may need to establish their location in order to orientate themselves. Locating tasks can also be a precursor to navigating tasks, as users search for the start and/or end point of a route to be navigated. Locating tasks can be amongst the most difficult mobile map-based tasks to perform due to the lack of an overview function in most mobile map-based applications. In cases where an overview is available, it is often too small to distinguish detail or obscures part of the display (Chittaro 2006). Locating tasks can be conducted by browsing the map and performing several identifying tasks to try and find the location in question. Alternatively, a search can first be executed and the desired location found by examining the returned results on the map (Figure 2.4).

2.3.5 Checking

Checking tasks involve determining the status of real-world people, places or objects (Reichenbacher 2004). This could include viewing the opening hours of a business or viewing traffic information for a particular road (Von Hunolstein and Zipf 2003). Many checking tasks therefore first require the user to locate the person, place or object in question and then perform a selection operation to view relevant details. In some cases (such as traffic volume visualisation), selection may not be necessary and the user may only be required to look at the map and interpret the visualisation. Checking tasks often involve time dependent information (Von Hunolstein and Zipf 2003).

Figure 2.5. Checking POI details in Google Maps
In Figure 2.5 a checking task performed using Google Maps is shown. In the screenshot on the left, the user is shown moving the cursor over an icon showing the Boardwalk Casino. After selecting the item, the user is able to view related information, including contact details for the casino. Checking tasks that involve selecting POIs in this manner therefore first require a searching or locating task to be performed to find the icon in question.

2.3.6 Interacting

Contemporary mobile map-based applications typically provide a range of options for interacting with POIs and other map content. Integration with social networking tools allows users to share the location of a POI or to “check in” to indicate their presence at the POI in question. Users are also often able to save the location of the POI, submit a review of the POI and to establish contact via email or phone. Nokia Maps and Google Maps both provide several options for interacting with POIs. Figure 2.5 (right side) and Figure 2.6 show the Google Maps and Nokia Maps interfaces for interacting with POIs.

![Figure 2.6. Interface for interacting with POIs in Nokia Maps](image)

2.4 Low-Level Operations

The six high-level tasks discussed above are accomplished through several low-level operations. Three low-level interaction operations are common for map-based tasks, namely panning, zooming and selection (Setlur et al. 2010, Looije et al. 2007). Text entry and interaction with user interface (UI) controls are required for tasks which are not purely map-oriented.
Table 2.1 provides a breakdown of the low-level operations required to complete the high-level tasks identified in the previous section. Given the fact that checking and identifying tasks usually follow on from locating or searching tasks, the summary in Table 2.1 shows that almost all of the high-level tasks discussed in Section 2.3 require panning, zooming and selection. It is important, therefore, that mobile map-based applications allow these three operations to be performed quickly and easily. The focus of this research is on interacting with map-based interfaces. Existing interaction techniques are well-suited to text entry and interacting with UI controls. The remainder of this research will focus on tasks requiring panning, zooming and selection. Locating, navigating, checking and identifying tasks will therefore be used in experimentation.

<table>
<thead>
<tr>
<th>Panning</th>
<th>Zooming</th>
<th>Selection</th>
<th>Text Entry</th>
<th>UI Controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locating</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Searching</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Navigating</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Checking</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Identifying</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Interacting</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 2.1. Breakdown of the low-level operations required by each task type

Existing interaction techniques are now analysed in more detail to determine how panning, zooming and selection are currently supported. This analysis will enable the identification of problems and shortcomings of existing techniques.

### 2.5 Existing Interaction Techniques

Keypad and touch-screen interaction are commonly used to support interaction with mobile map-based applications. This section will analyse how keypad and touch-screen interaction techniques have been used to support users in performing panning, zooming and selection operations. Sensor-based interaction techniques are analysed in the following chapter.

#### 2.5.1 Panning

Maps displayed on mobile devices are usually significantly larger than the screen space available on the mobile device. This presents significant usability challenges for mobile map-based applications and is sometimes referred to as the presentation problem (Spence 2007).
Panning is usually used to overcome this. The mobile display can be thought of as a window, through which the user can view a portion of the map at a time. Panning is performed to shift the viewing window across the map (Burigat, Chittaro and Gabrielli 2008). Panning can also conceptually be thought of as shifting the map underneath the viewing window. Panning is sometimes also referred to as scrolling, particularly when the interface provides scrollbar controls to shift the visible area of the map.

Various input modalities have been used to perform panning. Devices that do not provide touch-screen capabilities rely on directional keys, the numeric keypad, a trackpad or a small joystick built into the keypad to perform panning (Looije et al. 2007). Freedom of movement is typically limited to up, down, left and right panning, although in some cases diagonal panning is also possible. Touch-screen devices support a wider range of panning techniques.

Several different forms of panning interaction have been implemented on touch-screen devices. These include (Looije et al. 2007, Burigat et al. 2008):

- **Dragging:** Dragging the stylus across the screen in order to pan the map in the corresponding direction.
- **Scrollbars:** Scrollbar user interface controls allow the user to scroll the map horizontally and vertically (Figure 2.7).
- **Tapping to centre:** Tapping the screen to centre the map at the tapped location.
- **Tapping to pan:** Tapping at the edge of the screen to move in the corresponding direction.
- **Radial scroll:** The stylus must be moved in a clockwise or anti-clockwise direction across the screen in order to pan forwards or backwards. Used for applications in which only vertical panning is possible.
- **Touch-n-go:** The screen is touched with the stylus. The direction of panning is determined by the position of the stylus relative to the centre of the screen and the speed of panning by the distance between the selected point and the centre of the screen.
Chapter 2: Mobile Map-based Applications

Figure 2.7. On-screen controls for zooming and scrollbars for panning (Jones et al. 2005)

Many modern touch-screen phones are not equipped with a stylus. Instead, they allow users to interact with the phone using their fingers. Panning using this form of interaction typically relies on two gestures, namely dragging and flicking. These two gestures differ in terms of the level of control the user has over the panning motion. Dragging gestures involve touching the screen at a particular point and then dragging and releasing at a second point (Figure 2.8). The user is able to precisely control the speed and direction of panning. Flicking is a less precise panning gesture. The user places his/her finger on the screen and performs a rapid flicking gesture (Figure 2.9). The display will then continue to pan after the user’s finger is released, relying on momentum, with the initial panning speed and direction relative to the speed and direction of the flicking gesture.

Figure 2.8. Dragging panning gesture (Microsoft 2011a)

Figure 2.9. Flicking panning gesture (Microsoft 2011a)
2.5.2 Zooming

Most mobile map-based applications provide some facility for the user to adjust the zoom level or scale at which the map is viewed. Some applications (such as Google Maps) employ a fixed number of discrete zoom levels. Animated transitions between zoom levels are usually employed to create the illusion of continuity. Vector-based mobile map applications (such as Nokia Maps) allow continuous zoom level adjustments to be made. Non-sequential interaction, where users are able to skip to a specific zoom level is also possible (Burigat et al. 2008).

Non-touch screen mobile devices generally rely on menu options combined with keypad shortcuts (the use of the * and # keys is common) for zooming in and out. Numbers on the keypad (1-9) have also been mapped onto rectangular regions of the display (Robbins et al. 2004). In this technique, called ZoneZoom, the user is able to make a portion of the map fill the whole screen by pressing the corresponding number key.

Touch-screen devices provide a wider range of zooming implementations. These include:

- **Zoom-sliders**: The user is able to adjust the zoom level by manipulating a slider control (Burigat et al. 2008).
- **On-screen buttons**: Dedicated buttons allow the user to zoom in and out (Figures 2.4 and 2.7).
- **Dragging**: The user is able to zoom in or out by dragging upwards or downwards across the screen (Looije et al. 2007).
- **Rectangle selection**: The user selects a rectangle using the stylus and the zoom level is adjusted so that the area within the selected rectangle expands to fill the available display space (Looije et al. 2007).
- **Gestures**: Multi-touch devices allow the user to adjust the zoom level using touch-screen gestures. Zooming in can be performed by placing two fingers together on the screen and dragging them apart. Zooming out can be performed by placing two fingers apart on the screen and drawing them together (Figure 2.10) (Apple Inc. 2011c).
The limited screen size available on mobile phones means that the user is only able to view a small portion of the map at a time (Setlur et al. 2010, Looije et al. 2007). This can make it very difficult for users to orientate themselves in the global context. Traditional zooming techniques are not sufficient to address this shortcoming as overview (zoomed out) and detail (zoomed in) views of the map cannot be viewed simultaneously (Burigat et al. 2008). Several approaches have been developed to combat this problem, including Overview + Detail and Focus + Context.

Previous research has highlighted the importance of providing an overview in order to improve the user’s awareness of the content, structure or changes in an information space (Hornbæk and Hertzum 2011). Overviews have been shown to reduce task completion times and improve user satisfaction. Overview + Detail techniques show both an overview and a detailed view of the map simultaneously. Figure 2.11 shows an implementation of Overview + Detail previously successfully employed in a mobile map-based application, where an overview is overlaid on the map and displayed in the bottom right corner of the display (Burigat and Chittaro 2011). This technique proved effective for tasks requiring users to locate off-screen icons. This technique has the disadvantage that part of the detail view is obscured by the overview.

Focus + Context techniques attempt to integrate detail and overview in a single display (Cockburn, Karlson and Bederson 2008). Focus + Context techniques frequently employ distortion techniques such as fisheye lenses to visualise information (Gutwin and Fedak 2004). Fisheyes usually magnify a portion of the display while compressing or distorting information on the periphery. The user is therefore able to view detail while still being provided with a sense of how that detail fits into the broader global context. This approach is often considered unsuitable for visualising map-based information, because the distortion makes spatial reasoning difficult (Cockburn et al. 2008).
2.5.3 Combined Zooming and Panning

Existing interaction techniques for performing zooming and panning often require users to switch their attention from the map-based display to the visual controls for performing either zooming or panning (Igarashi and Hinckley 2000). Blurring when panning at high speeds can also result in user disorientation (Cockburn and Savage 2003). In order to simplify user interaction and allow users to concentrate on the display, a technique has been proposed which combines zooming and panning into a single operation. Speed Dependent Automatic Zooming (SDAZ) was proposed for desktop browsing of large information spaces in an attempt to address some of the problems and shortcomings of traditional panning techniques (Igarashi and Hinckley 2000). SDAZ adjusts the zoom level automatically in response to the speed at which the user pans the display in order to keep the visual flow across the screen at a constant speed. The faster the user pans the display, the more zoomed out the view. This technique allows zooming and panning to be performed with a single operation, eliminating the need for users to switch their attention between the display and the user interface controls. The blurring effect is also addressed by zooming out when panning at high speeds. Desktop implementations of SDAZ showed significant efficiency improvements for map-based tasks (Cockburn and Savage 2003).

SDAZ was initially developed for use in desktop visualisation systems. More recently, the technique has been adapted for use in mobile visualisation systems (Jones et al. 2005). Mobile SDAZ has been applied to both document and map navigation. The technique was shown to improve the accuracy of target location in a mobile map-based application, although at the cost of increased task completion times.
The need for cues indicating the position of the current view within the global context has previously been identified as being important in order to maintain a sense of orientation and prevent frustration when using map-based interfaces (Harrower and Sheesley 2005). One of the motivations for the use of SDAZ is that it helps users better maintain a mental picture of how the current view fits into the broader global context.

2.5.4 Selection

Selection is a common operation in mobile map-based applications. Selection operations allow users to select the start and end points for navigation tasks and to select points of interest in order to view details on demand. The use of a cursor to facilitate selection is common in non-touch screen devices (Setlur et al. 2010). This is usually fixed in the centre of the screen. The user simply moves the cursor over the target and presses a selection key in order to trigger a selection event. In some cases, it may not be necessary for the user to press any key to retrieve information of interest. Figure 2.4 shows an example in Nokia Maps, where the user is able to view information regarding a POI merely by moving the cursor over the POI in question. For touch-screen devices, selection simply requires users to tap the POI icon in order to perform a selection (Setlur et al. 2010).

2.6 Problems and Shortcomings

Traditional input techniques used by mobile map-based applications suffer from a range of problems and shortcomings. Many of these problems stem from the fact that interfaces for mobile map-based applications are often merely scaled-down versions of desktop user interfaces (Oakley et al. 2004, Oakley and O’Modhrain 2005). Interaction techniques developed for the desktop environment are not well suited to the mobile environment where screen real-estate and input mechanisms are limited (Hughes et al. 2004). The keypad and stylus, still widely used for input in mobile phones, are really just scaled down, portable versions of the keypad and mouse used in desktop computers. These forms of interaction may work well when the user is stationary, but are less effective when the user is mobile, distracted, or busy with another task (Dong et al. 2005). The techniques identified in the previous section for panning, zooming and selection are now analysed to identify problems and shortcomings.
2.6.1 Panning

Input mechanisms such as the keypad and stylus may be well suited to traditional phone functions, such as dialling and sending of text messages, but are less well suited to browsing large information spaces, such as maps on mobile phones (Hansen et al. 2005). These input mechanisms do not represent an intuitive form of control for applications of this type (Winkler et al. 2007). Keypad interaction suffers from inherent shortcomings in terms of the level of control the user has over panning speed and direction. The user’s freedom of movement for panning operations is usually restricted to four (or occasionally nine) predefined panning directions. Keypad interaction is also a binary form of interaction. Although some form of acceleration is occasionally built in, the user is usually restricted to a default panning speed. The default speed is often viewed as being too fast by beginners and too slow by experts (Kratz and Rohs 2008). Keypad interaction has therefore been described as a tedious form of interaction for panning long distances (Cho et al. 2007b).

Many of the touch-screen panning techniques identified in Section 2.5.1 require two-handed interaction (dragging, tapping to centre, tapping to pan and touch-n-go). Two-handed interaction is not desirable, given the context of use of mobile map-based applications (Karlson et al. 2006, Pascoe et al. 2000, Rekimoto 1996). Gestures such as flicking, where momentum is created by the speed of the flick gesture, allow for rapid one-handed thumb input for simple tasks such as list scrolling. In mobile map-based applications, however, such gestures may have to be repeatedly performed and can become physically demanding (Miyaki and Rekimoto 2009). Furthermore, flicking gestures do not allow for precise control over panning speed and direction (Lee, Lee and Chung 2011).

Touch-screen input, with and without a stylus, can result in a portion of the screen being occluded while the user is interacting with the display (Rekimoto 1996, Fitchett and Cockburn 2009). This problem is particularly severe for techniques which do not employ a stylus (Figure 2.12).

The use of touch-screen controls such as scrollbars for panning wastes valuable screen space (Fitchett and Cockburn 2009). Users are also required to switch their attention between the user interface controls and the map, which should be the primary focus of their attention (Eslambolchilar and Murray-Smith 2004). Touch-screen interaction can also be error prone in hot or cold conditions, where the user’s fingers may be sweaty or the user may be wearing gloves (Fitchett and Cockburn 2009).
2.6.2 Zooming

Keypad and touch-screen interaction techniques both frequently make use of separate controls for performing zooming, including hardware and touch-screen buttons and zoom sliders. This approach requires the user to switch between different controls for zooming and panning (Mooser, You and Neumann 2008). Furthermore, on-screen controls are often small and grouped closely together. As a result, they can be difficult to select, increasing the frequency of errors (Crossan et al. 2008). The user’s full attention is often required. Users will often divide their attention between the mobile device and their environment, making such attention-intensive interaction techniques undesirable (Dong et al. 2005).

Touch-screen gestures such as drawing rectangles and multi-touch zooming gestures (Figure 2.10) eliminate the need for on-screen controls for zooming. These techniques, however, require two-handed interaction, which is not always feasible in a mobile context of use.

Combined panning and zooming techniques such as SDAZ eliminate the need for switching between panning and zooming controls. SDAZ, however, does not allow for zooming independently of panning and has previously been criticised for taking control away from users (Büring et al. 2008).

2.6.3 Selection

Keypad and touch-screen interaction both provide established methods for performing selection operations (physical selection buttons and tapping the screen). Touch-screen selection without a stylus has often been described as inaccurate because of the imprecise
nature of fingertip and thumb input (Park and Han 2010). This problem occurs because the selection target is occluded by the user’s finger and because of the large area of contact between the fingertip and the screen (Vogel and Baudisch 2007). The user is also usually unable to reach the entire screen surface using one-handed input. User interface controls have to be made larger to compensate for the inaccuracies of touch selection (Karlson and Bederson 2007). This is problematic in mobile applications where screen space is limited. One-handed touch-screen selection accuracy problems are even more severe when the user is mobile (Schildbach and Rukzio 2010). Various techniques have been developed in attempts to address the inaccuracies of touch-screen selection and facilitate one-handed input (Roudaut, Huot and Lecolinet 2008, Vogel and Baudisch 2007, Holz and Baudisch 2010, Karlson and Bederson 2007). Many of these techniques have achieved positive results in short term user studies, but have yet to achieve widespread adoption.

2.7 Conclusions

Mobile map-based applications support users in performing six common tasks, namely locating, searching, navigating, checking, identifying and interacting. Locating, navigating, checking and identifying tasks primarily involve interacting with the mobile map-based display and will therefore be the focus of this research. These tasks can be broken down into three low-level operations, namely zooming, panning and selection.

Panning tasks can easily be performed using keypad input, but with little control over panning speed and limited freedom of movement. Touch-screen panning techniques allow for both precision and control, but require both hands. Physical buttons, on-screen controls and touch screen gestures have all been employed to perform zooming. Several techniques also exist which attempt to minimise the need for zooming by providing simultaneous overview and detail representations of the map. SDAZ was identified as a technique which links the current zoom level to the current panning speed, eliminating the need for separate panning and zooming operations. Selection operations can be performed through the use of physical buttons, mouse-over type effects (using a cursor) and touch-screen selection.

Existing interaction techniques, however, suffer from a range of problems and shortcomings. Keypad interaction is often very restrictive, with pre-defined panning speeds which can be frustrating for advanced users. Users are usually also limited to panning in four or nine pre-defined directions. Touch-screen interaction requires the user’s full attention and results in the display being partly obscured. Touch-screen selection without a stylus is inaccurate.
Existing techniques also frequently require the user to switch between different controls and modalities. All of these problems are undesirable in a mobile context of use. Instead of merely taking interaction mechanisms designed for the desktop and transferring these to mobile environments, new interaction techniques are required which consider the unique requirements and opportunities offered by map-based applications on mobile devices.

The following chapter discusses sensor-based interaction techniques for mobile map-based applications and examines the benefits which such techniques can provide for addressing the problems and shortcomings of the techniques discussed in this chapter.
Chapter 3: Sensor-Based Interaction Techniques

3.1 Introduction

This chapter presents a literature review of sensor-based interaction techniques. The objective of this chapter is to analyse existing sensor-based interaction techniques to identify benefits, problems and shortcomings. This analysis is intended to identify current best practices in the design of sensor-based interaction techniques and to identify opportunities for improvement in the domain of mobile map-based applications. Existing Application Programming Interfaces (APIs) are analysed in order to identify the level of support provided by mobile platforms for the development of sensor-based interaction techniques.

This chapter begins with an overview of sensor technology. The main stages involved in processing sensor data for sensor-based interaction are then identified. Existing sensor-based techniques are categorised in terms of the low-level operations required by map-based applications, namely panning, zooming and selection. Existing techniques are then analysed to identify the advantages and disadvantages of current sensor-based interaction techniques. Existing frameworks for designing sensor-based interaction techniques are also discussed. Finally, existing APIs supporting the acquisition and interpretation of sensor data are briefly analysed.

3.2 Sensors

A sensor can be defined as a (Van Laerhoven 2005):

"Device capable of detecting and responding to physical stimuli such as movement, heat or light."

The Java Micro Edition Sensor API specification describes sensors as "any measurement data source" (Nokia 2011a). The specification makes a distinction between physical sensors and virtual sensors (Goyal 2010). Physical sensors correspond to the above definition. Virtual sensors are described as sensors which combine and manipulate data from physical sensors (e.g. a battery level sensor display).

Mobile devices are increasingly being equipped with a wide array of microelectromechanical systems (MEMS) sensors, as manufacturing costs drop and the hardware capabilities of mobile devices improve. In 2009, it was estimated that 28% of mobile phones were equipped
Chapter 3: Sensor-Based Interaction Techniques

with an accelerometer sensor (iSuppli 2010). It is anticipated that by 2014, 65% of all phones will be equipped with an accelerometer. The number of phones equipped with digital compasses and gyroscopes is also rapidly increasing. This trend is likely to continue with platforms such as Windows Phone listing accelerometers as part of the required minimum hardware specifications. Other sensor types that are common include location sensors (including GPS receivers), proximity sensors (used to detect whether users are holding the phone to their ear) and ambient light sensors (used to adjust screen brightness according to ambient light conditions). Many of these sensors, including proximity and ambient light sensors, are not suitable for direct use as part of interaction techniques, but are more suitable for detecting information about the user’s current context.

Sensors can be either discrete or continuous (Rogers and Muller 2006). Discrete sensors provide binary input signals. Continuous sensors, in contrast, provide an ongoing stream of data. Most sensors that sample physical phenomena, such as accelerometers, are continuous sensors (Hartmann et al. 2007). User activities are also discrete or continuous. For example, selecting a POI icon on a map is a discrete action, while browsing a map looking for POIs matching certain criteria is a continuous action. It has been suggested that, for most activities, it is desirable to match discrete sensors onto discrete user activities (Rogers and Muller 2006). Many mobile map-based applications, however, use discrete interaction mechanisms for performing continuous activities. Map browsing, a continuous activity, is frequently supported through the use of discrete button inputs.

The following section examines the stages involved in using raw sensor data to perform panning, zooming and selection operations.

3.3 Processing Sensor Input

Previous research has highlighted various stages that are required to convert raw sensor data into meaningful interaction (Korpipää et al. 2003, Hartmann et al. 2007). Although the exact terms used may vary, the following five stages can typically be identified (Figure 3.1):

- **Sensor Measurement**: Accessing raw sensor data in a continuous fashion.
- **Pre-processing**: Smoothing or signal conditioning is generally required to filter out sudden variation and unwanted disturbance.
- **Feature Extraction**: Raw sensor data needs to be converted to some useful format, such as orientation angles or identified gestures.
• **Mapping to Operations**: The features extracted in the previous stage are mapped onto low-level operations such as panning, zooming and selection.

• **Performing Operations**: The appropriate low-level operations are performed.

![Stages in processing sensor data for sensor-based interaction](image)

**Figure 3.1. Stages in processing sensor data for sensor-based interaction**

Each of the above stages is now discussed in more detail.

### 3.3.1 Sensor Measurement

Most mobile sensors are continuous. This is in contrast to more traditional input techniques provided on both desktop computers and mobile phones, where interaction is generally modelled as a series of discrete events such as button presses or mouse clicks (Dix *et al.* 2004). One of the most common approaches to modelling continuous input is to simply regard continuous data as a succession of discrete events. For example, mouse movement can be broken up into a series of mouse-moved events (Dix *et al.* 2004). Existing Sensor APIs follow a similar approach, providing listener interfaces, whereby an application can register to “listen” for updates in sensor values. The programmer usually also has the option to select a customised sampling rate (Android 2011). One of the difficulties of implementing sensor-based interaction techniques on multiple platforms is that different devices and operating systems use different units of measurement for sensor values. Furthermore, as sensor technology is improved, the range and sensitivity of sensor input also varies from phone to phone.

### 3.3.2 Pre-Processing

Sensor-based interaction techniques frequently employ some form of signal conditioning or smoothing (Hartmann *et al.* 2007). This is necessary in order to ensure that the data in use represents intentional interaction, rather than signal noise or disturbance. Accelerometer data is particularly susceptible to disturbance from hand tremor and other linear acceleration not related to deliberate input. This can negatively affect user control (Cho *et al.* 2007a). For this reason, smoothing algorithms are generally employed to remove excessive variation from the raw sensor data. The Savitsky-Golay smoothing algorithm is an example of a smoothing
Algorithm that has previously been successfully used to filter raw, noisy accelerometer data (Mock and Rohs 2008). Figure 3.2 illustrates the contrast between the raw accelerometer data (shown in grey) and the smoothed data (shown in red). The official iOS Developer Library recommends the use of low-pass filtering to reduce the influence of sudden changes in accelerometer data (Apple Inc. 2011a). This approach has also previously been used in research into sensor-based interaction (Crossan et al. 2008). Most smoothing algorithms use a weighted average of several successive values, where the weightings and number of values used vary depending on the severity of the smoothing being performed.

![Figure 3.2. Smoothing accelerometer data with a Savitsky-Golay filter (Mock and Rohs 2008)](image)

**3.3.3 Feature Extraction**

In order to perform interaction, sensor data needs to be converted to a format which is useful for the purposes of interaction. This may involve processing accelerometer data to identify tilt angles (Cho et al. 2007b). Alternatively, sensor data may be used to identify discrete physical gestures (Jiayang et al. 2009).

In many cases, sensors are not used in isolation, but rather input from several sensors is combined to identify and process interaction or a change in context (Hughes et al. 2004, Schmidt, Beigl and Gellersen 1999). Various methods have been used to perform so-called sensor fusion. Aggregators have been proposed to combine input from multiple sensors in order to detect context changes. Sensors can run independently of any application. Aggregators are responsible for listening to the appropriate sensors and allowing the application to deal with the combined sensor data at a more abstract level (Dey, Abowd and Salber 2001).
3.3.4 Mapping onto Operations

Sensor-based interaction involves mapping processed sensor data onto low-level operations such as panning, zooming and selecting. The mapping stage in Figure 3.1 consists of converting the features extracted in the previous stage into a format suitable for performing panning, zooming or selection. For example, tilt angles could be mapped onto panning speeds, or shake gestures onto selection operations.

3.3.5 Performing Operations

The final stage involves simply performing the appropriate panning, zooming and selection operations. Existing research has employed a wide variety of sensor-based interaction techniques. The next section classifies existing sensor-based interaction techniques into panning, zooming and selection techniques.

3.4 Existing Techniques

Mobile sensors have been used and combined in several ways to support the low-level tasks identified in Section 2.4. In the following sections, existing techniques are categorised according to the low-level tasks supported. The discussion will not be entirely limited to mobile map-based applications, as many interesting applications of sensor-based interaction have been applied in similar domains and the potential for transfer to the domain of mobile map-based applications exists.

3.4.1 Panning

Panning operations are an essential part of mobile map-based applications. Previous research has employed a variety of sensor-based interaction techniques to perform panning. These techniques include tilt gestures, physical movement of the mobile phone (detected using the phone’s camera), pressure sensing and automatic panning of the display using location sensors.

The idea of using tilt gestures as a form of input was first suggested in the mid-1990s as a way for users to interact with menus, navigate two dimensional environments and interact with three dimensional object visualisations (Rekimoto 1996). Existing research has mainly made use of accelerometers to detect tilt gestures which are used to perform panning operations. Tilt interaction has been successfully applied in a range of domains including menu navigation, text entry, mobile museum guides, photo browsing and, most relevant to this research, interacting with mobile map-based applications (Tian et al. 2008, Mantyjarvi et
Chapter 3: Sensor-Based Interaction Techniques

al. 2006, Cho et al. 2007b, Kratz and Rohs 2008, Wigdor and Balakrishnan 2003). Tilt gestures around the x and y-axes are generally used to perform panning around a two-dimensional space.

The exact implementation of panning using tilt interaction varies between applications. As illustrated in Figure 3.3, horizontal panning is usually performed by tilting the phone left and right and vertical panning by tilting the phone up and down (Hinckley et al. 2005). Some implementations identify discrete inputs, where tilt interaction past a certain threshold is interpreted similarly to a button press (e.g. a left tilt is the same as pressing the left arrow) (Harrison et al. 1998, Mantyjarvi et al. 2006). Tilt interaction is more commonly used to perform panning in a continuous manner (Dachselt and Buchholz 2008). The mapping between tilt interaction and panning effect is usually performed using either position control or rate control (Rahman et al. 2009, Oakley et al. 2004).

Figure 3.3. Tilt gestures for two-dimensional panning. From left to right: down, neutral, up, left and right (Dachselt and Buchholz 2008)

Position control uses a direct mapping between tilt angles and the cursor position on the screen (Figure 3.4). In order to ensure that interaction is comfortable and that the display remains visible, a subset of the full range of motion needs to be employed (Partridge et al. 2002). Position control has been shown to be effective for short-range panning tasks such as menu navigation (Oakley and O’Modhrain 2005, Rahman et al. 2009). Position control has recently been used to perform panning in image gallery and browser applications on Android phones such as the LG Optimus Black (LG 2011). Position control is less well suited to mobile map-based applications, where long-range panning tasks may be required.
Chapter 3: Sensor-Based Interaction Techniques

Rate control uses the current tilt angle (relative to some neutral orientation) to control the current panning speed. For example, the further the device is tilted to the left, the faster the display pans to the left (Hinckley et al. 2005). A neutral zone is usually employed whereby small orientation changes are ignored to allow the user to maintain a steady cursor position (Dachselt and Buchholz 2009). The function mapping tilt interaction onto panning speed may not necessarily be linear. Previous research has made use of a linear mapping for small tilt angles, with the effect of larger tilt angles amplified (Figure 3.5). This approach is intended to allow for finer control with small tilt angles and faster panning with large tilt angles (Dachselt and Buchholz 2008, Mock and Rohs 2008). Rate control is better suited to mobile map-based applications, where long-range panning tasks make position control unsuitable (Oakley and O’Modhrain 2005). Rate control implementations can be more difficult to control, however, and often suffer from overshooting problems (Yim et al. 2011, Cho et al. 2007b).

Figure 3.4. Position control for menu navigation (Oakley and O’Modhrain 2005)

Figure 3.5. Mapping between tilt angles and panning speeds (Dachselt and Buchholz 2009)
Discretisation can also be used to improve the level of control users have over tilt interaction. In this approach, the range of possible tilt angles is split into intervals. Discretisation has largely been employed in conjunction with position control, with angles falling into the same interval mapped onto the same discrete tilt level. This is intended to minimise jitter and allow for finer-grained control. This approach has often been employed to improve control in tilt-controlled menu systems (Ni et al. 2011, Rahman et al. 2009, Oakley and Park 2009). In rate control systems, tilt angles falling into the same interval are mapped onto the same panning speed. This has the effect of turning the mapping between tilt angles and panning speeds into a step function. Linear, quadratic and sigmoidal discretisation are all possible (Figure 3.6), with existing research providing some evidence that quadratic discretisation allows for the highest level of efficiency and control (Rahman et al. 2009).

Most mobile phones are now equipped with built-in cameras. These cameras have frequently been used to detect movement of mobile devices. This movement has been used to allow mobile phones to act as a moveable porthole onto a large display area. Movement of the phone up, down, left and right is used to pan the display in the corresponding direction (Mooser et al. 2008). Other variants of camera-based input have made use of markers which are used to detect movement and perform panning in different directions (Winkler et al. 2007). Camera-based interaction techniques attempt to identify movement of the mobile device by comparing successive images captured using the camera. Differences between the images are used to detect movement which is then translated into panning operations (Yim et al. 2011).

The success of camera input depends on the composition of the images captured. In some cases, the nature of the background makes it very difficult to accurately detect movement. To compensate, camera-based interaction techniques often use some sort of reference markers to make it easier to distinguish movement. Figure 3.7 shows two examples. The dynamic peephole technique (left) uses a grid of visual markers to make it easier to detect movement. The magic lens technique (right) requires the user to move the device over a static physical
map. An array of black dots printed on the map is used as reference points for detecting movement of the phone (Rohs et al. 2007). The secondary camera on the front of 3G mobile phones has also been used to perform face detection, using the user’s face as a frame of reference to detect device movement (Hansen et al. 2005).

![Figure 3.7. Camera-based interaction using a grid (left) and static map (right) (Rohs et al. 2007)](image)

Hybrid approaches, using both accelerometer and camera input, have also been employed to support panning. Yim et al. (2011) proposed the use of a combination of tilt gestures and physical movement of the mobile device to navigate a photo collection. Tilt gestures were detected using an accelerometer and were used to navigate between views using rate control. Physical movement of the mobile device was detected using the device’s camera and an optical flow algorithm was used to pan through the current photo collection using position control.

Pressure sensing has been used in conjunction with touch-screen interaction to perform panning (Miyaki and Rekimoto 2009). The GraspZoom technique employed pressure sensors on the back of a phone to detect pressure input which is used to perform continuous panning. The direction of the input can be changed using a small touch-screen gesture prior to initiating pressure input. This technique is, however, limited to one dimensional scrolling and is therefore not suitable for interaction with map-based applications. Rate control has been shown to be superior to position control for scrolling through menu options using pressure input (Wilson et al. 2011).

Location sensors such as GPS sensors have also been used to perform automated panning in mobile map-based applications. These sensors are useful for aiding users in identifying their current position on a map, or helping them keep track of their position when performing navigation tasks (Looije et al. 2007).
3.4.2 Zooming

Tilt gestures have also been used to perform zooming operations in a variety of domains, including mobile map-based applications. Tilting around the x-axis has been used to adjust the zoom level while the user pans using the stylus (Büring et al. 2008). Büring et al.’s tilt zooming technique used a discrete approach, with pitch angles exceeding pre-defined thresholds used to activate zooming at a constant speed. Tilting the device upwards was used to activate zooming in, while tilting the phone downwards activated zooming out. Their results showed that some participants found this mapping between tilt and zooming directions confusing.

Touch and tilt interaction have also been combined to perform zooming operations, in a technique called Tilt-to-Zoom (Hinckley and Song 2011). Tilt-to-Zoom allows the user to select a point on the screen to be the focus of zooming. Zooming is then performed by tilting the phone around the x-axis while the user keeps his/her thumb still on the screen. In contrast to Büring et al.’s approach, rate control is used, with the change in pitch angle mapped onto the zooming speed. A neutral zone of five degrees either side of the neutral orientation is used to keep the zoom level constant. The same mapping between tilt direction and zooming action is used as Büring et al., although some participants in the subsequent evaluation felt that the mapping should be reversed. Tilt-to-zoom was well received by participants, although it was reported to be prone to overshooting.

Commercial applications of tilt zooming have also started to emerge. The Samsung Galaxy S2 smart phone employs position controlled tilt zooming in its browser and image gallery applications. The user is required to place two fingers on the screen for the duration of the zooming operation (Figure 3.8).

![Figure 3.8. Tilt zooming in the Samsung Galaxy S2](image_url)
Extensive research has recently been done regarding the use of accelerometers to detect discrete physical gestures for interacting with mobile applications, including performing zooming operations (Kela et al. 2006). Ruiz et al. (2011) conducted an exploratory study, in which participants were asked to design motion gestures for a series of common interactions required by mobile applications. The most commonly identified gestures for zooming in and out were gestures requiring users to flick the phone towards or away from them along the z-axis.

Camera-based interaction techniques have also been used to perform zooming. Special markers have been used to perform zooming with camera-based interaction (Winkler et al. 2007). The user uses these markers much like buttons, moving the camera over “zoom in” and “zoom out” markers in order to perform zooming operations. An alternative design made use of a single marker, with the camera moved towards or away from the marker to perform zooming. The use of markers is, however, not feasible for general purpose interaction in a mobile context of use.

Camera-based interaction techniques have been developed which simply require the user to move the mobile device forwards or backwards in order to zoom in or out. Optical flow and feature tracking algorithms, which attempt to track changes in successive images, are usually employed (Mooser et al. 2008). As with camera-based panning, techniques have been developed which rely on face detection, marker grids and physical maps as reference points for detecting movement of the mobile device (Hansen et al. 2005, Rohs et al. 2007).

Pressure sensing has been combined with stylus input in order to allow the user to perform zooming (Büring et al. 2008). The pressure with which the stylus is pressed against the screen is used to detect zooming input. Pressure input is divided into three classes. Soft touches are interpreted as zooming out gestures. Pressure in an intermediate range is interpreted as neutral input and is used to perform ordinary panning. Harder pressure is used to indicate zooming in. Participants in the subsequent evaluation complained that this technique was difficult to control. Pressure sensors have also been attached to the back of a phone in order to facilitate zooming operations (Miyaki and Rekimoto 2009).

### 3.4.3 Combined Panning and Zooming

Section 2.5.3 discussed the use of Speed-Dependent Automatic Zooming (SDAZ) for performing combined panning and zooming. SDAZ links the current zoom level to the
current panning speed. SDAZ has also been implemented using sensor-based interaction techniques. Eslambolchilar and Murray-Smith (2004) describe an implementation which relies on device tilting instead of stylus input in a mobile map-based application. Tilt-controlled SDAZ was used to perform combined zooming and panning in a mobile document browser. The system in question is shown in Figure 3.9. On the left of the figure is a mobile device with an accelerometer attached via the serial port. The next three screenshots show automatic zooming taking place as the user pans at decreasing speeds. In the left-most screenshot the user is panning quickly so the display is zoomed out. As the user decreases the angle of tilt around the x-axis, the panning speed slows. In the screenshot on the right, the device is now held in a neutral position and the display is no longer panning. Initial experiments revealed concerns about display visibility and the practicality of tilt interaction in a mobile context of use.

Figure 3.9. Tilt-controlled SDAZ in a mobile document browser (Eslambolchilar and Murray-Smith 2004)

Kratz and Rohs (2008) developed a variation on the tilt-controlled SDAZ technique described above. They identified that existing interaction techniques feature exclusively either manual or automatic zooming. They proposed a technique called SAZ, which combines an element of manual control over the zoom level with an SDAZ implementation (Kratz et al. 2010). Similar to Eslambolchilar and Murray-Smith (2004), they rely on tilt interaction to perform panning. Figure 3.10 shows a prototype of the SAZ interaction technique. The user is able to retain manual control over the zoom level using the touch-screen slider on the right of the display. SDAZ is used by default, but users are able to keep the zoom level fixed using the slider, should they so choose. Results of an evaluation comparing SAZ with SDAZ and multi-touch showed SAZ to be more efficient than SDAZ. Participants also reported better perceived workload and user satisfaction ratings for SAZ (Kratz et al. 2010).
Büring et al. (2008) implemented two sensor-based techniques for combining concurrent, but separate zooming and panning in stylus operated mobile devices. In the first technique, the stylus is used to perform panning, while the device is tilted about the x-axis to adjust the zoom level. In the second technique, the stylus is used to perform both panning and zooming. The pressure with which the stylus is pressed against the screen while the user is panning is used to adjust the zoom level. Their experimental results, however, showed that users didn’t take advantage of the capabilities provided by the techniques for performing both zooming and panning in parallel. The results also showed that users preferred tilt interaction for interacting with larger map-based displays.

### 3.4.4 Selection

Selection is an important operation, as it allows users of mobile map-based applications to interact with the content displayed on the map, such as points of interest, objects, places or people. Sensor-based interaction techniques often employ more conventional modalities such as keypad and touch-screen input to facilitate selection. Most of the sensors mentioned above for performing zooming and panning are continuous sensors which are well-suited for performing continuous tasks such as zooming and panning. Selection is generally a discrete action and traditional forms of input (which are largely discrete) including keypad and touch-screen interaction are well-suited to performing selection operations. A button positioned on the side of the mobile device has been used to allow selection operations to be performed with the same hand that is holding the mobile device (Mooser et al. 2008). Other researchers have made use of on-screen buttons, which users hold down while panning using tilt interaction and release when they wish to perform a selection (Oakley et al. 2004).
Some purely sensor-based implementations have been implemented to allow the user to perform selection without altering their grip on the phone. A fanning movement, in which the left hand side of the device is dipped and then raised relative to the y-axis, has been used to select thumbnail images in a mobile photo browsing application (Bartlett 2000). Timed selection techniques, where items are automatically selected after being highlighted for a pre-defined time period, have also been used in conjunction with both tilt interaction and pressure interaction (Crossan and Murray-Smith 2004, Wilson et al. 2011). Oakley et al. (2004) used vibration feedback to indicate to the user that he/she is passing over a selectable object in a mobile map-based application. The intensity of the vibrations increase the closer the user is to the selectable object. If the user remains within the area that triggers vibrations for longer than half a second, then the object in question is selected and related information is displayed to the user.

Other modalities that have been used to perform selection operations include pressure sensing and pen tilting. Pressure sensing interfaces usually make use of discrete pressure levels to distinguish selectable targets, such as menu options (Stewart et al. 2010). Tilting of the pen or stylus on a mobile device has also been used to perform selection operations by splitting the range of motion into discrete intervals (Xin, Bi and Ren 2011). These techniques both require the use of specialised hardware that has yet to achieve popular commercial success.

The following sections identify benefits and problems and shortcomings of the sensor-based interaction techniques discussed above.

### 3.5 Benefits

Sensor-based interaction techniques provide several advantages over traditional keypad and touch-screen interaction techniques. One of the most significant advantages of sensor-based interaction techniques is the ability to perform one-handed interaction with mobile devices (Hinckley and Song 2011, Miyaki and Rekimoto 2009). In contrast, touch-screen interaction techniques often require two-handed interaction (Section 2.6). Given that the user may be distracted or on the move (possibly holding other items), this is not ideal (Pascoe et al. 2000). A previous study has shown that, for a variety of mobile interaction tasks, including map interaction, the vast majority of users would prefer to only have to use one hand (Karlson et al. 2006). Sensor-based techniques allow the user to hold a mobile phone and perform interaction with the same hand. One-handed interaction also requires less visual attention than two-handed control (Eslambolchilar and Murray-Smith 2008b).
Previous studies have shown that users prefer using sensor-based techniques when walking (Lemmelä et al. 2008). Furthermore, sensor-based interaction techniques such as tilt interaction eliminate the occlusion problems associated with touch-screen interaction (Fitchett and Cockburn 2009). The user is able to interact with and view an unobscured display simultaneously (Cho et al. 2007b, Crossan et al. 2008). Sensor-based interaction techniques also do not require the use of on-screen controls, meaning that the entire display can be used for displaying information, rather than UI controls (Fitchett and Cockburn 2009). Sensor-based interaction techniques can, unlike touch-screen interaction, easily be used in cold conditions where the user may be wearing gloves (Fitchett and Cockburn 2009).

Keypad interaction techniques restrict users’ freedom of movement and offer little control over panning speeds (Kratz and Rohs 2008). Sensor-based interaction techniques, such as tilt interaction, facilitate more expressive input, providing greater freedom of movement and control over panning speeds. Sensor-based interaction techniques are frequently described as intuitive and often simply require the user to interact with the mobile phone as they would with objects in the real world (Cho et al. 2007b, Weberg et al. 2001, Bartlett 2000). Sensor-based interaction techniques allow for both continuous and discrete interaction, enabling a wide range of interaction possibilities (Hinckley and Song 2011).

Sensor-based interaction techniques have frequently been described as “fun” by participants in usability evaluations (Cho et al. 2007b, Büring et al. 2008, Kratz et al. 2010, Schwarten et al. 2008). Much of this can be attributed to the novelty of sensor-based interaction techniques, so it remains to be seen whether this perception remains in the long-term.

### 3.6 Problems and Shortcomings

Sensor-based interaction techniques also suffer from several problems and shortcomings. The discussion in Section 3.4 highlighted tilt interaction, physical movement of the mobile phone and pressure input as three of the most commonly used sensor-based interaction techniques. These techniques all suffer from a range of problems and consideration must be given to these issues when designing sensor-based interaction techniques.

#### 3.6.1 Physical Limitations

Tilt interaction is limited by the dexterity of the user’s wrist. Previous results have shown that users experience more difficulty performing tilting around the x-axis (vertical tilting) than when tilting around the y-axis (Crossan and Murray-Smith 2004, Mantyjarvi et al. 2006).
Figure 3.11 shows the maximum angle of rotation possible using different wrist-based gestures. As can be seen, a greater degree of flexibility is possible if the mobile phone is held flat in the user’s hand, as opposed to when the user needs to hold a thumb on the keypad.

Several other sensor-based interaction techniques also place limitations on the way the user can hold the mobile phone. Camera-based input techniques require that the camera lens is not obscured (Mooser et al. 2008). Techniques such as SAZ require a specific grip to be able to access the zoom level slider (Kratz et al. 2010). Tilt-to-Zoom also requires users to keep their thumb on the screen during interaction (Hinckley and Song 2011). The use of pressure sensors on the under-side of a mobile phone also requires the phone to be held in a particular manner (Miyaki and Rekimoto 2009). Restricting the user in terms of how the phone must be held introduces the possibility of discomfort and strain, particularly during prolonged use. When designing sensor-based interaction techniques, designers need to consider the limitations of the human wrist and the way users hold the mobile device (Büring et al. 2008).

![Figure 3.11. Maximum angle of rotation possible using different wrist-based gestures (Rahman et al. 2009)](image)

### 3.6.2 Toggling Sensor-based Control

Many sensor-based interaction techniques operate in a continuous fashion and rely on natural physical movements. While this can be an advantage, it can also easily lead to unintentional interaction taking place (Büring et al. 2008). It is essential, therefore, that designers of sensor-based interaction techniques incorporate some means of toggling sensor-based interaction on and off (Eslambolchilar and Murray-Smith 2004).
Several approaches have been used to address this problem. The use of some sort of clutch mechanism is common, where the user is required to keep a button depressed for the duration of interaction (Figure 3.12) (Rekimoto 1996, Oakley et al. 2004). As discussed in the previous section, this restricts how the user is able to hold the phone, introducing the possibility for discomfort. LG introduced a dedicated gesture button on the side of their Optimus Black phone, which users need to hold in when performing tilt gestures (LG 2011). Other techniques require the user to maintain contact with a touch-screen for the duration of interaction (Hinckley and Song 2011, Büring et al. 2008). While clutch mechanisms avoid the problem of inadvertent tilting, this approach can place physical strain on the user as an uncomfortable grip needs to be maintained for a long time (Büring et al. 2008).

An alternative approach is to make use of a toggle mechanism to switch sensor-based interaction on and off. Buttons, device squeezing and tilt gestures have been used to activate and deactivate tilt interaction (Mantyjarvi et al. 2006, Harrison et al. 1998, Bartlett 2000).

It has also been suggested that continuous monitoring of sensor readings can help to avoid accidental interaction. Sensor readings during deliberate interaction are very different from those typically observed as a result of random movement or when the device is resting on a surface. In this way, sensors can be used to identify when user interaction is intended and automatically toggle sensor interaction when necessary (Crossan and Murray-Smith 2004). This approach has not been widely employed.

3.6.3 Establishing a Neutral State

Tilt interaction implementations, particularly those relying on rate control, depend on orientation changes, rather than absolute orientation. It is important, therefore, to establish a
neutral angle, relative to which tilt interaction takes place. This helps to ensure that tilt interaction is correctly interpreted, given the current orientation of the mobile phone. Establishment of a neutral angle is often performed at the same time as tilt interaction is activated (Büring et al. 2008, Hinckley and Song 2011). Touch sensors and tilt gestures have both been used to establish a neutral orientation (Hinckley et al. 2000, Bartlett 2000). Crossan and Murray-Smith (2004) required calibration to take place before interaction, ensuring that the cursor was centred when the device was at its starting position. This approach introduces an additional step before interaction can occur which some users may perceive as an irritation. Pressure-based interaction techniques also require some sort of neutral state. Intermediate pressure values have previously been used for this purpose (Büring et al. 2008).

3.6.4 Display Visibility

Sensor-based interaction techniques requiring physical movement of the mobile phone introduce the possibility of display visibility problems. These could be as a result of screen reflections caused by tilting the phone (Eslambolchilar and Murray-Smith 2004). Alternatively, the interaction technique could involve gestures that result in viewing angles that are less than optimal (Oakley et al. 2004). Changes in display contrast have even been suggested as a means of compensating for the display visibility problems associated with tilt interaction (Hinckley et al. 2000).

3.6.5 Controllability

Tilt interaction techniques have been described as providing too coarse a level of control to be feasible for general use as an input technique (Karlson et al. 2006). Sensor-based interaction techniques used for panning also tend to suffer from overshooting problems, where users will pan past their intended target and have to reverse direction (Cho et al. 2007a, Hinckley and Song 2011, Crossan and Murray-Smith 2004). Interaction techniques which rely on cameras built into the mobile device suffer from the problem that the accuracy of this form of input depends to a large degree on the nature of the background image picked up by the camera. Camera-based input often does not work well when pointed at the ground because of a lack of variability in the background which makes movement difficult to detect (Mock and Rohs 2008).

In order to address the problem of overshooting, the use of “attractor” mechanisms has been suggested. Cho et al. (2007a) proposed this idea for use in a photo browsing application.
employing tilt interaction. They observed that users often had trouble converging on a particular target and that the display often settled in between two photos, rather than on a particular photo (contrary to user intentions). In order to address this problem, they modified their processing of tilt interaction to introduce a bias term to make it easier for users to settle on a particular photo. This approach could be beneficial to mobile map-based applications and could help make it easier for users to select icons on the map (Kratz and Rohs 2008).

Input from accelerometers contains a large amount of noise, which can make fine-grained control difficult to achieve using tilt interaction. This can make it difficult for users to converge on a target, with oscillation occurring as a result of hand tremor and disturbance in the signal from walking motion (Cho et al. 2007a, Crossan et al. 2008). As discussed in Section 3.3.2, smoothing algorithms are often employed to address this problem (Mock and Rohs 2008). Orientation information is calculated by averaging multiple sensor readings in order to eliminate spikes in the raw data. Discretisation has also been used to improve controllability (Rahman et al. 2009) (Section 3.4.1).

Sensor-based interaction techniques which require users to distinguish between discrete levels are also limited by human sensory and cognitive capabilities. For example, research regarding the use of pressure as a form of interaction has previously identified that users are not able to distinguish more than 7 distinct levels of pressure (Büring et al. 2008). Research regarding position controlled tilt interaction found that users were able to control between 8 and 12 distinct tilt levels, depending on the axis of rotation (Rahman et al. 2009).

Feedback can play an important role in helping users control sensor-based interaction. Various modalities have previously been used to provide feedback to users of sensor-based interaction techniques. Visual, audio and vibrotactile feedback have all previously been used in efforts to help users control tilt interaction (Eslambolchilar and Murray-Smith 2004). If visual feedback is to be provided, it is important that the display is still visible during interaction (Rahman et al. 2009). It has also been suggested that audio and haptic feedback can play an important role in a mobile environment, where the user’s attention may be divided, and can help support intermittent interaction, where the user is not constantly focused on the display (Eslambolchilar and Murray-Smith 2008b, Rath and Rohs 2006). In mobile map-based applications, users often do not keep their focus fixed on the display of their mobile device, and prefer to only look at the display when something of interest is being
displayed (Oakley et al. 2004). Non-visual feedback can even enable “eyes-free” interaction (Oakley and Park 2009).

Sound and vibrotactile feedback have been employed in mobile image gallery, document browsing and mobile map-based applications (Cho et al. 2007a, Eslambolchilar and Murray-Smith 2008b, Oakley et al. 2004). Vibration amplitude and pitch were used to indicate crossing of map gridlines and were linked to the rate of scrolling in different directions (Oakley et al. 2004). It has also been suggested that vibrotactile feedback can be useful when employing large tilt angles, where it may be difficult for the user to view visual feedback (Hughes et al. 2004).

3.6.6 Summary

The previous sections have highlighted how the various sensors available on mobile devices have been used to perform panning, zooming and selection. Tilt interaction, physical motion using camera input and pressure input were identified as the three most widely used forms of sensor-based interaction. All of these techniques suffer from significant problems and shortcomings. Tilt interaction is, however, superior to camera-based motion input and pressure input for several reasons, namely:

- Tilt interaction is feasible given current mobile technology. Pressure input is not currently widely supported.
- Camera input is dependent on the background not being of a uniform nature.
- Tilt input is well-suited to panning in large, unbounded information spaces such as maps, in contrast to both camera and pressure input.
- Tilt interaction can be directly measured and does not require complex image processing algorithms.
- Tilt interaction is better supported at an API level than camera and pressure input.
- Tilt interaction represents a more intuitive form of input for panning operations than pressure input.

The remainder of this research will therefore focus on tilt interaction as a sensor-based interaction technique. The following section discusses the various sensors that can be used to support tilt interaction.
3.7 Tilt Sensing

Accelerometer, gyroscope and magnetic field sensors can all play an important role in detecting orientation changes. These sensors are therefore widely used in tilt interaction techniques. Each of these sensors will now be discussed in more detail.

3.7.1 Accelerometers

Accelerometers detect acceleration along one or more axes. Accelerometers differ in several respects, including the number of axes relative to which they are capable of detecting acceleration, the range of acceleration they are able to detect and the unit of measurement in which they report acceleration. Most smart phones are now equipped with three-axis accelerometers which allow detection of acceleration relative to the x, y and z axes of a mobile phone (Figure 3.13). Accelerometers typically measure acceleration between -2g and 2g along all three axes.

![Figure 3.13. X, Y and Z axes, relative to which acceleration and orientation are calculated](image)

Accelerometers allow tilt angles to be determined using the acceleration detected due to gravity (Tuck 2007). The axis (or axes) relative to which this acceleration is detected depends on the orientation of the mobile device. Gravity is a fixed external source of acceleration, so this information can be used to calculate the absolute orientation of the mobile phone (Rohe and Essl 2007). In order to detect orientation changes, neutral angles relative to which the absolute orientation can be compared, need to be established (Rahman et al. 2009). Accelerometers also detect linear acceleration as a result of hand tremor, movement of the phone and walking (Vargha, Maia and Devries 2009). Distinguishing between the different sources of acceleration can be difficult (Nasiri, Sachs and Maia 2009). The use of the
accelerometer alone to calculate tilt orientation is therefore not ideal in a mobile context of use (Colton 2007).

### 3.7.2 Digital Compasses

Digital compasses (also sometimes referred to as magnetometers or magnetic field sensors) are now available on many smart phones. These digital compass sensors can be used to detect the ambient magnetic field relative to the x, y and z axes. This allows the compass heading of the phone to be determined while compensating for the orientation of the phone. Digital compasses are frequently used in mobile map-based applications and navigation applications to detect the current heading. The orientation of the map is sometimes adjusted in order to match the user’s current heading (Seager and Stanton Fraser 2007). The disadvantage of digital compass sensors is that they are susceptible to electromagnetic interference from (amongst others) vibration motors and audio speakers in phones. Furthermore, digital compass sensors are not adequate for processing rapid orientation changes, which is necessary for supporting responsive sensor-based interaction (Nasiri et al. 2009).

### 3.7.3 Gyroscopes

The widespread use of gyroscope sensors in mobile phones started with the release of the Apple iPhone 4 in June 2010 (Apple Inc. 2011b). Rapid growth in the use of gyroscope sensors has since taken place and the global market for gyroscopes in mobile phones is predicted to be worth $190 million US dollars in 2014 (iSuppli 2010). Gyroscopic sensors report angular velocity about the x, y and z-axis of the mobile phone. Angular velocity data can be integrated to retrieve absolute pitch, roll and heading angles. Successive readings can therefore be used to calculate orientation changes. Gyroscope input allows orientation changes to be rapidly detected, without the unwanted influence of linear acceleration (Nasiri et al. 2009). In theory, this would facilitate responsive, accurate tilt interaction with mobile applications. Unfortunately, integration of gyroscopic data to convert from angular velocity to angle data results in the gradual introduction of error. This results in a phenomenon known as gyroscopic drift, where gradual orientation changes are reported, even when the device is stationary and the reported angles diverge from the true orientation angles (Nasiri et al. 2010).
3.8 Existing Frameworks

Existing frameworks for sensor-based interaction have largely considered sensor-based interaction from a theoretical or design perspective. Benford et al. (2005) proposed a framework for planning and evaluating the design of sensor-based interaction techniques. This framework focuses on three aspects of the physical movements involved in interaction:

- Which physical movements are expected (by users) given the context of use?
- Which physical movements can be sensed given the available sensors?
- Which physical movements are desired given the application in question and the tasks users need to perform?

The authors of the framework suggest that the relationships and overlaps between the above three categories of movements should be considered to analyse existing sensor-based interaction techniques and to identify opportunities for the design of interaction techniques.

Bellotti et al. (2002) identified five questions which need to be considered in the design of sensor-based interaction techniques. These five questions are:

1. How do I address one (or more) of many possible devices?
2. How do I know the system is ready and attending to my actions?
3. How do I effect a meaningful action, control its extent and possibly specify a target or targets for my action?
4. How do I know the system is doing (has done) the right thing?
5. How do I avoid mistakes?

For each of these questions, Bellotti et al. (2002) identify associated challenges and possible problems. Many of the challenges and problems identified relate directly to those identified in Section 3.6. Questions 1 and 5 have associated challenges in terms of avoiding accidental interaction and differentiating between noise and actual input. Questions 2, 3 and 4 relate to the problems which exist in terms of controlling sensor-based interactions and the crucial role that feedback can play in helping users identify whether the system is responding correctly to their input.

Rahman et al. (2009) proposed a framework for analysing tilt interaction looking specifically at tilt interaction from a design perspective. They identified five key aspects of tilt interaction which affect users’ performance. These five aspects are the range of wrist motion required by
the interaction technique, the use of position or rate control, discretisation, the selection technique employed and the use of feedback. These five aspects will be considered in the design of the tilt interaction techniques employed in this research. Their results highlighted the benefits of quadratic discretisation, visual feedback and the use of position control for browsing menus.

The above frameworks all highlight similar themes that need to be considered in the design of a sensor-based interaction technique. These issues include:

- Selecting the most appropriate gestures, considering the task at hand and the user’s physical limitations;
- Providing appropriate feedback to support the user’s sense of control; and
- Designing sensor-based interaction techniques in such a way as to maximise controllability.

The following section discusses the level of API support for sensor-based interaction that currently exists within major mobile platforms.

### 3.9 Existing APIs

All the major mobile software platforms allow the development of third-party applications and therefore need to make information from sensors available to developers if such applications are to take full advantage of these sensors. In this section, the different sensor APIs available on these platforms are briefly analysed in order to determine the level of support which currently exists for the development of sensor-based interaction techniques. Java ME, Symbian C++, Windows Phone, Android and iOS APIs are discussed in this section as these APIs represent the dominant mobile software platforms available today.

#### 3.9.1 Java ME

Java ME provides a Mobile Sensor API for accessing sensor data (Nokia 2011a). The Mobile Sensor API allows developers to identify available sensors and to use data from these sensors in their applications. Retrieving sensor information using this API involves the following steps:

- Finding and identifying sensors;
- Opening a sensor connection;
• Retrieving sensor information (either synchronously or asynchronously through the use of data listeners). Conditions can also be set to only retrieve sensor information matching certain criteria; and
• Closing the sensor connection.

3.9.2 Symbian C++

Symbian C++ provides APIs which support the use of a wide array of sensors in mobile applications. The Symbian C++ Sensor APIs consist of two parts, namely the Sensor Channel API and the Sensor Definition API (Nokia 2011d). In some recent implementations, an additional API called the Sensor Data Compensator API is included, which can be used to automatically adjust sensor data according to the orientation of the display and the device.

The Sensor Channel API is used to access data from various sensors. This API functions very similarly to its Java counterpart. An application can search for available sensors, connect to a particular sensor and listen for data from the sensor in question (optionally setting conditions to listen for). Currently, the Symbian Sensor Channel API provides access to a wider range of sensors than the Java Mobile Sensor API on most Nokia devices, including proximity and ambient light sensors. The Sensor Definition API merely defines what information is available for each sensor, including the number of values and the corresponding data types.

3.9.3 Android

Android provides a listener interface to manage access to sensor data. Programmers can implement this interface and register to receive raw sensor data from a particular sensor in an asynchronous fashion. When registering an implementation of the listener interface, programmers can also specify the required sampling rate. Various sampling rates are provided for different purposes, including adjusting screen orientation, game input and user interface input. Android supports a wide array of sensors, including accelerometers, gyroscopes, magnetic field, temperature, pressure, proximity and ambient light sensors. Since Android 2.3, virtual sensor types have also provided separate access to linear acceleration and gravity data, both of which are detected by the accelerometer. Along with the raw sensor data, developers can access the accuracy of the sensor information and timestamp information. Android also provides methods to assist in calculating the current device orientation using accelerometer and magnetic field (compass) sensor data (Android 2011).
3.9.4 iOS

Initial iOS versions supported access to sensor data in much the same way as the other platforms. For example, accelerometer data can be accessed using the UIAccelerometer class, which provides raw accelerometer data along the x, y and z axes. Programmers can specify a delegate to receive accelerometer data at a specific update interval (Apple Inc. 2011a).

iOS Version 4 introduced gyroscope sensors to Apple mobile devices. The Core Motion framework was added, allowing programmers access not only to raw data, but also performing sensor fusion to calculate device orientation, rotation rate, linear acceleration and gravity. This data can be accessed either in a push or pull fashion. Pull access is more efficient, as the data is retrieved asynchronously and the latest values sampled when needed (Apple Inc. 2011a).

3.9.5 Windows Phone

Windows Phone is a comparatively new mobile platform and its sensor API has been improved from the initial version. Much like the other platforms, Windows Phone allows programmers to register event handlers to receive updates from a particular sensor. Accelerometer, compass and gyroscope sensors are all supported. Programmers are also able to specify a particular sampling rate.

A recent addition to Windows Phone is the Combined Motion API. This API shields the programmer from the underlying calculations needed to process sensor data and retrieve device orientation and movement data. The Combined Motion API allows programmers easy access to device orientation data (roll, pitch and yaw), rotational acceleration and linear acceleration data (Microsoft 2011b). At the time of writing, however, Windows Phones equipped with gyroscope sensors were not widely available.

3.9.6 Summary

Section 3.3 identified five major stages in processing sensor data for sensor-based interaction, namely sensor measurement, pre-processing, feature extraction, mapping and performing operations. During the course of this research, rapid improvement has taken place in terms of the level of support for sensor-based interaction in the mobile platforms discussed above. Whereas most mobile platforms initially only supported the first stage (Sensor Measurement), several platforms (Android, iOS and Windows Phone) now support pre-processing and feature extraction to some degree. Most of the research discussed in this thesis focuses on the
fourth and fifth stages identified in Figure 3.1 and investigates how sensor data can be used to perform low-level operations such as panning, zooming and selection.

### 3.10 Conclusions

Sensors provide access to a wealth of data regarding the current context of a mobile phone. This data can be used in the development of a wide range of expressive interaction techniques, utilising both discrete and continuous input.

Five major stages can be identified in processing sensor data for the purpose of sensor-based interaction. Firstly, sensor data is sampled at a particular sampling rate suitable for the technique in question. Secondly, pre-processing is often performed to remove signal noise and disturbances. A variety of smoothing algorithms are often used for this purpose. Thirdly, the processed sensor data is converted to a format suitable for interaction (such as orientation angles). In the fourth stage, the calculated data is mapped onto low-level operations such as panning, zooming and selection. Finally, the relevant operations are performed.

Tilt interaction, camera input and pressure sensing have all previously been employed to perform panning. For tilt interaction, the use of discretisation has been successfully used to improve controllability, by splitting the range of motion into discrete intervals. Position control has been shown to be effective for short-range panning (e.g. for menu navigation), but rate control provides a more practical solution for domains such as mobile map-based applications. Zooming techniques have also been implemented using tilt interaction, camera input and pressure sensing. Some of these techniques have involved performing zooming independently of panning. Others have linked the current zoom level to the current panning speed in order to minimise the need for switching between panning and zooming operations. Selection operations have been implemented using a wide variety of techniques, using both sensor-based techniques (including gestures, dwell-time and pressure) as well as more traditional techniques such as hardware buttons and touch-screen selection.

Existing sensor-based interaction techniques provide a wide range of benefits. One of the most frequently cited benefits is the ability to perform one-handed interaction. Tilt, camera input and pressure sensing all allow the user to perform interaction with a single hand. Furthermore, most sensor-based interaction techniques leave the display unobscured during interaction. Sensor-based interaction techniques are also intuitive and expressive and are frequently described as fun to use.
Sensor-based interaction techniques also suffer from several problems and shortcomings. Many of these shortcomings require designers of sensor-based interaction techniques to consider practical issues such as physical limitations and how the phone is held during interaction. Sensor-based interaction techniques need to be able to be deactivated and require the establishment of a neutral state relative to which interaction takes place.

Several clear advantages of tilt interaction can be identified as compared to other sensor-based interaction techniques such as camera input and pressure sensing. Tilt interaction is a more robust form of input and can be used independently of the current lighting and background conditions. Tilt interaction is better suited to browsing unbounded information spaces than camera and pressure input. Tilt interaction is also better supported at an API level than camera and pressure input. Pressure input has limitations in terms of the number of different levels users can perceive and is not widely supported by current mobile phones.

Accelerometers, digital compasses and gyroscopes are widely used to implement tilt interaction. These sensors, however, all suffer from individual shortcomings. Accelerometers are susceptible to disturbance from a variety of sources, including hand tremor and walking motion. Compass data is subject to interference from magnetic sources such as speakers. Gyroscopes suffer from drift error which is introduced when converting from angular velocity to orientation angles.

Existing frameworks for sensor-based interaction allow several important design issues to be identified. These include the importance of implementing intuitive gestures, the need for feedback to improve the user’s sense of control and the need for measures to compensate for the lack of controllability suffered by sensor-based interaction techniques.

Platform support for sensor-based interaction has improved dramatically over the course of this research. Several major platforms now support not only sensor measurement, but also calculation of device orientation, angular and linear acceleration data.

In the following chapter, the design of a basic sensor-based interaction technique is described, based on the best practices identified in this chapter. This technique was compared to keypad interaction, an established one-handed form of interaction. This evaluation was conducted in order to gain more insight into the problems and shortcomings of sensor-based interaction for supporting typical mobile map-based tasks.
Chapter 4: Designing a Basic Tilt Interaction Technique

4.1 Introduction

This chapter describes the design and evaluation of a basic sensor-based interaction technique for mobile map-based applications (van Tonder and Wesson 2010). This tilt interaction technique was developed based on related work regarding the use of tilt interaction and was designed to support panning and automatic zooming in a mobile map-based application. The design of the tilt interaction technique was undertaken in order to allow for experimentation to identify shortcomings of tilt interaction for performing typical mobile map-based tasks. Existing research has identified several shortcomings of tilt interaction (Section 3.6). For the purposes of this research, however, it is important to identify how these shortcomings affect users’ abilities to successfully complete typical mobile map-based tasks.

Tilt interaction was incorporated into a prototype mobile map-based application and compared to an equivalent keypad interaction technique. This comparative user study was undertaken in order to determine the relative merits of tilt interaction compared to keypad interaction for performing typical map-based tasks on a mobile device. Tilt interaction was selected as it was identified in the previous chapter as being the most suitable sensor-based interaction technique for mobile map-based applications (Section 3.6.6). Keypad interaction was selected for comparison purposes as it represents an established one-handed mobile interaction technique. The results of this user study will be used to identify opportunities for improving tilt interaction for mobile map-based applications.

4.2 Background

Existing studies of tilt interaction do not provide a thorough understanding of the relative merits of different one-handed interaction techniques for mobile map-based applications. Dong et al. (2005) compared tilt interaction to keypad interaction in a mobile map-based application. This study did not, however, consider the specific requirements of different types

---

1 This chapter includes content published in the following paper: Bradley van Tonder and Janet Wesson. 2010. Is tilt interaction better than keypad interaction for mobile map-based applications? In Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT '10), 322-331, ACM.
of mobile map-based tasks. In order to gain a better understanding of the advantages and disadvantages of tilt interaction as opposed to keypad interaction, a wider variety of mobile map-based tasks need to be examined. Büring et al. (2008) performed a more detailed study of mobile map-based tasks, but the use of tilt in this study was limited to zooming. Furthermore, their study was conducted in a simulated mobile environment (with very different ergonomic considerations from a mobile phone) and involved interaction techniques requiring the use of both hands. Tilt and keypad interaction have also previously been compared in a mobile image editing application (Schwarten et al. 2008). The results of this evaluation showed that keypad interaction was more efficient and also achieved greater accuracy than tilt interaction.

Dong et al.’s study showed no significant improvements in terms of efficiency, effectiveness or preference for the use of tilt interaction over keypad interaction. This study took place over five years ago. Since then, the use of sensor-based interaction techniques such as tilt interaction has become more pervasive. A larger percentage of the user population will likely have previous experience with tilt or gesture interaction. The purpose of this evaluation is to determine whether the results reported by Dong et al. also apply to other mobile map-based tasks and using more recent mobile and sensor technology. In contrast to existing research, the user study described in this chapter considers a wider range of variables, with perceived workload, controllability and ease of use also considered in addition to the metrics covered in Dong et al.’s study.

4.3 Tilt Interaction Technique

Tilt interaction has previously been successfully used to perform combined panning and zooming in a mobile document browsing application, using Speed-Dependent Automatic Zooming (SDAZ) (Section 3.4.3). Promising results have also been achieved using SDAZ in a mobile map-based application (Section 2.5.3). SDAZ eliminates the need to switch between separate panning and zooming controls. It was therefore decided to implement panning and zooming using an SDAZ approach, linking the zoom level to the current panning speed. Selection was implemented using a more traditional touch-screen input approach.

Figure 3.1 identified five stages involved in sensor-based interaction. Figure 4.1 illustrates how these five stages can be applied to implement tilt interaction in a mobile map-based application. Rendering of the updated display is included as a sixth stage. The design of the basic tilt interaction technique to be compared to keypad interaction is discussed in terms of
these six stages in the following sections. The technique is designed to incorporate best practices regarding the implementation of tilt interaction identified in Chapter 3.

![Diagram of six stages](image)

Figure 4.1. The different stages involved in using tilt interaction to perform panning and zooming

### 4.3.1 Sensor Measurement

Tilt interaction was initially implemented using the Java ME Mobile Sensor API to access data from the accelerometer. Java ME was selected as the implementation platform as it provided easy asynchronous access to sensor data. Due to a memory leak in the firmware version running on the Nokia N97 used for evaluation purposes it was necessary to implement this functionality in Python (PyS60) and send the sensor data to a Java ME application via a socket connection. No phones with gyroscope sensors were available at the time when the user study described in this chapter was conducted.

The Python Sensor API allows access to accelerometer data for the x, y and z axes. This data is reported in m/s² and is used to calculate the current orientation of the mobile phone. Acceleration data is collected asynchronously in a separate thread, in order to maximise
performance. This data is then collected in a circular buffer, with the last ten values retained. Accelerometer data is sampled as quickly as possible (an average frequency of 23 Hz).

4.3.2 Pre-Processing

Simple pre-processing of the accelerometer data is performed by calculating moving averages for the three acceleration axes. The last ten observations recorded in the circular buffer are used to calculate the average acceleration for the x, y and z axes. This has the effect of smoothing the accelerometer data and ensuring that outlier values do not cause sudden unpredictable input (Section 3.3.2).

4.3.3 Feature Extraction

Simple trigonometry is used to convert the smoothed acceleration data along the x, y and z axes into pitch and roll angles. Acceleration relative to the x, y and z axes is retrieved from the accelerometer and converted to pitch (about the x-axis) and roll (around the y-axis) angles, using Formulae 4.1 and 4.2 (Tuck 2007):

\[
\text{pitch} = \arctan\left(\frac{A_x}{\sqrt{A_y^2 + A_z^2}}\right) \quad (4.1)
\]

\[
\text{roll} = \arctan\left(\frac{A_y}{\sqrt{A_x^2 + A_z^2}}\right) \quad (4.2)
\]

where \(A_x\), \(A_y\) and \(A_z\) represent the acceleration along the x, y and z axes.

4.3.4 Mapping to Operations

A rate control mapping is used to control panning operations, with tilt angles mapped onto panning speeds (Section 3.4.1). While previous research has identified position control as being easier to use, rate control offers a more practical mapping for map-based applications, where users may be required to pan long distances (Rahman et al. 2009, Oakley and O’Modhrain 2005).

Discretisation is used to map pitch and roll angles onto panning speeds (Section 3.4.1). This process consists of splitting the range of motion into intervals, with tilt angles falling in the same interval resulting in the same effect. Discretisation has previously been employed to address the controllability problems of tilt interaction (Section 3.6.5). Quadratic discretisation is used, allowing for fine control with small tilt angles and faster panning with larger tilt
angles. Previous research has shown optimal results with this approach (Rahman et al. 2009). The user is able to move in any direction, with larger tilt angles resulting in faster panning speeds. The user is therefore able to fully control the direction and speed of panning. A “dead zone” is included to avoid unintentional panning as a result of minor orientation adjustments and hand tremor. Tilt angles smaller than 5 degrees are thus ignored.

Table 4.1 shows the discretisation intervals employed. Smaller intervals are used for slower panning speeds. At higher panning speeds, the intervals are larger. This allows for finer control of the panning speed for small adjustments, while also providing for easy acceleration for long-distance panning operations. Angles larger than 12 degrees result in the display zooming out. This is discussed in the following section. The mapping employed was based on previous research, with experimentation conducted to refine the size of the discretisation intervals and associated panning speeds (Dachselt and Buchholz 2008, Rahman et al. 2009).

<table>
<thead>
<tr>
<th>Tilt Angle (degrees)</th>
<th>Panning Speed (pixels/repaint)</th>
</tr>
</thead>
<tbody>
<tr>
<td>angle &lt;= 4</td>
<td>0</td>
</tr>
<tr>
<td>5 &lt;= angle &lt; 7</td>
<td>1</td>
</tr>
<tr>
<td>7 &lt;= angle &lt; 9</td>
<td>2</td>
</tr>
<tr>
<td>9 &lt;= angle &lt; 12</td>
<td>4</td>
</tr>
<tr>
<td>13 &lt;= angle</td>
<td>4 (SDAZ triggered)</td>
</tr>
</tbody>
</table>

Table 4.1 Discretisation of tilt angles and corresponding panning speeds

4.3.5 Performing Operations

Panning is split into horizontal and vertical panning operations. Horizontal panning takes place as a result of changes in the roll angle (tilting the phone left and right), while vertical panning takes place as a result of changes in the pitch angle (tilting the phone up and down). Horizontal and vertical panning operations are interleaved to give the user a free range of motion, rather than step-like panning. Panning speeds are calculated using the discretisation intervals in Table 4.1, with the differences in the relative speeds of horizontal and vertical panning providing for a full range of motion.

Zooming is performed using SDAZ (Section 3.4.3). The zoom level is linked to panning speed in order to prevent the problem of extreme visual flow, where the display can become blurred when panning at high speeds. In order to combat this problem, the display is
automatically zoomed out in order to prevent blurring. The zoom level is a function of the panning speed. According to the original implementation of SDAZ, the zoom level is calculated as the normalised displacement between the current tilt angle and the minimum tilt angle which triggers zooming (Formula 4.3):

\[
\text{scale} = s_o \left( \frac{d_y - d_0}{d_1 - d_0} \right)
\]  

(4.3)

Where \(s_o\) is the minimum scale, \(d_y\) is the current tilt angle, \(d_0\) is the minimum tilt angle which triggers zooming out and \(d_1\) is the maximum tilt angle beyond which no increase in panning speed takes place. The current panning speed was taken as the larger of the horizontal and vertical panning speed components.

The panning speed is determined using the discretisation intervals in Table 4.1. Once the panning speed is determined, this information is used to calculate the appropriate zoom level. Some previous implementations of SDAZ have also linked the panning speed to changes in the zoom level (Hinckley et al. 2000, Jones et al. 2005). The implementation described here employs a simpler approach, with the panning speed remaining constant once SDAZ is triggered. Therefore once SDAZ is triggered, instead of further panning speed increases taking place, the zooming scale is adjusted.

Figure 4.2 depicts the different state transitions which take place when panning and zooming using SDAZ. The idle state represents the default state where no panning or zooming takes place. If the tilt angle does not exceed a four degree angle, the application remains in the idle state. If the tilt angle exceeds four degrees, then the display is panned in the direction the device is tilted. If the tilt angle exceeds the threshold to trigger automatic zooming out, the display is zoomed out to a lower zoom level, while panning continues. If the tilt angle is reduced, the display is once again zoomed in. Once the tilt angle is less than or equal to four degrees, the application returns to the idle state where no panning or zooming takes place.

Tilt angles of less than 13 degrees do not trigger SDAZ. This threshold was designed to allow users to make minor adjustments to the cursor position without unwanted zoom level changes. The threshold of 13 degrees was chosen after experimentation with initial prototypes and analysis of the tilt angles employed during selection operations. Tilt angles of less than five degrees were ignored, as experimentation showed that this threshold allowed hand tremor and insignificant orientation adjustments to be ignored.
In order to avoid repeated zooming in and out when the tilt angle fluctuates around the angle which triggers zooming out, a technique known as asymmetric hysteresis was implemented (Miyaki and Rekimoto 2009). This technique ensures stable operation, by using different thresholds to activate zooming out and zooming in. Figure 4.3 illustrates this concept. In this case, if the same tilt angle was used as the threshold for zooming in and out, the zoom level would rapidly fluctuate. By using a lower threshold for zooming back in, however, this problem is avoided, as a more significant change in tilt angle (likely as the result of a deliberate adjustment) is required.
4.3.6 Rendering Display

The final stage in the tilt interaction process is updating the display to reflect user interaction. Once panning speed, direction and zoom level have been determined, the display is updated to reflect the appropriate latitude, longitude and zoom level.

The following section introduces the prototype which was used to compare tilt interaction with keypad interaction for performing mobile map-based tasks.

4.4 Prototype: MapExplorer

A prototype mobile map-based application, called MapExplorer, was created as a platform for comparing tilt interaction with keypad interaction for the purposes of performing mobile map-based tasks. MapExplorer was written in Java Micro Edition (Java ME), with Python used to access sensor data. A Nokia N97 (Figure 4.7), which provides for both keypad and touch-screen interaction, was used for testing purposes. Microsoft Bing Map tiles were used and a caching system implemented to maximise system performance. MapExplorer supports two main functions, namely:

- **Browsing Points of Interest (POIs):** Three categories of POIs are included in the system, namely restaurants, hotels and tourist attractions, each denoted by a different icon (Figure 4.4). Selecting a POI icon displays detailed information such as a description, contact details and opening hours (Figure 4.5).

- **Navigating:** The prototype allows users to plan and follow routes. A route planning web-service was used to calculate routes and retrieve routing instructions. By moving the cursor over route information markers, users are able to read routing instructions (Figure 4.6).

Six typical mobile map-based application tasks were identified in Section 2.3. MapExplorer supports four of these six tasks (locating, navigating, identifying and checking). Searching and interacting tasks were not supported as these tasks largely rely on text entry and the manipulation of standard user interface controls.

4.4.1 Identifying

Identifying tasks consist of the user determining the identity of a person, place or object. MapExplorer denotes POIs through the use of icons. A label and thumbnail preview image
are displayed when the user moves the cursor over one of these icons, allowing the user to identify the POI in question (Figure 4.4).

![Figure 4.4. POI icons in MapExplorer](image)

### 4.4.2 Locating

Locating tasks consist of determining the position of a person, place or object. MapExplorer provides a map view that the user is able to browse using zooming and panning operations. Users are then able to locate a particular POI by moving the cursor over POI icons to identify the icons in question. Several identifying tasks may need to be performed in order to locate the desired POI.

### 4.4.3 Checking

Checking involves determining the condition of a person, place or object (e.g. determining the opening hours for a shop). In MapExplorer, checking tasks require the user to select a POI in order to view more detailed information (Figure 4.5).

![Figure 4.5. Viewing detailed information for a POI](image)

### 4.4.4 Navigating

Navigating involves planning and following a route from start to finish. MapExplorer supports route planning through the use of a third-party web service provided by
CloudMade\(^2\). The user enters navigating mode by selecting the navigation button (indicated by number 3 in Figure 4.4). MapExplorer provides the service with the latitude and longitude of the start point and destination (selected by the user) and the preferred mode of transport (car, bicycle or pedestrian). Car transport was selected by default. The service then returns the route as a series of points, with routing instructions at the appropriate points. The response is formatted using JavaScript Object Notation (JSON). Figure 4.6 shows how a route is represented as a red line with a series of route markers in MapExplorer. The user can move the cursor over the different route markers in order to view routing instructions.

![Figure 4.6. Navigation in MapExplorer](image)

4.5 Keypad Interaction

In Section 2.4, three low-level tasks were identified which are required to perform the high-level tasks described above. These low-level tasks are panning, zooming and selection. The following section describes how these three tasks are performed using keypad interaction in MapExplorer.

4.5.1 Panning

The keypad version of MapExplorer supports panning through the use of the physical up, down, left and right directional keys (Figure 4.7). The user is therefore able to move in four different directions. An acceleration mechanism is built into the system to enable faster panning over long distances. Panning speed increases the longer a directional key is held down, up to a maximum speed. Panning speed is successively increased after a key is held down for 1.5, 3 or 5 second intervals.

\(^2\) www.cloudmade.com
Chapter 4: Designing a Basic Tilt Interaction Technique

4.5.2 Zooming

In the keypad version of MapExplorer, the user is able to perform zooming operations using touch-screen buttons (indicated by numbers 1 and 2 in Figure 4.4). The user is also able to use dedicated keypad buttons (“Q” and “A”) for this purpose. The proximity of these keys to the keys used for panning was the main motivation for this choice.

4.5.3 Selection

MapExplorer makes use of a cursor which is stationary in the centre of the screen. The user is able to perform a selection operation by moving the cursor over an icon and pressing the keypad button in the centre of the directional keys. When the cursor passes over a selectable POI icon, a preview thumbnail image is displayed, along with the name of the POI (Figure 4.4). Pressing the selection button then displays more detailed information for that POI (Figure 4.5).

4.6 Tilt Interaction

Tilt interaction was implemented using the tilt interaction technique described in Section 4.3. Implementation issues specific to MapExplorer are discussed in the following sections.

4.6.1 Panning

Previous research has highlighted the importance of providing a means of toggling tilt interaction and establishing a neutral state relative to which tilt angles are calculated (Sections 3.6.2 and 3.6.3). In order to prevent accidental panning in MapExplorer, the user is required to toggle panning mode when required. This is done by tapping anywhere on the phone display. The status bar at the top of the screen indicates whether tilt-controlled panning
is currently activated (Figure 4.4). Activating panning mode also establishes the neutral orientation. Pitch and roll angles are calculated relative to this orientation. In order to reset the neutral orientation, the user has to deactivate panning mode, orientate the phone as required and then reactivate panning mode. In order to avoid problems with display visibility and physical discomfort, the effective range of motion for both pitch and roll angles was limited to 17 degrees either side of the neutral position. Tilting beyond this does not affect panning speed or zoom level.

Very simple visual feedback is provided to the user in order to indicate the direction and speed of panning. This information is indicated through the use of arrows at the border of the screen (Figure 4.8).

### 4.6.2 Zooming

Automatic zooming is depicted in Figures 4.8 – 4.10. Figure 4.8 shows panning at low speeds (where the zoom level is not adjusted). Figure 4.9 shows panning at medium speed, while Figure 4.10 shows panning at high speed (where the display is zoomed out two levels).

When automatic zooming takes place, a red rectangle is displayed to denote the area that will fill the screen when the tilt angle is decreased again (Figures 4.9 and 4.10). Such an approach has been used before in the tilt interaction SDAZ implementation used for document browsing described by Eslambolchilar and Murray-Smith (2008a). A zoom level indicator is also temporarily displayed while the user is zooming (Figure 4.9). Automatic zooming is configured to zoom out a maximum of two discrete zoom levels from the starting zoom level (25% of the starting scale). Animated transitions were employed to make automatic zooming as smooth as possible. Manual zooming can still be performed using the on-screen controls to allow for zooming independently of panning.
4.6.3 Selection

Previous research has suggested that continuous input, such as tilt interaction, is not well suited to performing discrete operations such as selections (Rogers and Muller 2006). For this reason, selection in MapExplorer was implemented using touch-screen interaction, rather than tilt interaction. Selection operations are performed by moving the cursor over an icon and tapping anywhere on the screen to perform a selection operation.

Table 4.2 summarises how panning, zooming and selection are supported using keypad and tilt interaction in MapExplorer.

<table>
<thead>
<tr>
<th>Panning</th>
<th>Zooming</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keypad</td>
<td>Directional keys</td>
<td>Keypad selection key</td>
</tr>
<tr>
<td>Tilt</td>
<td>Tilt gestures (Rate control)</td>
<td>SDAZ (triggered by tilt gestures) or touch-screen buttons</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tapping touch-screen</td>
</tr>
</tbody>
</table>

Table 4.2. Keypad and tilt interaction support for panning, zooming and selection
4.7 Experimental Design

A controlled user study was conducted to compare tilt interaction to keypad interaction for performing typical mobile map-based tasks. While a controlled evaluation does not take into account variables such as context of use, this approach allows the two interaction techniques to be compared in a uniform environment across all participants. Using this approach, quantitative performance data (such as task times) are not influenced by external distractions.

4.7.1 Objectives

The primary objective of the evaluation was to compare tilt interaction to keypad interaction for the purposes of performing typical mobile map-based tasks. This was to identify shortcomings of tilt interaction as compared to keypad interaction and to identify opportunities for improving tilt interaction in the domain of mobile map-based applications.

4.7.2 Metrics

The following self-reported metrics were collected:

- **Perceived workload**: This was measured using the six factors of the NASA-TLX questionnaire (mental demand, physical demand, temporal demand, performance, effort and frustration) (Hart and Staveland 1988).
- **Efficiency**: Perceived efficiency was measured for each group of tasks. Overall perceived efficiency was also measured for each interaction technique.
- **Effectiveness**: Perceived effectiveness was measured for each group of tasks. Overall perceived effectiveness was also measured for each interaction technique.
- **Controllability**: Perceived controllability was measured for each group of tasks. Overall perceived controllability was also measured for each interaction technique.
- **Ease of Use**: Perceived ease of use was measured for each group of tasks. Overall perceived ease of use was also measured for each interaction technique.
- **Preference**: Participants were asked to rate their preferred interaction technique on a seven-point interval scale (where 1 = tilt and 7 = keypad).

The following performance metric was measured:

- **Efficiency (task times)**: Efficiency data was measured using a logging mechanism built into the prototype. The start time for each task was measured as the time when
the participant closed the dialogue displaying task instructions. The end time was measured as the time when the participant confirmed task completion.

The NASA-TLX factors are frequently used to assess and compare interaction techniques, and have been used in several previous related studies (Büring et al. 2008, Jones et al. 2005). Efficiency and effectiveness were both included because they are both fundamental aspects of usability. Furthermore, efficiency, effectiveness and preference have all been considered in related studies (Dong et al. 2005, Büring et al. 2008). Controllability was included as it has previously been identified as a problem area for tilt interaction (Section 3.6.5). Preference was elicited as an overall comparative measure of the two interaction techniques.

4.7.3 Participants

Thirty-two participants were recruited for the evaluation. All participants were students at the Nelson Mandela Metropolitan University (NMMU) and were predominantly students in the Department of Computing Sciences. Participants were all in the age range 18-29 and included both undergraduate and postgraduate students. Six participants were left-handed. Participants of ten different nationalities took part in the evaluation.

Twenty participants reported prior experience with tilt interaction, with the Nintendo Wii, various phone models and the Apple iPod Touch listed among the devices used. Most participants reported occasional use of mobile map-based applications.

4.7.4 Tasks

Tasks were grouped into three sub-groups, according to task type, namely locating, navigating and checking tasks. Fictional POIs were used to avoid any performance benefit being obtained from local knowledge.

- **Locating tasks:** Participants were given the name of a POI and required to locate the POI on the map. Selection was not required. The final locating task did not provide a name, but asked the user to locate the nearest POI falling in a particular category.

- **Navigating tasks:** Participants were given the names of the start and end points of a route (both POIs) and asked to plan and follow a route from the start point to the end point.

- **Checking tasks:** Participants were given the name of a POI and asked to retrieve a specific detail for that POI (opening hours, phone number, etc.).
These three groups of tasks cover four of the six task types identified in Section 2.3, given that locating and navigating both incorporate identifying tasks.

Participants completed five locating tasks, four navigation tasks and four checking tasks for each interaction technique. Participants were given an opportunity to experiment with the interaction techniques before commencing the tasks. The first task in each group was considered a training task and not used for timing purposes.

4.7.5 Independent Variables

A within-subjects approach was used for all test conditions. The following independent variables were tested in this user study:

- **Interaction Technique:** All participants used both the keypad and tilt interaction techniques and the order in which the interaction techniques were used was counterbalanced to offset any order effects.

- **Task Type:** All participants performed locating, navigating and checking tasks. Two similar task sets were used and counterbalanced across participants to offset any learning effect. The tasks involved were identical, except for the start and end points (for navigation tasks) and POIs involved. The task sets were also designed to involve similar panning distances. Four different permutations of interaction technique and task set were therefore possible, with eight participants performing each permutation.

4.7.6 Test Environment

The user study was conducted in a laboratory environment at the Department of Computing Sciences at NMMU (Figure 4.11).

![Figure 4.11. Evaluation participant using tilt interaction in MapExplorer](image)
4.7.7 Materials
Participants were provided with a Nokia N97 with MapExplorer pre-installed.

4.7.8 Procedure
Participants were provided with an informed consent form, describing the goals of the study and the data to be collected (Appendix A). Participants were required to read and sign this form at the beginning of the user study. Participants were also asked to complete a pre-test biographical questionnaire (Appendix B). A brief overview of the system was then provided, along with a demonstration of how to use the first interaction technique. Participants were allowed a few minutes to experiment with the interaction technique and to ask any questions relating to its use.

Tasks were presented sequentially to participants via the user interface (Figure 4.12). Participants were required to select a “task complete” touch-screen button (indicated by number 4 in Figure 4.4) when satisfied that they had successfully completed a task (Figure 4.13). A printed copy of the task instructions was presented to users for ease of reference. Tasks were divided into groups according to task types. Participants were required to complete a questionnaire upon completion of each group of tasks. This questionnaire was based on the six factors of the NASA-TLX workload questionnaire (mental demand, physical demand, temporal demand, performance, effort and frustration) (Hart and Staveland 1988) and included four questions measuring perceived effectiveness, efficiency, controllability and ease of use based on the standard After-Scenario Questionnaire (Appendix C) (Lewis 1991). Once all the tasks had been completed with the first interaction technique, the process was repeated with the second interaction technique. Seven point interval scales were used throughout.

Figure 4.12. Task instructions presented to participants
Participants were required to complete a post-test questionnaire once all tasks were completed. Participants were required to rate their preferred interaction technique on a seven point interval scale (where 1 = tilt and 7 = keypad interaction). The post-test questionnaire also required participants to rate both interaction techniques in terms of efficiency, effectiveness, ease of use and controllability. This portion of the questionnaire was the same as the post-task questionnaire. Finally, participants were required to state one positive and negative aspect of each interaction technique. The post-test questionnaire is shown in Appendix D.

4.8 Results: Self-Reported Metrics

The following sections report on the results of several comparative statistical tests, comparing keypad and tilt interaction. In all cases, tests for statistical significance (using $\alpha = 0.05$) were conducted prior to testing for practical significance. Therefore, where a difference is reported to be practically significant, it is also statistically significant. Where a difference was not statistically significant, practical significance was not considered.

The results are grouped according to the three groups of tasks which were considered (locating, checking and navigating tasks). Overall results (post-test questionnaire) are also presented.

4.8.1 Locating Tasks

Table 4.3 presents the results of t-tests comparing mean workload and user satisfaction ratings for keypad interaction and tilt interaction for the locating group of tasks. Mean interval scale ratings for the subjective workload categories were all lower (better) for keypad interaction than for tilt interaction. The differences were statistically significant for mental demand ($p = 0.042$), temporal demand ($p = 0.004$), performance ($p = 0.027$) and effort ($p =$
0.034). Cohen’s d values indicated small practical significance for mental demand, performance and effort and a medium practical difference for temporal demand.

Mean ratings for perceived effectiveness, efficiency, controllability and ease of use were also better for keypad interaction, but none of these differences were statistically significant.

<table>
<thead>
<tr>
<th></th>
<th>Keypad</th>
<th>Tilt</th>
<th>t (31)</th>
<th>p-value</th>
<th>Cohen’s d</th>
<th>Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mental Demand</td>
<td>2.50</td>
<td>3.16</td>
<td>-2.12</td>
<td><strong>0.042</strong></td>
<td>0.37</td>
<td>Small</td>
</tr>
<tr>
<td>Physical Demand</td>
<td>2.81</td>
<td>3.19</td>
<td>-1.09</td>
<td>0.284</td>
<td>n.a.</td>
<td>Not</td>
</tr>
<tr>
<td>Temporal Demand</td>
<td>2.28</td>
<td>3.03</td>
<td>-3.16</td>
<td><strong>0.004</strong></td>
<td>0.56</td>
<td>Medium</td>
</tr>
<tr>
<td>Performance</td>
<td>1.53</td>
<td>2.03</td>
<td>-2.32</td>
<td><strong>0.027</strong></td>
<td>0.41</td>
<td>Small</td>
</tr>
<tr>
<td>Effort</td>
<td>2.63</td>
<td>3.34</td>
<td>-2.21</td>
<td><strong>0.034</strong></td>
<td>0.39</td>
<td>Small</td>
</tr>
<tr>
<td>Frustration</td>
<td>2.19</td>
<td>2.47</td>
<td>-1.04</td>
<td>0.306</td>
<td>n.a.</td>
<td>Not</td>
</tr>
<tr>
<td>Effectiveness</td>
<td>5.84</td>
<td>5.53</td>
<td>1.06</td>
<td>0.299</td>
<td>n.a.</td>
<td>Not</td>
</tr>
<tr>
<td>Efficiency</td>
<td>5.44</td>
<td>5.34</td>
<td>0.28</td>
<td>0.779</td>
<td>n.a.</td>
<td>Not</td>
</tr>
<tr>
<td>Controllability</td>
<td>5.63</td>
<td>5.31</td>
<td>0.92</td>
<td>0.366</td>
<td>n.a.</td>
<td>Not</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>5.69</td>
<td>5.53</td>
<td>0.52</td>
<td>0.608</td>
<td>n.a.</td>
<td>Not</td>
</tr>
</tbody>
</table>

Table 4.3. Results of t-tests comparing keypad and tilt interaction for locating tasks (bold p-values are statistically significant) (n=32)

### 4.8.2 Navigating Tasks

Table 4.4 presents the results of t-tests comparing mean workload and user satisfaction ratings for keypad interaction and tilt interaction for the navigating group of tasks. Mean workload ratings were lower for tilt interaction than for keypad interaction for all categories except mental demand. None of these differences were, however, statistically significant using an alpha value of 0.05.

Mean ratings for perceived effectiveness, efficiency, controllability and ease of use were also better for tilt interaction. The differences in terms of controllability (p = 0.020) and ease of use (p = 0.025) were both statistically significant. Effectiveness (p = 0.070) and efficiency (p = 0.073) were narrowly not statistically significant. Cohen’s d values indicated small practical significance for the differences in controllability and ease of use.
### 4.8.3 Checking Tasks

Table 4.5 presents the results of t-tests comparing mean workload and user satisfaction ratings for keypad interaction and tilt interaction for the checking group of tasks.

<table>
<thead>
<tr>
<th>Keypad</th>
<th>Tilt</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
</tr>
<tr>
<td>Mental Demand</td>
<td>1.91</td>
<td>1.15</td>
</tr>
<tr>
<td>Physical Demand</td>
<td>2.38</td>
<td>1.21</td>
</tr>
<tr>
<td>Temporal Demand</td>
<td>2.34</td>
<td>1.33</td>
</tr>
<tr>
<td>Performance</td>
<td>1.69</td>
<td>1.42</td>
</tr>
<tr>
<td>Effort</td>
<td>2.22</td>
<td>0.91</td>
</tr>
<tr>
<td>Frustration</td>
<td>1.94</td>
<td>1.16</td>
</tr>
<tr>
<td>Effectiveness</td>
<td>5.97</td>
<td>1.26</td>
</tr>
<tr>
<td>Efficiency</td>
<td>5.81</td>
<td>1.26</td>
</tr>
<tr>
<td>Controllability</td>
<td>5.78</td>
<td>1.29</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>5.72</td>
<td>1.28</td>
</tr>
</tbody>
</table>

Table 4.5. Results of t-tests comparing keypad and tilt interaction for checking tasks (bold p-values denote statistical significance) (n=32)
Mean workload ratings were in favour of keypad interaction for mental demand, physical demand, effort and frustration. The mean rating for performance was in favour of tilt interaction. Equal mean ratings were reported for temporal demand. Only the difference in terms of mental demand was statistically significant ($p = 0.021$). The corresponding Cohen’s $d$ value ($d = 0.43$) indicates a small practical significance. Mean ratings for effectiveness, efficiency, controllability and ease of use were all better for tilt interaction, although none of these differences were statistically significant. The difference in terms of efficiency was narrowly not statistically significant ($p = 0.068$).

4.8.4 Overall Satisfaction

Participants were asked to rate their preferred interaction technique on a seven-point interval scale (Tilt = 1 and Keypad = 7). Nineteen of the 32 participants rated tilt interaction as their preferred technique (ratings of 3 or lower) and 12 participants rated keypad interaction as their preferred technique (ratings of 5 or higher). One participant selected the intermediate value of 4. The mean rating was 3.5 with a median value of 3. The difference from the intermediate value of 4 was not statistically significant at an alpha value of 0.05 ($p = 0.174$). Figure 4.14 illustrates the distribution of participants’ responses, demonstrating the preference for tilt interaction amongst participants.

![Distribution of Preferred Interaction Technique Responses](image)

**Figure 4.14. Distribution of ratings regarding participants' preferred interaction technique ($n=32$)**
As part of the post-test questionnaire, participants were also asked to rate the two interaction techniques in terms of controllability, efficiency, effectiveness and ease of use (Figure 4.15). Mean ratings for controllability, effectiveness and ease of use were greater for keypad interaction, while the mean rating for efficiency was greater for tilt interaction. Table 4.6 shows the results of t-tests comparing the means of the two interaction techniques. Controllability was the only metric for which a significant difference was recorded, with participants finding keypad interaction significantly easier to control than tilt interaction ($p = 0.018$). The corresponding Cohen’s $d$ value (0.44) indicates a small practical significance.

![Figure 4.15. Mean interval scale ratings for post-test questionnaire (95% confidence intervals shown) ($n=32$)](image)

<table>
<thead>
<tr>
<th></th>
<th>Keypad</th>
<th>Tilt</th>
<th>Significance</th>
<th>Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Controllability</strong></td>
<td>Mean 5.63, SD 1.29</td>
<td>Mean 4.63, SD 1.34</td>
<td>$t(31) = 2.51$, $p = 0.018$</td>
<td>Cohen's $d = 0.44$, Small</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>Mean 4.69, SD 1.57</td>
<td>Mean 5.25, SD 1.44</td>
<td>$t(31) = -1.20$, $p = 0.239$</td>
<td>n.a., Not</td>
</tr>
<tr>
<td><strong>Effectiveness</strong></td>
<td>Mean 5.91, SD 1.25</td>
<td>Mean 5.41, SD 1.27</td>
<td>$t(31) = 1.46$, $p = 0.154$</td>
<td>n.a., Not</td>
</tr>
<tr>
<td><strong>Ease of Use</strong></td>
<td>Mean 5.63, SD 1.41</td>
<td>Mean 5.28, SD 1.22</td>
<td>$t(31) = 0.90$, $p = 0.376$</td>
<td>n.a., Not</td>
</tr>
</tbody>
</table>

Table 4.6. Results of t-tests comparing overall controllability, efficiency, effectiveness and ease of use for keypad and tilt interaction (bold $p$-values denote statistical significance) ($n=32$)
A section for qualitative feedback was also provided, with participants asked to rate the most positive and negative aspect of each interaction technique. Tables 4.7 and 4.8 show frequency information for themes identified in this section of the post-test questionnaire.

Positive feedback for keypad interaction focused on the precision and ease of control that this interaction technique provided (Table 4.7). The familiarity and ease of use of the keypad interaction technique was also favourably received. Negative feedback regarding keypad interaction focused on a perceived lack of efficiency. Participants also felt constrained by the limited freedom of movement offered by this interaction technique (up, down, left and right). Several users also commented that they found keypad interaction to be physically demanding.

<table>
<thead>
<tr>
<th>Category</th>
<th>n</th>
<th>Example Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keypad Interaction – Positive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controllability</td>
<td>14</td>
<td>“Easy to control”</td>
</tr>
<tr>
<td>Precision</td>
<td>10</td>
<td>“Ability to navigate to precise locations and not overshoot targets”</td>
</tr>
<tr>
<td>Familiarity</td>
<td>3</td>
<td>“Familiarity of use was there”</td>
</tr>
<tr>
<td>Keypad Interaction - Negative</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Freedom of Movement</td>
<td>12</td>
<td>“Lack of natural movement, e.g. diagonal movement”</td>
</tr>
<tr>
<td>Efficiency</td>
<td>10</td>
<td>“Slower than tilt interaction”</td>
</tr>
<tr>
<td>Physical Demand</td>
<td>7</td>
<td>“…a lot of physical effort required”</td>
</tr>
</tbody>
</table>

Table 4.7. Frequencies of top three positive and negative themes for each keypad interaction

Positive feedback for tilt interaction highlighted the efficiency of this interaction technique (Table 4.8). Participants also liked the freedom of movement offered by tilt interaction. Several participants liked the automatic zooming that took place when panning at high speeds and the fact that this reduced the amount of manual zooming necessary. Participants also commented favourably on the intuitive nature of tilt interaction, the lack of physical effort required and the ability to control panning speed.

Negative feedback highlighted that users found tilt interaction to be difficult to control. Several participants found automatic zooming to be distracting or control inhibiting. Some participants also felt that tilt interaction would take a while to learn to use.
Chapter 4: Designing a Basic Tilt Interaction Technique

<table>
<thead>
<tr>
<th>Category</th>
<th>n</th>
<th>Example Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tilt Interaction – Positive</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficiency</td>
<td>11</td>
<td>“Fast and responsive”</td>
</tr>
<tr>
<td>Freedom of Movement</td>
<td>9</td>
<td>“Freedom to move around or search quickly with less</td>
</tr>
<tr>
<td></td>
<td></td>
<td>effort than keypad”</td>
</tr>
<tr>
<td>Automatic Zooming</td>
<td>5</td>
<td>“Fast and auto-zoom makes finding distant POIs easier”</td>
</tr>
<tr>
<td><strong>Tilt Interaction - Negative</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controllability</td>
<td>15</td>
<td>“Not as accurate when stopping on exact locations”</td>
</tr>
<tr>
<td>Automatic Zooming</td>
<td>8</td>
<td>“Zooming in and out can be annoying”</td>
</tr>
<tr>
<td>Learnability</td>
<td>4</td>
<td>“The need for experience to develop control”</td>
</tr>
</tbody>
</table>

Table 4.8. Frequencies of top three positive and negative themes for tilt interaction

### 4.9 Results: Performance Metrics

Table 4.9 shows the results of t-tests comparing mean task times for tilt and keypad interaction for each of the three groups of tasks.

<table>
<thead>
<tr>
<th></th>
<th>Keypad Mean</th>
<th>Keypad SD</th>
<th>Tilt Mean</th>
<th>Tilt SD</th>
<th>t (31)</th>
<th>p-value</th>
<th>Cohen's d</th>
<th>Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locating</td>
<td>118.93</td>
<td>55.12</td>
<td>161.87</td>
<td>64.98</td>
<td>-3.89</td>
<td>0.001</td>
<td>0.69</td>
<td>Medium</td>
</tr>
<tr>
<td>Navigating</td>
<td>298.52</td>
<td>132.06</td>
<td>292.84</td>
<td>109.57</td>
<td>0.28</td>
<td>0.779</td>
<td>n.a.</td>
<td>Not</td>
</tr>
<tr>
<td>Checking</td>
<td>78.94</td>
<td>35.91</td>
<td>115.01</td>
<td>51.35</td>
<td>-3.16</td>
<td>0.003</td>
<td>0.56</td>
<td>Medium</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>496.40</td>
<td>196.40</td>
<td>569.73</td>
<td>181.43</td>
<td>-2.13</td>
<td>0.042</td>
<td>0.38</td>
<td>Small</td>
</tr>
</tbody>
</table>

Table 4.9. Results of t-tests comparing mean total task times for each group of tasks for keypad and tilt interaction (n=32)

#### 4.9.1 Locating Tasks

The mean task completion time using keypad interaction for the locating tasks was 118.93 seconds, while a mean completion time of 161.87 seconds was recorded for tilt interaction. This difference was statistically significant (p = 0.001). The Cohen’s d value of 0.69 indicates a medium practical significance.

#### 4.9.2 Navigating Tasks

The mean task completion time using keypad interaction for the navigating tasks was 298.52 seconds, while a mean completion time of 292.84 seconds was recorded for tilt interaction. This difference was not statistically significant.
4.9.3 Checking Tasks

The mean task completion time using keypad interaction for the checking tasks was 78.94 seconds, while a mean completion time of 115.01 seconds was recorded for tilt interaction. This difference was statistically significant (p = 0.003). The Cohen’s d value of 0.56 indicates a medium practical significance.

4.9.4 Overall

Figure 4.16 and Table 4.9 provide a summary of the task times across the different groups of tasks for keypad and tilt interaction. Keypad interaction outperformed tilt interaction in locating and checking tasks, but tilt interaction was slightly more efficient for navigating tasks. The total time taken to complete all tasks was lower for keypad interaction than tilt interaction. This difference was statistically significant (p = 0.042). The Cohen’s d value of 0.38 indicates a small practical significance.

![Mean Task Times](image)

Figure 4.16. Mean total task times organised by task type (95% confidence intervals shown) (n=32)

4.10 Results: Correlations

Table 4.10 shows a breakdown of participants’ preferred interaction technique according to prior experience with tilt interaction (20 of the 32 participants indicated prior experience). There is a noticeable difference in terms of preference between those with prior tilt

interaction experience and those without prior experience. This difference is narrowly not statistically significant (Chi-square(1) = 3.18, p = 0.075).

<table>
<thead>
<tr>
<th>Previous Tilt Experience</th>
<th>Preferred Technique</th>
<th>Tilt (n)</th>
<th>Keypad (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No (n=12)</td>
<td>Tilt</td>
<td>5 (42%)</td>
<td>7 (58%)</td>
</tr>
<tr>
<td>Yes (n=19)</td>
<td>Tilt</td>
<td>14 (74%)</td>
<td>5 (26%)</td>
</tr>
</tbody>
</table>

Table 4.10. Preferred interaction technique vs. previous experience with tilt interaction (1 participant indicated a neutral preference rating and is not included)

Table 4.11 shows a breakdown of participants’ preferred interaction technique according to prior experience with tilt interaction on a mobile phone (Nintendo Wii not included). There is a noticeable difference in terms of preference between those with prior tilt interaction experience and those without prior experience. In this case, the difference is statistically significant (Chi-square(1) = 5.13, p = 0.023).

<table>
<thead>
<tr>
<th>Previous Phone Tilt Experience</th>
<th>Preferred Technique</th>
<th>Tilt (n)</th>
<th>Keypad (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No (n=18)</td>
<td>Tilt</td>
<td>8 (44%)</td>
<td>10 (56%)</td>
</tr>
<tr>
<td>Yes (n=13)</td>
<td>Tilt</td>
<td>11 (85%)</td>
<td>2 (15%)</td>
</tr>
</tbody>
</table>

Table 4.11. Preferred interaction technique vs. previous experience with tilt interaction on a mobile phone

Pearson product moment correlations were also calculated between participants’ questionnaire responses and task times. These results indicated that participants’ questionnaire responses regarding workload were influenced by the time it took them to complete the tasks. Furthermore, a statistically significant correlation of 0.42 was calculated between the amount of time it took participants to complete the tasks using tilt interaction and the interval scale rating for preference. The longer participants took to complete the tasks using tilt, the more likely they were to indicate a preference for keypad interaction. This would suggest that the preference ratings given by participants were not merely influenced by the novelty of tilt interaction, but that efficiency played an important role.

Pearson product moment correlations were calculated between participants’ task completion times and the mean and variance pitch and roll angles they employed. This was done in order to ascertain whether any noticeable trends can be observed between participants’ tilting behaviour and the resultant task times. Table 4.12 shows the resultant Pearson product
moment correlations. In all but one case, negative correlations were observed, implying that larger tilt angles (and greater variances) were employed by participants who achieved shorter task times. This is to be expected, as participants who were able to complete tasks more rapidly were more likely to have employed larger pitch and roll angles (which result in faster panning speeds). The fact that larger tilt angles allow for faster task completion times supports the assertion that the opportunity to control speed offered by tilt interaction results in improved efficiency (Section 3.5). These correlations were larger for navigating tasks, where panning speed plays an important part (navigation tasks comprised more than half of the total evaluation time, and therefore had a greater influence on the overall correlations). Much smaller correlations were found for locating and checking tasks, where accurate selection is important. However, none of the correlations observed were statistically significant, limiting any conclusions which can be drawn from this data.

<table>
<thead>
<tr>
<th>Pearson Product Moment Correlations</th>
<th>% of Total Time</th>
<th>Mean Pitch</th>
<th>Mean Roll</th>
<th>Variance Pitch</th>
<th>Variance Roll</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locating Task Times</td>
<td>28.41</td>
<td>0.13</td>
<td>-0.12</td>
<td>-0.01</td>
<td>-0.05</td>
</tr>
<tr>
<td>Navigating Task Times</td>
<td>51.40</td>
<td>-0.25</td>
<td>-0.23</td>
<td>-0.33</td>
<td>-0.31</td>
</tr>
<tr>
<td>Checking Task Times</td>
<td>20.19</td>
<td>-0.18</td>
<td>-0.13</td>
<td>-0.04</td>
<td>-0.04</td>
</tr>
<tr>
<td>Total Task Times</td>
<td>100.00</td>
<td>-0.28</td>
<td>-0.343</td>
<td>-0.26</td>
<td>-0.29</td>
</tr>
</tbody>
</table>

Table 4.12. Correlations between task times and pitch and roll angles

4.11 Discussion

Participants completed the set of tasks significantly faster using keypad interaction. This efficiency advantage was, however, limited to the tasks in which accurate cursor positioning and selection would have played a greater role (locating and checking tasks). Participants found controllability to be significantly better for keypad interaction. Observation of participants using tilt interaction showed that they would often find the target POI quickly, but would then move back and forth, attempting to perform an exact selection. During checking tasks, in which participants were required to select a POI icon by tapping anywhere on the screen (using tilt interaction), several participants accidentally panned away from the target icon as a result of tilting the device when reaching to touch the screen.

Tilt interaction performed noticeably better for navigation tasks than for locating and checking tasks. Participants were easily able to follow planned routes and the freedom of
movement offered by the tilt interaction was commented on favourably by several participants. While the efficiency advantage was not statistically significant, the subjective ratings provided in the questionnaire showed that tilt interaction was rated significantly better in terms of ease of use and controllability for navigation tasks. Tilt interaction was also rated higher in terms of effectiveness and efficiency for navigation tasks, but these results were narrowly not statistically significant ($p = 0.07$ for both effectiveness and efficiency).

The fact that tilt interaction performed better for navigation tasks can largely be attributed to the fact that these tasks did not include selection operations. As a result, the superior freedom of movement and speed control offered by tilt interaction resulted in a less frustrating user experience than that offered by keypad interaction.

Subjective workload was significantly lower for keypad interaction for several categories for the first set of tasks (locating set of tasks). It is interesting to note that this was not the case for the third set of tasks (checking tasks), which required a very similar set of low-level operations (checking tasks merely added a selection operation to the locating tasks). This indicates that participants initially struggled with the tilt interaction, rating it poorly in terms of mental demand and effort for locating tasks, but soon became more confident using this interaction technique. Mental demand was the one category in which tilt interaction scored worse than keypad interaction across all three task types. This difference was statistically significant for both locating and checking tasks, suggesting that participants felt that they needed to concentrate harder when using tilt interaction.

Keypad interaction outperformed tilt interaction in terms of total time taken to complete the assigned tasks. It is therefore interesting, that many users expressed the opinion that tilt interaction was more efficient than keypad interaction, both in the post-test ratings and in the qualitative comments section (Figure 4.15 and Table 4.8). Several possible reasons can be suggested. Firstly, tilt interaction was slightly more efficient than keypad interaction for navigation tasks. Navigation tasks took up a significant portion of the total evaluation time (across both interaction techniques), and therefore it is likely that performance during these tasks would have had a greater influence on participants’ perceptions of efficiency. Secondly, the continuous nature of tilt interaction as compared to the discrete nature of keypad interaction may have created the impression that tilt interaction was more efficient. Tilt interaction was also perceived to be less physically demanding than keypad interaction.
These factors could have contributed to participants regarding tilt as being faster than keypad interaction.

The fact that more participants preferred tilt interaction is noteworthy, given that keypad interaction performed better in terms of task times. If the positive qualities of tilt interaction identified by participants are considered (Table 4.8), it would suggest that those who preferred tilt interaction were influenced by perceived efficiency gains and greater freedom of movement. Those who preferred keypad interaction were influenced by the lack of precision offered by tilt interaction, as compared to the control and familiarity offered by keypad interaction. Several participants also commented that tilt interaction was intuitive to use. Several users also commented that using tilt interaction felt like “playing a game”.

It is interesting to note that of the 32 test participants, 20 had prior experience with tilt interaction (participants were not asked about prior experience before being recruited). Twelve of those with prior experience had also used multiple tilt-controlled devices. In most cases, this was a Nintendo Wii controller and mobile phone games. This is an interesting distinction between the evaluation conducted in this chapter and previous research and demonstrates how tilt interaction is rapidly becoming more pervasive. Those with prior experience with tilt interaction only performed slightly better than those without such experience. It is, however, noteworthy that those with prior experience were more likely to prefer tilt interaction over keypad interaction (Table 4.11). This provides evidence to suggest that prior experience made users more comfortable with tilt interaction.

Participants were divided about the use of SDAZ. Several participants commented favourably on the fact that it cut down on the need to constantly zoom in and out using the manual zooming controls. Automatic zooming was, however, also criticised by several participants as being annoying or disorienting. Observation of participants showed that while SDAZ was useful when participants wanted to pan over longer distances, zooming out was often accidentally triggered when attempting to quickly pan over a short distance. This would momentarily disorient users and hamper selection operations. It was clear that in some cases, automatic zooming contributed to the comparatively poor ratings tilt interaction received in terms of controllability and negatively affected participants’ confidence. As it is impossible to detect whether a user intends to pan over a long distance, perhaps it would be better for automatic zooming to be an optional setting, or for a clutch mechanism to be used to temporarily enable automatic zooming for longer panning actions.
A small minority of participants seemed to expect the opposite mapping between tilt direction and vertical panning. These participants struggled at first to get accustomed to moving in the opposite direction to which they instinctively expected to move. This problem has been frequently reported in previous research (Büring et al. 2008, Mooser et al. 2008).

### 4.12 Improving Tilt Interaction

The objective of the user study described in this chapter was to identify weaknesses of tilt interaction for mobile map-based applications and corresponding opportunities for improvement. In order to identify aspects of tilt interaction to be improved, it was necessary to analyse the usability issues identified in terms of the low-level operations (panning, zooming and selection) which were negatively affected. Table 4.13 summarises this information.

<table>
<thead>
<tr>
<th>Usability Issue</th>
<th>Reasons</th>
<th>Panning</th>
<th>Zooming</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllability</td>
<td>Nature of accelerometer input</td>
<td>Low</td>
<td>Low</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Lack of adequate feedback</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precise Selection</td>
<td>Nature of accelerometer input</td>
<td>N/A</td>
<td>N/A</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>Trigger mechanism</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zooming Technique</td>
<td>Lack of user control</td>
<td>N/A</td>
<td>Low</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Unpredictable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mental Demand</td>
<td>Unfamiliar interaction technique</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Continuous form of input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lack of feedback</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Practicality</td>
<td>Activating/deactivating</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>Controllability in different contexts</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Table 4.13. Usability issues regarding tilt interaction

The following scale was used to rate the severity of the usability issues identified, based on the results of the user study (Tullis and Albert 2008):

- **Low**: An issue which frustrates users but does not affect task success or failure.
- **Medium**: An issue which contributes to task failure, without actually causing task failure.
- **High**: An issue which causes task failure.
Each of the above usability issues is discussed in more detail in the following sections.

4.12.1 Controllability

Controllability is a well-documented weakness of tilt interaction (Section 3.6.5). In the user study of MapExplorer, this problem manifested itself in several ways. Several participants complained that they found the mapping between tilt gesture and the corresponding effect to be slightly unpredictable, particularly for controlling panning speed. One possible cause is that input from the accelerometer is inherently noisy for several reasons. These include hand tremor and the fact that users do not sit rigidly still while interacting with applications on their phone. Furthermore, because of changes in posture while interacting with MapExplorer, participants often struggled to rediscover the neutral orientation (when they last activated tilt interaction).

Another possible cause is the lack of feedback that was provided in MapExplorer. It is difficult for users to accurately predict the result of their actions if the application in question does not provide adequate feedback. Several of Bellotti et al.’s (2002) five questions regarding sensor-based interaction (Section 3.8) highlight the importance of letting users know that the system is doing (or has done) the right thing in response to their input. Feedback is important in order for users to control the extent of their actions. In the context of MapExplorer, this was evident in the manner in which some participants had difficulty controlling panning speed.

The implementation of SDAZ meant that the zoom level was directly linked to the current panning speed. As a result, the problem of controllability also affected zooming to some extent.

The relative ease with which participants were able to follow planned routes using tilt interaction suggests that the controllability problem did not extend to controlling the panning direction. The problem of controllability seemed largely limited to controlling panning speed. Participants seemed to struggle to predict the panning speed that would result from a particular tilt angle.

Several participants also pointed out that the sensitivity of the tilt interaction could be a significant problem were this form of interaction to be used in less controlled settings. These participants felt that it would be difficult to exercise sufficient control when on the move or distracted.
The severity of the controllability usability issue is listed as low (Table 4.1). This is because the controllability problems observed did not usually contribute to task failure, but rather (as demonstrated in the user study described in this chapter) led to increased task times. As a result, controllability problems generally led to user frustration as a result of decreased efficiency.

4.12.2 Precise Selection
Performing precise selection operations using tilt interaction proved to be a difficult challenge for several participants. Participants were often observed to move back and forth across a target icon attempting to settle on it before tapping the screen to perform a selection. In general, participants seemed to struggle to settle on a particular target without significant effort. This was one of the primary reasons that task times for locating and checking times were significantly longer for tilt interaction than for keypad interaction. This problem was less significant for navigating tasks, where accurate selections played a less significant role.

Problems performing selection are largely as a result of the nature of tilt interaction. While tilt interaction allows for freedom of movement and control over panning speed, controlling the precise position of the cursor can be difficult, particularly for inexperienced users.

A secondary problem with precise selection in MapExplorer is that participants often tilted the phone while reaching to tap the screen. This would result in the cursor moving off the target icon. The participant would then have to correct the cursor position and attempt the selection operation again.

Precise selection is listed as a medium severity usability issue in Table 4.1. This is because difficulties in performing accurate selections can contribute to user errors. Multiple attempts may be necessary for a user to correctly select an icon using tilt interaction.

4.12.3 Zooming Technique
Several participants commented that they felt that automatic zooming was distracting. Participants did not like the fact that SDAZ took control over the zoom level away from them. Similar results have been reported in research conducted prior and subsequent to this user study (Büring et al. 2008, Kratz et al. 2010). Some participants were observed to deliberately pan at slow speeds to avoid triggering SDAZ and maintain the current zoom level. Automatic zooming was also described as disorienting.
Chapter 4: Designing a Basic Tilt Interaction Technique

The zooming technique problem is listed as a low severity usability issue in Table 4.13. While unwanted automatic zooming can result in momentary disorientation and decreased efficiency, this usability issue does not generally affect whether or not users are able to successfully complete tasks.

### 4.12.4 Mental Demand

Participants consistently reported tilt interaction to be more mentally demanding than keypad interaction across all three sets of tasks (locating, navigating and checking). Several participants commented that they felt that intense concentration was required to complete the tasks correctly using tilt interaction. This problem seemed to affect all three low-level operations, with concentration required to complete panning, zooming and selection operations using tilt interaction.

This problem can partly be ascribed to the fact that the participants were largely unfamiliar with tilt interaction. Even though many participants had prior experience with tilt interaction, it remains a novel and unfamiliar interaction technique. Another problem with tilt interaction is that while it is active, it is “always on”. Such a continuous form of interaction means that users have to be careful not to accidentally trigger input merely by changing position or posture. This makes it difficult for users to focus on other tasks while using tilt interaction. Several participants commented that this made it difficult for them to relax when using the interaction technique and would likely detract from the feasibility of the interaction technique in real world use.

The lack of adequate feedback in MapExplorer was likely also a contributing factor to the high perceived mental demand. Very little feedback (other than panning and zooming taking place) is provided to help users know that the application is interpreting and executing their actions correctly.

The severity of the mental demand usability issue is listed as low in Table 4.13. The fact that tilt interaction requires high levels of concentration does not affect whether or not users are able to successfully complete tasks using this form of interaction.

### 4.12.5 Practicality

Several participants expressed concerns regarding the practicality of using tilt interaction in a mobile context of use. Participants felt that this form of interaction would be too sensitive to
their movements to be feasible for everyday use. Concerns were expressed that they would have to concentrate and exert control over their physical movements to avoid accidental or erroneous input.

Practicality is listed as a medium severity usability issue in Table 4.13. This is because difficulties in controlling tilt interaction while walking could be severe enough to contribute to user errors.

4.13 Discussion

The user study described in this chapter represents a more thorough comparison of keypad and tilt interaction than previous related studies (Dong et al. 2005, Schwarten et al. 2008, Büring et al. 2008). This user study considered a wider range of variables than those covered in previous research, with perceived workload, controllability and ease of use added to those employed in Dong et al.’s study. This experimental design allowed for a more detailed analysis of the shortcomings of tilt interaction for the domain of mobile map-based applications. Some of the results confirm previous findings, with problems regarding controllability and precise selection previously documented (Cho et al. 2007a, Hinckley and Song 2011, Crossan and Murray-Smith 2004). Shortcomings such as high mental demand and usability problems around tilt-controlled SDAZ are less well documented. The explicit identification of the usability shortcomings of tilt interaction provides a starting point for the design of enhancements to basic tilt interaction.

4.14 Conclusions

A basic tilt interaction technique was designed based on the best practices identified in the literature review of sensor-based interaction in Chapter 3. Accelerometer data was sampled asynchronously and a simple moving average smoothing algorithm was employed to remove signal noise. A rate control approach to performing panning was used as this mapping was identified in Chapter 3 as the most suitable for browsing large information spaces such as maps. Quadratic discretisation was employed in order to improve controllability. Zooming was implemented using SDAZ, as this technique has previously successfully been employed in conjunction with tilt interaction.

A prototype mobile map-based application, called MapExplorer, was developed in order to compare keypad and tilt interaction (both one-handed interaction techniques) for the domain of mobile map-based applications. Keypad interaction uses the traditional approach of
dedicated keypad keys for performing panning, zooming and selection. Tilt interaction supports combined zooming and panning, through the implementation of SDAZ. SDAZ links the zoom level to the current panning speed, with faster panning speeds resulting in automatic zooming out. Panning is performed through tilt gestures, with the user able to control the directionality and speed of panning operations. Selection in MapExplorer using tilt interaction is performed by tapping on the screen.

A user study was conducted with 32 participants, comparing keypad and tilt interaction in MapExplorer in terms of perceived workload, efficiency, effectiveness, controllability, ease of use and preference. Task times were also recorded. Participants were required to complete locating, navigating and checking tasks using both keypad and tilt interaction.

Post-task questionnaire results showed that keypad interaction performed significantly better than tilt interaction for locating tasks, particularly in terms of perceived workload. For navigating tasks this trend was reversed, with all post-task questionnaire ratings (except mental demand) being better for tilt interaction than for keypad interaction. For the last group of tasks (checking tasks), the results were mixed. While perceived workload was lower for keypad interaction, tilt interaction performed better in terms of perceived efficiency, effectiveness, controllability and ease of use.

Post-test questionnaire results showed that while participants perceived tilt interaction to be more efficient, a statistically significant difference in favour of keypad interaction was recorded in terms of controllability. Ease of use and effectiveness were also rated in favour of keypad interaction.

Qualitative comments showed that participants liked the controllability, precision and familiarity of keypad interaction. However, participants disliked the restrictive movement offered by keypad interaction, and perceived it to be less efficient than tilt interaction. Participants liked the efficiency and freedom of movement offered by tilt interaction. Opinions regarding SDAZ were strongly divided, with automatic zooming receiving several positive and negative comments. In general, participants felt that tilt interaction was comparatively difficult to control.

The results of the user study showed that keypad interaction was more efficient than tilt interaction for the two groups of tasks (locating and checking) that focused on selection operations. Tilt interaction was more efficient for navigation tasks, benefiting from greater
freedom of movement. Despite the fact that keypad interaction was more efficient than tilt interaction, the majority of participants preferred tilt interaction over keypad interaction.

The results of the user study revealed that tilt interaction provides several advantages over keypad interaction, particularly for tasks requiring freedom of movement. Several major shortcomings were also identified, namely controllability, performing accurate selection, zooming problems, high mental demand and practicality concerns.

In the next chapter, several techniques are proposed to address the shortcomings of tilt interaction identified in this chapter.
Chapter 5: Proposed Enhanced Tilt Interaction Techniques

5.1 Introduction

Tilt interaction was compared to keypad interaction in the previous chapter using a prototype mobile map-based application called MapExplorer. The results of this user study revealed several shortcomings of tilt interaction related to controllability, accurate selection, zooming, mental demand and practicality. In this chapter, several enhancements to basic tilt interaction are proposed to address these shortcomings (van Tonder and Wesson 2011b). A tilt interaction implementation incorporating these enhancements was compared to basic tilt interaction in order to determine whether the proposed techniques helped to address the shortcomings identified in the previous user study.

This chapter is organised into two main sections. The first section describes and motivates the design of the proposed enhanced tilt interaction techniques. The second section describes the user study that was conducted and presents an analysis of the results. Implications for the design of tilt interaction techniques for mobile map-based applications are also identified.

5.2 Proposed Techniques

The results of the previous user study highlighted five usability problems related to the use of tilt interaction in a mobile map-based application. Table 5.1 lists these problems and the corresponding techniques which were proposed to address each problem. The proposed techniques were built on top of the implementation of basic tilt interaction described in Chapter 4. The core functionality of mapping acceleration onto tilt angles and then onto panning speeds and the use of discretisation remains largely unchanged. In particular the following techniques are proposed:

- **Visual feedback**: Providing explicit visual feedback to users regarding the effects of their tilt input in terms of panning speed and direction. Existing implementations of tilt interaction make little or no use of visual feedback.

---

Chapter 5: Enhanced Tilt Interaction Techniques

- **Vibrotactile feedback**: Providing vibrotactile feedback when moving over icons to help users perform selection operations.
- **Attractors**: Aiding users in performing selection operations by drawing the cursor towards selectable icons when panning slowly. Similar techniques have been successfully applied for browsing photo collections using tilt.
- **Gesture zooming**: Allowing users to perform zooming using a discrete physical gesture.
- **Sensitivity adaptation**: Automatically reducing the sensitivity of tilt interaction when the user is mobile.
- **Dwell-time selection**: Supporting one-handed selection by allowing users to perform selections by placing the cursor over an icon for a pre-defined time period.

Each of the techniques listed in Table 5.1 are discussed in more detail in the sections that follow.

<table>
<thead>
<tr>
<th>Problem/Shortcoming</th>
<th>Proposed Technique(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllability</td>
<td>• Visual feedback&lt;br&gt;• Sensitivity adaptation</td>
</tr>
<tr>
<td>Precise Selection</td>
<td>• Attractor mechanisms&lt;br&gt;• Vibrotactile feedback&lt;br&gt;• Dwell-time selection</td>
</tr>
<tr>
<td>Zooming Technique</td>
<td>• Gesture zooming</td>
</tr>
<tr>
<td>Mental Demand</td>
<td>• Visual and vibrotactile feedback</td>
</tr>
<tr>
<td>Practicality</td>
<td>• Sensitivity adaptation</td>
</tr>
</tbody>
</table>

Table 5.1. Usability problems of tilt interaction and corresponding proposed techniques

5.2.1 **Visual Feedback**

Controllability is a well-known shortcoming of tilt interaction techniques (Section 3.6.5). Tilt interaction has been shown to suffer from overshooting problems because users misjudge the panning speed, particularly in rate control implementations (Crossan and Murray-Smith 2004). Jitter is also a problem as a result of noise in accelerometer data, hand tremor and walking motion. Even as tilt interaction techniques improve, users are still relative novices with this form of interaction and therefore take a while to get accustomed to tilt interaction.
The results of the previous experiment confirmed the findings of literature and showed that the participants struggled to control tilt interaction. Qualitative comments revealed that some participants found the exact effects of tilt gestures (in terms of panning speed) difficult to predict. In order to accurately control tilt interaction, participants need to be aware of the effect the current orientation of the mobile phone has on the direction and speed of panning.

Various modalities have previously been employed to aid in controlling tilt interaction, including visual, vibrotactile and audio feedback (Oakley and O’Modhrain 2005, Kratz et al. 2010, Eslambolchilar et al. 2004). Audio feedback was considered to be impractical because it is likely to be annoying to other people (and the use of headphones is not always feasible). Previous research has made only limited use of visual feedback to indicate cursor position and to control SDAZ (Kratz et al. 2010). Visual feedback in tilt interaction implementations has largely been limited to position controlled menu applications, where visual feedback is used to highlight the currently selected item (Rahman et al. 2009). Designers of tilt interaction techniques have not concentrated on the use of visual feedback because of the fact that users of mobile devices are generally partially distracted and may not always be looking at the screen. In many situations, however, users are able to give the display their full attention. In such situations, visual feedback can play an important role. Bellotti et al. (2002) identified five questions to be addressed in the design of sensor-based interaction techniques such as tilt interaction. Feedback can play an important part in addressing three of these questions, namely:

- How do I know the system is ready and attending to my actions?
- How do I effect a meaningful action, control its extent and possibly specify a target or targets for my action?
- How do I know the system is doing (has done) the right thing?

The use of two modalities to provide feedback to the user is proposed in this chapter. Visual feedback is proposed in conjunction with vibrotactile feedback. Visual feedback is used to provide more detailed feedback on panning speed and direction, and is complemented by the use of vibrotactile feedback, which allows the user to control selection operations even while partially distracted. These two forms of feedback therefore help to address the above three questions when using tilt interaction.
The proposed visual feedback technique makes use of arrows attached to the cursor, showing the vertical and horizontal panning speed (Figure 5.1). The length of these arrows is used to indicate the current panning speed (the longer the arrow the faster the panning speed). A simple linear mapping between panning speed and arrow length is used. The decision to show horizontal and vertical panning speeds separately was made due to the discretisation approach used in the original implementation of basic tilt interaction. Using this approach, horizontal and vertical panning speeds are calculated separately by using intervals to map pitch and roll angles onto panning speeds. Diagonal movement in any direction is still possible by combining different vertical and horizontal panning speeds. Using a single arrow would require horizontal and vertical panning speeds to be combined into a single arrow length, making interpretation more difficult. The use of a single arrow also resulted in more distracting visual feedback, because of rapid changes in the direction and length of the arrow because of sudden movements. The positioning of the arrows (attached to the cursor) allows users to view the visual feedback without having to take their eyes off the cursor.

![Figure 5.1. Visual feedback](image)

### 5.2.2 Vibrotactile Feedback

The use of vibrotactile feedback to reinforce the visual feedback channel is also proposed. This form of feedback is particularly useful in situations where the user may be partially distracted (such as while walking). Previous research regarding the use of vibrotactile feedback in conjunction with tilt interaction for menu navigation showed that users liked the use of short vibration pulses to indicate movement from one menu item to another (Oakley and O’Modhrain 2005). This form of feedback is used to indicate when the user moves over POI icons and route markers. Short vibration pulses (250 ms) are used to indicate when the cursor passes over such icons. This duration was selected as it has previously been employed...
for similar purposes and experimentation revealed it to be long enough to be noticeable, but short enough to not irritate users (Oakley and O’Modhrain 2005). Vibration feedback is only provided when panning slowly (and the user is likely to be performing selection operations) to avoid excessive feedback. The available map area is divided into grid cells based on latitude and longitude, with POI icons indexed into these cells. This enables the application to efficiently determine whether the cursor is close enough to a POI or route marker icon to trigger vibrotactile feedback.

The use of vibrotactile feedback as an indicator of panning speed was also considered. This was eventually discarded, as initial experimentation revealed that the panning speed changed very frequently. This resulted in a barrage of vibrotactile feedback. The use of vibrotactile feedback for two different purposes (selection and panning speed) led to users becoming confused regarding how to interpret this form of feedback.

5.2.3 Attractor Mechanisms

One of the most significant problems encountered during the use of tilt interaction in mobile map-based applications is that users experience difficulty settling the cursor on a particular target icon. Previous research has employed so-called attractor mechanisms to make it easier to perform precise selection operations. In a previous system using tilt interaction to browse photo collections on a mobile phone, attractors were employed to make it easier to settle on a particular photo (Cho et al. 2007b). This was done in order to overcome overshooting problems, where users would struggle to settle on a particular photo while browsing through their photo collection using tilt interaction. The idea of drawing the cursor towards a selectable target has previously been successfully applied in conjunction with touch-screen interaction (Roudaut et al. 2008).

This idea was extended to the two-dimensional domain of map browsing. Attractors are designed to work in conjunction with a discretisation approach, which splits tilt input into discrete speed levels. If both the current horizontal and vertical panning speeds are in the slowest two discretisation intervals (slower panning speeds usually precede a selection operation), the algorithm then determines whether any selectable icons are within a specified distance. The appropriate distance to be used as a threshold is largely application dependent. The threshold distance used needs to be balanced between too short a distance, which will limit the attractors’ usefulness, and too long a distance, which will result in unwanted attractions. Nearby icons are identified by making use of the same index used to implement
vibrotactile feedback. This enables the algorithm to run efficiently. If a nearby icon exists, the position of the cursor is then adjusted vertically and horizontally to draw the cursor towards the centre of this icon. The horizontal and vertical cursor position is adjusted one pixel at a time to create the effect of gradually drawing the cursor towards the icon. Figure 5.2 illustrates the use of attractors to draw the cursor (shown in blue) towards the nearest icon. If the cursor moves close enough to a POI icon (within the “Attractor Range” shown in Figure 5.2), the attractor mechanism is activated.

The resulting effect is one of drawing the cursor slowly towards selectable icons. The effects of the attractors are easily overcome by increasing panning speed, so as to avoid being drawn onto icons when this is not desired. The use of attractors eliminates the need to exactly position the cursor, as the user is able to approximately position the cursor and is assisted by the attractor to perform exact selection.

### 5.2.4 Sensitivity Adaptation

User performance when interacting with mobile applications is often negatively affected when the user is walking (Brewster 2002). Previous research has shown that users are significantly less efficient and less accurate when performing targeting and scrolling tasks using tilt interaction while walking as opposed to while seated (Crossan et al. 2008, Fitchett and Cockburn 2009). When walking, accelerometer signal disturbance as a result of the natural sway of the user’s gait can negatively impact cursor control (Crossan et al. 2009, Crossan et al. 2005). One of the problems identified in the previous experiment was that users felt that tilt interaction would be impractical in mobile contexts of use because of the need to control physical movements. When the user is mobile, there is significantly more
noise in the data received from the accelerometer, making it difficult to control this form of interaction. This makes tilt interaction difficult to design, as making it too insensitive makes it inefficient while seated, while making it too sensitive makes it difficult to use while walking.

Sensitivity adaptation was implemented in order to compensate for the increased variation in accelerometer data when walking. Streams of accelerometer data along the x, y and z axes are constantly recorded and monitored. Accelerometer data is sampled at 20 Hz and a sliding window of data from the last second (last 20 samples) is used to identify whether the user is currently stationary or mobile. Several data samples were recorded of users using MapExplorer while stationary and while walking, in order to gauge the baseline acceleration variances for the three axes. While performing tilt operations, increases in acceleration along one or more of the axes are typically recorded, so a single axis increase in acceleration is not sufficient to identify when the user is walking. Variances were calculated for acceleration along the three axes, as these provide a more accurate measurement of variability in the acceleration data than the raw values themselves. Analysis of the recorded data showed that while walking, all three acceleration variance values were typically greater than 0.125 (where acceleration is measured in g-forces). While seated, all three values almost never exceeded this threshold (Figure 5.3). This observation allows a computationally inexpensive method of identifying when the user is walking that can be performed quickly enough not to detract from system performance. Preliminary testing revealed this method to be extremely accurate at identifying when users are walking (> 98% accuracy in informal trials).

In order to compensate for increased variability when walking, the algorithm decreases the sensitivity of tilt interaction. This is done by increasing the size of the discretisation intervals used by the algorithm which converts tilt angles to panning speeds. The size of the “dead
zone”, where tilt interaction has no effect, is also increased when the user is walking. The algorithm therefore constantly observes the variation in the acceleration data in order to attempt to identify when to implement sensitivity adaptation. Figure 5.4 illustrates how sensitivity is decreased by increasing the size of the various discretisation intervals by 2 degrees. The size of the dead zone is increased by a larger amount (3 degrees either side of the neutral position) in order to ensure that unintentional panning does not occur as a result of the user’s walking motion. The initial threshold (5 degrees) proved insufficient for maintaining a steady cursor position when the user is mobile. These values were chosen after experimentation with increases of differing sizes. Larger increases result in greater control, but also detract from efficiency (very large tilt angles are required to achieve faster panning speeds) and also introduce problems regarding display visibility.

5.2.5 Gesture Zooming

Previous implementations of tilt interaction have frequently relied on SDAZ, where the zoom level is linked to the panning speed (Büring et al. 2008, van Tonder and Wesson 2010, Kratz et al. 2010, Eslambolchilar and Murray-Smith 2008a). This form of zooming, however, has been shown to be frustrating for users who feel that it takes control away from them. In a bid to find a form of zooming that would offer better usability and user satisfaction, a gesture-based zooming technique was proposed. Such an approach has previously been used in a camera-controlled mobile map-based application in which zooming out is performed by moving the device towards the user and zooming in by moving the device away (Mooser et al. 2008).
Gesture zooming relies on vertical movement of the mobile phone (note that gesture zooming is not related to the touch-screen GestureZoom technique (Patel et al. 2006)). Zooming out was implemented using an upward movement of the device (Figure 5.5), while zooming in was implemented using a downward movement of the device. These gestures are identified by constantly examining the z-axis acceleration data received from the accelerometer in order to identify spikes in the data. Subsequent to this design and the user study described in this chapter, related research showed these gestures to be perceived as the most intuitive for performing zooming in mobile applications (Ruiz et al. 2011).

![Figure 5.5. Zooming out using gesture zooming](image)

Acceleration data was captured and stored of users performing the vertical zooming gestures using a Nokia N97. Figure 5.6 shows an example of the z-axis acceleration data captured. The spikes in the data show where zooming gestures were performed. Zooming in and zooming out gestures can be differentiated from each other based on the order in which the lower and upper spikes in acceleration occur (Figure 5.7). Experimentation was conducted in order to identify suitable thresholds for the changes in acceleration used to identify zooming gestures. Setting the thresholds too low resulted in unintentional zooming occurring as a result of natural movement. Setting the thresholds too high makes the zooming gestures difficult to perform as the necessary movements are too dramatic. Experimental trials resulted in changes of 0.15g either side of the baseline z-acceleration (when the phone is at rest) being used to identify zooming in and zooming out gestures. While walking, this interval is increased to 0.3g to allow for the greater variability in z-axis acceleration data.

The baseline z-axis acceleration (indicated by the dotted line in Figure 5.6) value is dependent on the orientation of the phone. For example, if the phone is held parallel to the ground, a z-acceleration value of around 0.5g is reported (left side of Figure 5.6). If the phone
is held at a steeper angle, this value decreases (right side of Figure 5.6). In order to take the phone’s orientation into account, the baseline z-axis acceleration value is constantly updated to ensure that zoom gestures take this information into account. Failing to do so can result in misinterpretation of zooming gestures, as well as failure to identify zoom gestures. As a further measure to avoid accidental zooming, zooming gestures have to be completed within 300ms (normal movement is generally more gradual). This threshold was determined after experimentation to determine the most suitable interval.

![Z-Acceleration vs. Time](image)

**Figure 5.6. Z-axis acceleration data showing zooming gestures**

![Z-Acceleration Plot](image)

**Figure 5.7. Z-acceleration plot showing typical zooming out (left) and zooming in (right) z-acceleration patterns**

### 5.2.6 Dwell-Time Selection

During the previous experiment, it was observed that users would often accidentally pan away from a target icon while touching the screen to perform a selection operation.
Furthermore, the original design required both hands to perform selection operations. Dwell-time selection was proposed to address this problem. If the cursor is over an icon for longer than 2.75 seconds, the icon is selected without the user having to do anything. Vibration pulses of increasing durations (25ms, 50ms and 75ms) and a flashing cursor are used to indicate to the user that a selection operation is about to be performed. The duration of 2.75 seconds was selected after experimentation and was set relatively long as the timing starts when the cursor enters the boundary region of an icon, rather than when the cursor is directly over an icon. Shorter durations resulted in unintentional selections taking place. Dwell-time selection should be able to be deactivated when not desired. Previous research has employed dwell-time selection for similar purposes (Crossan and Murray-Smith 2004, Oakley et al. 2004).

5.2.7 Algorithm Design

The enhancements proposed above were integrated into the original algorithm for basic tilt interaction outlined in Figure 4.1. Figure 5.8 illustrates how the proposed techniques modify the original basic tilt interaction algorithm design. The stages of the original tilt interaction implementation are shown in blue, while the different proposed techniques are highlighted in green. Sensitivity adaptation is performed using the raw acceleration data retrieved from the accelerometer sensor. This information is used to analyse the variation in the accelerometer data in order to determine if sensitivity adaptation is necessary. If sensitivity adaptation is found to be necessary, the default discretisation intervals are modified. Accelerometer data is also used to detect zooming gestures. If zooming gestures are detected, the appropriate zooming operations are performed before the display is updated. The length and directionality of the visual feedback arrows are calculated after the panning speed and direction have been determined using discretisation. The implementation of attractors, vibrotactile feedback and dwell-time selection all require the current cursor position. Therefore, once the cursor position has been updated using the current panning speed and direction, the attractor algorithm determines whether any POI or route information icons are close enough to modify the panning speed or direction. If any icons are close enough to the current cursor position, vibrotactile feedback is triggered. If the cursor has been over the current icon for longer than a pre-defined interval, the icon is selected. All the proposed modifications to tilt interaction are therefore integrated into the original algorithm design used to implement basic tilt interaction.
Figure 5.8. Enhanced tilt interaction algorithm design

5.3 Implementation

The various proposed enhancements were incorporated into the MapExplorer prototype used in the previous user study. The following sections describe the implementation of each of the individual enhancements.

5.3.1 Visual and Vibrotactile Feedback

Visual feedback was proposed which makes use of arrows to indicate the direction and speed of panning that is currently taking place. The length of these vector-like arrows shows the horizontal and vertical panning speeds. Figure 5.9 shows an example of the use of this type of visual feedback in MapExplorer. The visual feedback arrows are attached to the cursor which
is fixed in position at the centre of the display. In the example shown, the user is currently panning diagonally down and to the right, with a faster vertical than horizontal panning speed (i.e. longer vertical arrow).

![Figure 5.9. Visual feedback in MapExplorer](image)

Vibrotactile feedback is provided to the user when moving within 25 pixels of a POI icon or route marker icon. A short vibration pulse (250 ms) is given by the phone’s built in vibration motor. Java ME, which was used as the implementation language, does not currently allow the intensity or frequency of vibration feedback to be controlled. The standard vibration intensity was therefore used.

### 5.3.2 Attractor Mechanisms

Attractor mechanisms were implemented for both POI and route marker icons in MapExplorer. A distance of 25 pixels was used as the threshold to identify nearby icons. This distance was selected as the cursor will begin to overlap icons within 25 pixels of its current position. This algorithm is executed as part of the main program loop, so the cursor will continue to be drawn to the nearest icon until it settles on the nearest one. Attractors are only activated if the user is panning slowly and within range of an icon. The simplicity of this algorithm ensures that the implementation of attractors in MapExplorer does not detract from the performance of the application. In order to avoid controllability difficulties and conflicting attractions as a result of densely packed icons, attractors are temporarily disabled when more than one icon could trigger attraction.

### 5.3.3 Sensitivity Adaptation

Sensitivity adaptation is performed as part of the main program loop. Accelerometer input is constantly monitored to determine whether there is a large degree of variability in the accelerometer data and consequently whether sensitivity adaptation is necessary. The appropriate discretisation mapping is used depending on whether the user is currently seated or walking. The user is not made aware that sensitivity adaptation is taking place.
5.3.4 Gesture Zooming

Gesture zooming was implemented in MapExplorer as to provide for discrete control over the current zoom level. Each zooming gesture increases or decreases the current zoom level by one discrete level. A zoom level indicator is displayed to indicate the new zoom level to the user. Gesture zooming is operated in an “always on” fashion in order to avoid the use of an activation mechanism and facilitate easy switching between panning and zooming operations.

5.3.5 Dwell-Time Selection

Dwell-time selection was implemented in MapExplorer to allow for one-handed selection. Leaving the cursor over an icon for more than 2.75 seconds (measured from the time the cursor moves within 25 pixels of the centre of the icon) results in the icon being selected. Dwell-time selection is an optional setting and can be deactivated when it is unnecessary by selecting a toggle icon on the screen.

A user study was conducted to compare an enhanced tilt interaction technique incorporating the proposed techniques to basic tilt interaction. The methodology used for the experiment and the results of the user study are presented in the following sections.

5.4 Experimental Design

A user study was conducted in order to compare enhanced tilt and basic tilt interaction. The user study was conducted in order to determine whether the proposed enhancements helped to alleviate any of the usability problems identified in the user study described in Chapter 4.

5.4.1 Objectives

The primary objective of the experiment was to determine whether the proposed enhancements helped to address the shortcomings of tilt interaction identified in Section 4.12. The following were therefore objectives of this user study:

- To determine whether the proposed enhancements offer any controllability improvements over basic tilt interaction;
- To determine whether the proposed enhancements help improve selection speed and accuracy;
- To determine whether gesture zooming is preferred over SDAZ;
- To determine whether the proposed enhancements help to reduce mental demand; and
To determine whether the proposed enhancements offer controllability improvements for walking tasks.

In order to achieve the above objectives, several hypotheses were evaluated. These are outlined in the next section.

5.4.2 Hypotheses

The following null hypotheses will be tested:

- $H_{0,1}$: Enhanced tilt does not exhibit better controllability than basic tilt interaction.
- $H_{0,2}$: Enhanced tilt does not exhibit better selection speed and accuracy than basic tilt interaction.
- $H_{0,3}$: Participants do not prefer gesture zooming to SDAZ.
- $H_{0,4}$: Enhanced tilt does not exhibit lower perceived mental demand than basic tilt interaction.
- $H_{0,5}$: Enhanced tilt does not exhibit better controllability for walking tasks than basic tilt interaction.

The proposed enhancements were evaluated collectively and compared to basic tilt interaction. While this approach inevitably means that it will be difficult to attribute causality to any one design feature, this approach was chosen for several reasons. Firstly, many of the design features (such as vibrotactile feedback and attractors) are designed to work together. Secondly, testing every single combination of design feature, task type and context of use would yield an infeasible experimental design. Finally, many of the design enhancements, such as gesture zooming and sensitivity adaptation, can be evaluated together, with little risk of overlapping influence on the results. Further investigation and experimentation will be conducted where necessary.

5.4.3 Metrics

Similar metrics were used to the previous user study. Participants were required to complete a post-task questionnaire after using each interaction technique. This questionnaire included the six perceived workload measures from the NASA-TLX questionnaire (mental demand, physical demand, temporal demand, performance, effort and frustration) (Hart and Staveland 1988). Six additional questions were also included to measure perceived efficiency, effectiveness, ease of use, controllability, ease of performing selections and ease of use while
walking. This portion of the questionnaire was based on the ASQ post-task questionnaire (Lewis 1991). The combined post-task questionnaire is provided in Appendix E. Seven point interval scales were used throughout.

After completing the task sets using both interaction techniques, participants were required to complete a post-test questionnaire. The post-test questionnaire required participants to rate their preferred interaction technique while seated and while walking and also to identify their preferred zooming technique. Participants were also asked to rate the usefulness of the visual and vibrotactile feedback and to identify positive and negative aspects of each interaction technique. The post-test questionnaire is provided in Appendix F.

Logging functionality was built into MapExplorer to record timing and user interaction data. The resultant log data was used to measure participants’ performance, including task times and route-following accuracy.

5.4.4 Participants

Sixteen participants (eleven male, five female) took part in the evaluation. Participants were all students in the Department of Computing Sciences at NMMU and were recruited via email. Nine participants had some prior experience with tilt interaction, but in most cases this was only limited use of mobile games or the Nintendo Wii. All participants were between the ages of 20 and 29. Four participants were left handed and most indicated occasional use of mobile map-based applications. The participant profile was therefore similar to the participants used in the first user study.

5.4.5 Tasks

Participants were required to complete three types of tasks with each interaction technique. The tasks selected were similar to those used in the first experiment and were designed to be representative of typical mobile map-based tasks. Fictional POIs were used to avoid any benefit being gained from local knowledge. The three task types were:

- **Locating tasks**: Participants were required to find a particular POI icon (e.g. Locate the Ritz Hotel).
- **Navigating tasks**: Participants were required to plan and follow a route from a particular start point to a particular end point (e.g. Plan a route from the beach to the airport. Follow the route from start to finish).
• **Checking tasks:** Participants were required to check specific information for a particular POI (e.g. check the opening hours of Starlight Restaurant).

### 5.4.6 Independent Variables

A within-subjects approach was used for all test conditions. The following independent variables were tested in this user study:

- **Interaction technique:** Two versions of tilt interaction were used, namely a basic tilt interaction technique and an enhanced tilt interaction technique incorporating the techniques proposed in this chapter. The order in which the techniques were used was counterbalanced across participants. The two versions were simply named “version 1” and “version 2” (depending on the order in which they were used) in all questionnaires and documentation to avoid any bias as a result of naming.

- **Task Type:** Locating, navigating and checking tasks were performed. Two similar task sets were used and counterbalancing was performed to off-set any order effects.

- **Context of use:** Seated and walking tasks were performed. Seated tasks were performed first in order to allow users to get accustomed to the techniques before performing walking tasks.

### 5.4.7 Test Environment

The user study was conducted in a laboratory environment in the Department of Computing Sciences at NMMU. Walking tasks were conducted indoors, with participants required to walk up and down a corridor.

### 5.4.8 Materials

A Nokia N97 with MapExplorer installed was used for the user study.

### 5.4.9 Procedure

Participants were required to complete the same informed consent form and biographical questionnaire as in the first user study (Appendices A and B). Participants were given an overview of each interaction technique (basic tilt and enhanced tilt) prior to use and encouraged to experiment with the techniques before commencing the tasks. Participants were required to complete two tasks of each task type while seated followed by two tasks of each type while walking. The first user study (Chapter 4) included three tasks of each type. In order to keep the user study to a reasonable duration, and because of the introduction of a
walking context of use, it was decided to use two tasks of each type in each context of use (a total of four tasks for each task type). Walking tasks were included in order to evaluate the two interaction techniques in a more natural context of use in which participants are distracted by their environment and in which there is likely to be more variation in the accelerometer data as a result of movement. This context of use is essential for evaluating hypothesis $H_{0.5}$. Participants were required to complete a post-task questionnaire (Appendix E) after using each interaction technique and a post-test questionnaire (Appendix F) after completing all tasks.

5.5 Results

The following sections discuss the performance, user satisfaction, perceived workload and qualitative results from the user study. The results are then analysed in Section 5.5.5. For ease of reference, the two configurations are referred to as basic tilt and enhanced tilt. Non-parametric Wilcoxon signed-rank tests were used to compare mean values for this user study because the relatively small sample size does not allow us to assume normally distributed data.

5.5.1 Performance Results

Table 5.2 shows the mean task times while seated and walking for the two interaction techniques. Enhanced tilt proved slightly more efficient for seated tasks, while basic tilt proved slightly more efficient for walking tasks. The mean total time for both walking and seated tasks was almost identical. None of the differences between the two techniques are statistically significant using Wilcoxon signed-rank tests and alpha values of 0.05.

<table>
<thead>
<tr>
<th></th>
<th>Basic Tilt</th>
<th>Enhanced Tilt</th>
<th>Z</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Seated</strong></td>
<td>275.73</td>
<td>259.81</td>
<td>1.14</td>
<td>0.255</td>
</tr>
<tr>
<td><strong>Walking</strong></td>
<td>267.74</td>
<td>284.90</td>
<td>0.72</td>
<td>0.469</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>543.47</td>
<td>544.71</td>
<td>0.31</td>
<td>0.756</td>
</tr>
</tbody>
</table>

Table 5.2. Wilcoxon signed-rank test results comparing mean task times (in seconds) for seated and walking tasks ($n = 16$)

Mean task times for different task types are shown in Table 5.3. Enhanced tilt achieved marginally better task times for locating and checking tasks, while basic tilt was slightly more efficient for navigating tasks. None of these differences are statistically significant using
Wilcoxon signed-rank tests and alpha values of 0.05. No significant differences were recorded for the various combinations of task type and context of use.

<table>
<thead>
<tr>
<th></th>
<th>Basic Tilt</th>
<th>Enhanced Tilt</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>124.38</td>
<td>116.47</td>
<td>0.52</td>
</tr>
<tr>
<td><strong>SD</strong></td>
<td>20.15</td>
<td>16.92</td>
<td>0.605</td>
</tr>
<tr>
<td><strong>Locating</strong></td>
<td>314.16</td>
<td>328.56</td>
<td>0.52</td>
</tr>
<tr>
<td><strong>Navigating</strong></td>
<td>104.93</td>
<td>99.68</td>
<td>0.57</td>
</tr>
<tr>
<td><strong>Checking</strong></td>
<td>19.85</td>
<td>19.26</td>
<td>0.569</td>
</tr>
</tbody>
</table>

Table 5.3. Wilcoxon signed-rank test results comparing mean task times (in seconds) for different task types (n = 16)

Table 5.4 shows the mean accuracy results for navigating tasks for each technique. Accuracy was measured as the mean distance (in pixels) between the planned route and the user’s actual path (as determined from log data). Enhanced tilt was more accurate for both seated and walking tasks. A statistically significant accuracy improvement was achieved for walking tasks (p = 0.038).

<table>
<thead>
<tr>
<th></th>
<th>Basic Tilt</th>
<th>Enhanced Tilt</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean</strong></td>
<td>7.28</td>
<td>6.06</td>
<td>1.08</td>
</tr>
<tr>
<td><strong>SD</strong></td>
<td>4.08</td>
<td>2.17</td>
<td>0.279</td>
</tr>
<tr>
<td><strong>Seated</strong></td>
<td>13.94</td>
<td>10.34</td>
<td>2.07</td>
</tr>
<tr>
<td><strong>Walking</strong></td>
<td>6.52</td>
<td>3.78</td>
<td><strong>0.038</strong></td>
</tr>
</tbody>
</table>

Table 5.4. Wilcoxon signed-rank test results comparing mean accuracy for navigating tasks (measured as mean deviation from planned route in pixels) (n = 16)

Figure 5.10 shows plots for participants performing a route-following task while walking (created from log data). All participants who performed the task in question with no sensitivity adaptation (left) are shown separately from those who performed the task with sensitivity adaptation (right). Due to counterbalancing, half the participants used each setting for the task in question. The plots show some indication that sensitivity adaptation allowed greater control while walking. It can also be seen on the left side of Figure 5.10 how participants struggled to select the starting point (the green icon) accurately without the help of attractors (note that the plots only show data from once the route had been successfully retrieved from the routing web service). In contrast, all participants were able to perfectly select the green icon with the help of attractors.
Figure 5.10. Route plots for participants performing route-following without sensitivity adaptation (left) and with sensitivity adaptation (right). The actual route is shown in blue.

Participants had the option to use either dwell-time selection or touch-screen selection when using the enhanced tilt interaction technique. Table 5.5 shows that more participants used dwell-time selection for both seated and walking tasks, although this difference was not statistically significant. Participants were also provided with an option to disable dwell-time selection. Only one participant made use of this option.

<table>
<thead>
<tr>
<th>Selection Method</th>
<th>Seated</th>
<th>Walking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dwell-time selection</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>Touch selection</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Equal use</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.5. Number of participants employing each selection method employed for checking tasks (n = 16)

Table 5.6 shows the mean distance in pixels between the cursor location and the centre of the target POI icon when performing selection operations for each technique. Using enhanced tilt, participants were able to almost perfectly position the cursor in all cases. For basic tilt, the cursor was on average 3.91 pixels from the centre of the target for seated tasks and 5.72 pixels from the centre of the target icon for walking tasks. The differences between basic tilt and enhanced tilt are statistically significant for both seated (p < 0.001) and walking tasks (p < 0.001).
Table 5.6. Wilcoxon results comparing mean distances (in pixels) from POI icons when performing selection operations ($n = 16$)

<table>
<thead>
<tr>
<th></th>
<th>Basic Tilt</th>
<th>Enhanced Tilt</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
</tr>
<tr>
<td>Seated</td>
<td>3.91</td>
<td>3.22</td>
<td>0.69</td>
</tr>
<tr>
<td>Walking</td>
<td>5.72</td>
<td>3.75</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Figure 5.11 illustrates this trend graphically. The data for all participants for the same task is shown overlaid on the map. Due to counter-balancing, half the participants performed the task using basic tilt (left) and half using enhanced tilt (right). As can be seen, participants were able to approach the target POI icon more directly using enhanced tilt. The basic tilt data exhibits more erratic movement and evidence of overshooting. The task in question was performed while walking, but a similar trend was observed for seated tasks.

Overshooting errors (also referred to as slip-off errors) have previously been used to measure selection accuracy (Xin et al. 2011, Crossan and Murray-Smith 2004). An overshooting error is illustrated in Figure 5.12 and occurs when the user overshoots when attempting to select an icon and has to reverse panning direction in order to perform a selection. The number of times a user enters the region of a target icon when performing selection has previously been used to measure overshooting errors. This metric is referred to as the number of crossings (Xin et al. 2011). For checking tasks (which required selection operations), significantly fewer crossings were observed for enhanced tilt for both seated and walking tasks (Table
5.7). Wilcoxon signed-rank tests showed this difference was statistically significant for both seated (p = 0.013) and walking (p = 0.002) checking tasks.

The mean time between entering the POI boundary region (where attractors and vibrotactile feedback are activated) and settling on the target icon was also calculated. Selection time was not included because the time delay involved in dwell-time selection would make the results difficult to interpret. The results (Table 5.8) show that participants were able to settle on the target icon quicker for both seated and walking tasks using the enhanced tilt technique. The differences were statistically significant for seated (p = 0.004) and walking tasks (p = 0.043). The overall difference was also significant (p = 0.017).
5.5.2 Perceived Workload Results

Figure 5.13 shows the mean perceived workload ratings for the two interaction techniques. Perceived workload was lower (better) in all categories for enhanced tilt. These differences were significant for mental demand (Wilcoxon Z = 2.78, p = 0.005), temporal demand (Wilcoxon Z = 2.20, p = 0.028) and effort (Wilcoxon Z = 2.55, p = 0.011).

![Mean Perceived Workload Ratings](image)

Figure 5.13. Mean perceived workload ratings (95% confidence intervals shown) (n = 16)

5.5.3 User Satisfaction Results

Figure 5.14 shows the mean user satisfaction ratings for the two interaction techniques used in the evaluation. Participants were asked to rate the two techniques on a seven point interval scale in terms of effectiveness, efficiency, controllability, ease of use, ease of selection and ease of use while walking. The results show that the enhanced tilt received higher user satisfaction ratings for all six satisfaction questions. These differences were statistically significant for effectiveness (Wilcoxon Z = 2.19, p = 0.028), efficiency (Wilcoxon Z = 2.71, p = 0.007), controllability (Wilcoxon Z = 2.17, p = 0.030), ease of selection (Wilcoxon Z = 2.82, p = 0.005) and ease of use while walking (Wilcoxon Z = 2.13, p = 0.033).
Figure 5.14. Mean user satisfaction interval scale ratings (95% confidence intervals shown) ($n = 16$)

Table 5.9 shows the participant preference ratings from the post-test questionnaire. Seven point interval scales were used. An intermediate value of four indicated a neutral response. In Table 5.9, a value of 1 indicates a preference for basic tilt and 7 indicates a preference for enhanced tilt. Participants preferred enhanced tilt while seated and while walking. The preference for enhanced tilt was stronger when participants were walking (mean of 5.88 when walking as opposed to 4.94 while seated). The difference between the mean values and the neutral value of four was statistically significant for both seated ($p = 0.041$) and walking ($p = 0.002$) tasks. Gesture zooming was preferred over SDAZ, with a mean value of 4.44. The difference between this mean value and the neutral value of four was not, however, statistically significant.

<table>
<thead>
<tr>
<th>Preferences</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td>Preferred Technique (Seated)</td>
<td>4.94</td>
</tr>
<tr>
<td>Preferred Technique (Walking)</td>
<td>5.88</td>
</tr>
<tr>
<td>Preferred Zooming Technique</td>
<td>4.44</td>
</tr>
</tbody>
</table>

Table 5.9. Wilcoxon results comparing participant post-test preference ratings to intermediate value of 4 ($1 =$Basic tilt, 7 = Enhanced tilt) ($n = 16$).

Participants were also asked to indicate whether they felt vibration feedback was useful for performing selections and whether visual feedback was useful for controlling panning speed.
and direction. Seven point interval scales were used (1 = Strongly Disagree and 7 = Strongly Agree). The high mean values obtained indicate that participants found these forms of feedback to be very useful (Table 5.10). The difference between the mean and the neutral value of four was statistically significant for vibration feedback ($p < 0.001$) and for visual feedback ($p < 0.001$).

<table>
<thead>
<tr>
<th>Ratings</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vibration feedback</td>
<td>6.25 0.68 3.52 &lt;0.001</td>
</tr>
<tr>
<td>Visual feedback</td>
<td>6.00 0.63 3.52 &lt;0.001</td>
</tr>
</tbody>
</table>

Table 5.10. Wilcoxon results comparing feedback usefulness ratings to intermediate value of 4 (1=Strongly Disagree, 7 = Strongly Agree) ($n = 16$)

### 5.5.4 Qualitative Feedback Results

Opinions regarding SDAZ and gesture zooming were sharply divided, with 9 participants preferring gesture zooming and 7 preferring SDAZ. Participants who indicated a preference for SDAZ identified two common reasons. Firstly, several participants felt that SDAZ enabled them to effectively and efficiently locate distant POI icons. These participants felt that SDAZ helped to provide a greater sense of perspective when panning long distances. Secondly, several participants who indicated a preference for SDAZ did so because of negative experiences with gesture zooming. These participants generally indicated that they felt that it would take a while for them to grow accustomed to using gesture zooming and some complained of gesture recognition errors. Participants who indicated a preference for gesture zooming identified the intuitiveness and controllability of this zooming technique as the reasons for their preferences. Several participants also indicated that they felt that SDAZ was confusing and difficult to control while walking and as a result, they preferred gesture zooming. They reported that SDAZ was frequently triggered by accident while walking and that this contributed to a perceived lack of controllability.

The frequencies of the three most frequently identified positive and negative aspects of the basic tilt interaction technique are presented in Table 5.11. Positive feedback regarding basic tilt interaction largely focused on the SDAZ zooming technique. Several participants commented that they felt the use of SDAZ allowed them to be more efficient when searching for points of interest, particularly where long panning distances were involved. Some
participants also liked the relative simplicity of this technique and the fact that it allowed them to interact with the map without having to worry about controlling the zoom level.

Negative feedback about basic tilt also focused on SDAZ. While feedback regarding the use of SDAZ while seated was mixed, most participants strongly disliked SDAZ while walking. Participants found that SDAZ detracted from controllability when walking and was too easy to trigger accidentally. Participants also complained that SDAZ was not wanted when following a route and resulted in disorientation. Participants felt that the basic tilt interaction technique did not offer the same level of controllability as enhanced tilt.

<table>
<thead>
<tr>
<th>Category</th>
<th>n</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Issues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDAZ</td>
<td>7</td>
<td>“Automatic zooming makes locating landmarks easier”</td>
</tr>
<tr>
<td>Simplicity</td>
<td>2</td>
<td>“Simpler than the other technique”</td>
</tr>
<tr>
<td>Smooth panning</td>
<td>1</td>
<td>“Allows for smooth movements”</td>
</tr>
<tr>
<td>Negative Issues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDAZ</td>
<td>7</td>
<td>“Automatic zooming is generally awkward”</td>
</tr>
<tr>
<td>Controllability</td>
<td>4</td>
<td>“Lack of control”</td>
</tr>
<tr>
<td>Lack of feedback</td>
<td>3</td>
<td>“No feedback on [panning] direction and status”</td>
</tr>
</tbody>
</table>

Table 5.11. Frequencies of positive and negative issues for basic tilt interaction

The frequencies of the three most frequently identified positive and negative aspects of enhanced tilt are presented in Table 5.12. Participants provided strongly positive feedback regarding the use of attractor mechanisms, visual and vibrotactile feedback, as well as the controllability of the enhanced tilt technique. Participants felt that the use of attractors made selection easier and more accurate. Participants described vibrotactile feedback as useful and stated that the direction and speed feedback provided by the use of arrows around the cursor helped to foster a greater sense of control. Several participants commented that enhanced tilt offered improved controllability as compared to basic tilt interaction. Gesture zooming was also rated highly for supporting one-handed control and providing explicit control over the current zoom level.

Negative feedback regarding enhanced tilt mainly focused on problems with gesture zooming. Learnability emerged as a major shortcoming of this interaction technique, with several participants remarking that they felt that they might have liked gesture zooming more
if they had more experience with it. Some participants also found the zooming gestures uncomfortable to perform. One participant stated that they felt the attractor mechanisms used were too strong.

<table>
<thead>
<tr>
<th>Category</th>
<th>n</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Issues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attractors</td>
<td>6</td>
<td>“Easier to select items (by drawing you to them automatically)”</td>
</tr>
<tr>
<td>Visual and vibrotactile feedback</td>
<td>6</td>
<td>“Vibrations make selection and navigation easy”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>“Direction and speed feedback useful”</td>
</tr>
<tr>
<td>Controllability</td>
<td>4</td>
<td>“Easier to scroll to and select points of interest”</td>
</tr>
<tr>
<td>Negative Issues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zooming</td>
<td>6</td>
<td>“Zooming takes a while to get used to”</td>
</tr>
<tr>
<td>Dwell-time selection</td>
<td>2</td>
<td>“Automatic selection of icons”</td>
</tr>
<tr>
<td>Attractors</td>
<td>1</td>
<td>“Sometimes the attraction to a POI is too strong”</td>
</tr>
</tbody>
</table>

Table 5.12. Frequencies of positive and negative issues for enhanced tilt interaction

5.5.5 Discussion

Efficiency results were very similar for both interaction techniques. Similar task times were recorded for both the seated and walking contexts of use and no significant efficiency differences were found for any of the three task types (Tables 5.2 and 5.3). Despite the fact that no significant efficiency differences were found, participants perceived enhanced tilt to be more efficient than basic tilt interaction.

Positive results were achieved with regards to the controllability of the enhanced tilt interaction technique while walking. Analysis of log data showed participants to be significantly more accurate for route-following tasks performed while walking (Table 5.4). Participants were also able to perform selection operations significantly more accurately (Table 5.6). Participants’ perceptions supported the objective results, with participants reporting a statistically significant increase in perceived ease of use for walking tasks (Figure 5.14). These results are particularly encouraging for the use of sensitivity adaptation and (in the case of selection accuracy) the use of attractors to address concerns regarding the practicality of tilt interaction in a mobile context of use.
Chapter 5: Enhanced Tilt Interaction Techniques

The evaluation results clearly show that enhanced tilt improved the accuracy and efficiency with which participants were able to perform selection operations. Analysis of log data for checking tasks showed that participants were able to select POI icons more accurately (Table 5.6). Participants were also able to settle the cursor over POI icons more quickly (Table 5.8). Furthermore, significantly fewer overshooting errors took place using the enhanced tilt interaction technique (Table 5.7). There is also some indication that the selection of POI icons as start and end points in navigation tasks was made easier through the use of attractors (Figure 5.10). The accuracy improvements were more significant for walking tasks, indicating that attractors can play an important role in a mobile context of use. Participants’ perceptions confirm the objective results that were achieved. Participants found accurate selection significantly easier using enhanced tilt (Figure 5.14). These results are noteworthy as precise selection was identified as one of the shortcomings of tilt interaction (Table 5.1). Attractors, vibrotactile feedback and dwell-time selection were all designed to address this shortcoming. Given that attractors were the (tied) most frequently reported positive aspect of the enhanced tilt interaction technique (Table 5.12), much of the improvement achieved can be attributed to the use of attractors.

Opinions regarding the preferred zooming technique were divided. While some users quickly mastered the gesture zooming technique, others found the learnability of this technique to be problematic. Gesture zooming appeared to suffer from occasional gesture recognition errors and further investigation is required in this regard. SDAZ also resulted in divided opinions. SDAZ was identified as being useful for locating POIs, but difficult to use when following a route. Participants also found SDAZ to be difficult to use when walking. Preference ratings showed that participants were divided regarding their preferred zooming technique, with slightly more participants in favour of gesture zooming.

Enhanced tilt provided statistically significant improvements in three aspects of perceived workload, namely mental demand, temporal demand and effort (Figure 5.13). The improvement in mental demand is particularly encouraging, as this is one of the areas in which tilt interaction was previously identified as being inferior to keypad interaction. While visual and vibrotactile feedback were designed to address this shortcoming, it is likely that the controllability and selection accuracy improvements achieved through the use of sensitivity adaptation and attractors also helped to put participants more at ease.
Feedback regarding the use of visual and vibrotactile feedback was strongly positive. Participants rated both forms of feedback as very useful (Table 5.10). Qualitative comments regarding these forms of feedback was also very positive, with only one participant remarking that vibrotactile feedback could become annoying after prolonged use (Table 5.12).

Enhanced tilt was preferred both while seated and while walking (Table 5.9). This preference was stronger while walking. Several possible reasons for this exist. Firstly, enhanced tilt included sensitivity adaptation which makes allowance for the greater variability in accelerometer data while walking. Secondly, it is likely that the attractor mechanisms and feedback play a greater role in improving controllability while the user is mobile. Performance data showed that enhanced tilt achieved greater improvements in route-following and selection accuracy for walking tasks as compared to seated tasks (Tables 5.4, 5.6 and 5.7). Finally, SDAZ in the basic tilt technique was perceived to be difficult to use while walking.

5.6 Hypothesis Testing

In Section 5.4.2, various hypotheses were presented. In this section, these hypotheses are revisited in order to determine whether the results of the user study provide sufficient evidence to reject any of the null hypotheses (Table 5.13).

$H_{0,1}$: Enhanced tilt does not exhibit better controllability than basic tilt interaction.

The results of the user study show that participants perceived enhanced tilt to be easier to control than basic tilt interaction (Figure 5.14). Controllability improvements were also cited as one of the most positive aspects of enhanced tilt (Table 5.12). Participants were also able to follow routes more accurately while walking (Table 5.4) and perform selection operations more accurately (Tables 5.6 and 5.7). There is therefore sufficient evidence to reject the null hypothesis $H_{0,1}$.

$H_{0,2}$: Enhanced tilt does not exhibit better selection speed and accuracy than basic tilt interaction.

Enhanced tilt was shown to provide for more accurate and efficient selection (Tables 5.6 and 5.8). Furthermore, attractors helped to reduce overshooting errors in both seated and walking contexts of use (Table 5.7). Attractors were also the most frequently cited positive aspect of
enhanced tilt (Table 5.12). There is therefore sufficient evidence to reject the null hypothesis $H_{0,2}$.

$H_{0,3}$: Participants do not prefer gesture zooming to SDAZ.

Participants indicated a slight preference for gesture zooming (Table 5.9). However, many participants were observed to struggle with this form of zooming and gesture recognition problems were encountered. There is therefore insufficient evidence to reject the null hypothesis $H_{0,3}$.

$H_{0,4}$: Enhanced tilt does not exhibit lower perceived mental demand than basic tilt interaction.

A statistically significant improvement in perceived mental demand was reported for enhanced tilt as compared to basic tilt interaction (Figure 5.13). There is therefore sufficient evidence to reject the null hypothesis $H_{0,4}$.

$H_{0,5}$: Enhanced tilt does not exhibit better controllability for walking tasks than basic tilt interaction.

A statistically significant accuracy improvement was recorded for route-following tasks while walking (Table 5.4). Furthermore, participants were able to perform more accurate selections while walking (Table 5.6). Participants also perceived enhanced tilt to be significantly easier to use while walking (Figure 5.14). There is therefore sufficient evidence to reject the null hypothesis $H_{0,5}$.

<table>
<thead>
<tr>
<th>#</th>
<th>Null Hypothesis</th>
<th>Reject?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{0,1}$</td>
<td>Enhanced tilt does not exhibit better controllability than basic tilt interaction</td>
<td>✓</td>
</tr>
<tr>
<td>$H_{0,2}$</td>
<td>Enhanced tilt does not exhibit better selection speed and accuracy than basic tilt interaction</td>
<td>✓</td>
</tr>
<tr>
<td>$H_{0,3}$</td>
<td>Participants do not prefer gesture zooming to SDAZ</td>
<td>✗</td>
</tr>
<tr>
<td>$H_{0,4}$</td>
<td>Enhanced tilt does not exhibit lower perceived mental demand than basic tilt interaction</td>
<td>✓</td>
</tr>
<tr>
<td>$H_{0,5}$</td>
<td>Enhanced tilt does not exhibit better controllability for walking tasks than basic tilt interaction</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 5.13. Summary of hypothesis testing results
5.7 Design Implications

The results of the user study provide insight into the design of tilt interaction. In this section, implications for the design of tilt interaction techniques are presented in the form of design recommendations based on the results of this user study.

R1. Use attractor mechanisms to allow for selection using approximate cursor positioning.

The implementation of attractor mechanisms proved very successful. Participants in the user study felt that attractors helped to improve selection accuracy and qualitative feedback was strongly positive. Observation of participants using MapExplorer showed that they often relied on attractors by approximately positioning the cursor and allowing the system to perform precise selection. Analysis of log data showed that attractors allowed for faster and more accurate selection (Tables 5.6 - 5.8).

R2. Reduce tilt interaction sensitivity when the user is mobile.

Sensitivity adaptation was implemented to adapt the sensitivity of tilt interaction according to the variance in the accelerometer input. This was done in order to make tilt interaction easier to control when the user is mobile and there is more variation in accelerometer data as a result of user movement. The results of the user study showed that participants found enhanced tilt incorporating sensitivity adaptation significantly easier to use while walking (Figure 5.14). Analysis of log data showed that a statistically significant improvement in accuracy was achieved for route-following tasks performed while walking (Table 5.4). Several positive comments were also received regarding the controllability of enhanced tilt while walking.

The implementation of basic tilt interaction did not provide for sensitivity adaptation. Several negative comments were received regarding the ease of use of basic tilt interaction while walking. Noticeable behavioural changes were evident when participants used basic tilt interaction while walking. Participants frequently had to slow down or stop in order to gain better control over interaction. This trend was less noticeable for enhanced tilt. Some participants felt comfortable enough to walk with one hand in their pocket while using this interaction technique, a particularly encouraging sign regarding the ease of use of this interaction technique while walking.
The results of the user study demonstrate that sensitivity adaptation provides usability benefits for tilt interaction, particularly in terms of controllability while walking. Sensitivity adaptation could be enhanced by incorporating a more flexible mapping between accelerometer data variance and the sensitivity of tilt interaction.

**R3. Use visual and vibrotactile feedback to improve perceived controllability and reduce mental demand.**

The use of visual and vibrotactile feedback in MapExplorer was well received by participants in the user study. The results of the comparative user study showed that enhanced tilt was perceived as significantly less mentally demanding than basic tilt interaction. Visual and vibrotactile feedback were both included in enhanced tilt to address this shortcoming of tilt interaction (Table 5.1). Qualitative feedback regarding visual and vibrotactile feedback was very positive. Both forms of feedback were rated as useful by participants in the user study (Table 5.10). Participants specifically remarked that visual feedback provided a greater sense of control.

On the evidence of this experiment, the use of simple visual and vibrotactile feedback both contributed positively to the usability of tilt interaction. These forms of feedback could even be extended to provide additional feedback to the user. Where the frequency of tilt interaction can be controlled, this could be used to convey the distance to selectable icons. Visual feedback could be extended to apply to zooming operations.

### 5.8 Discussion

The techniques proposed in this chapter extend existing research into the design of tilt interaction techniques. Previous research has made only limited use of visual feedback and has left users to interpret the results of their tilt input. Attractors have not previously been used in conjunction with tilt interaction to support selection operations in a two-dimensional application such as mobile map browsing. Furthermore, the design described in this chapter allows for efficient operation by indexing icons based on geographic location. The use of physical gestures to perform zooming operations in conjunction with tilt-controlled panning is novel and allows for both zooming and panning to be explicitly controlled using one-handed interaction. Sensitivity adaptation is a novel technique and is one of the first techniques specifically designed to improve the usability of tilt interaction in a mobile context of use. The results of the user study show that many of the above techniques
produced significant benefits for the usability of tilt interaction, with the use of attractors and sensitivity adaptation proving particularly successful.

The inclusion of a walking condition in the user study provided interesting and novel results. SDAZ was found to be particularly difficult to control in conjunction with tilt interaction when the user is mobile. Sensitivity adaptation was shown to improve the level of control users have over tilt interaction while mobile, an area of research that has received only limited attention in the past (Crossan et al. 2005).

5.9 Conclusions

This chapter proposed several enhancements to basic tilt interaction designed to address the shortcomings of basic tilt interaction identified during the previous user study (Chapter 4). Visual and vibrotactile feedback were designed to improve the controllability of tilt interaction and also to reduce the high mental demand which was reported during the previous user study comparing tilt interaction with keypad interaction. Attractor mechanisms were introduced to make precise selection easier by automatically drawing the cursor towards POI and route instruction icons. Sensitivity adaptation was introduced to improve controllability while walking, one of the issues that was previously identified as limiting the practicality of tilt interaction in mobile contexts. Gesture zooming was designed as a more intuitive zooming technique, allowing the user full control over the zoom level using physical gestures. Dwell-time selection was introduced as an optional setting which allows users to perform one-handed selection operations. These techniques were implemented in a prototype mobile map-based application, MapExplorer.

An enhanced tilt interaction technique incorporating the above techniques was compared to the basic tilt interaction technique which was used in the previous user study. The user study required participants to use both interaction techniques and incorporated walking tasks to evaluate the comparative usability of the two interaction techniques in a mobile context of use. Enhanced tilt was shown to provide several significant objective and subjective benefits over basic tilt interaction.

Objective benefits included improved accuracy for route-following tasks while walking, improved selection accuracy and improved selection efficiency. Subjective benefits included reduced perceived workload (including mental demand), improved perceived controllability and improved perceived ease of use for walking tasks.
Given the shortcomings identified in the previous user study, several of these improvements are of particular significance. These include a significant improvement in controllability for both selection and route-following tasks and significant improvements in perceived mental demand, controllability and ease of use. Participants also found enhanced tilt to be significantly easier to use than basic tilt interaction. A statistically significant preference was reported for enhanced tilt over basic tilt interaction while seated and while walking. Results for gesture zooming were more equivocal, with participants being divided between the two zooming techniques. These results provide empirical evidence that the design modifications incorporated in enhanced tilt helped to mitigate several of the usability shortcomings of basic tilt interaction.

The results of the user study were used to motivate the proposal of three recommendations for the design of tilt interaction techniques. Future tilt interaction implementations could benefit from the use of sensitivity adaptation, attractors and visual and vibrotactile feedback. Several possibilities also exist for further improving these techniques.

The user study results provide empirical evidence that the proposed enhancements helped to address several of the shortcomings of tilt interaction identified in Chapter 4. Improvements in terms of controllability, precise selection, mental demand and practicality (controllability while walking) were achieved. Gesture zooming, however, did not achieve a significant improvement over SDAZ.

The use of sensor fusion has previously been suggested as a means of addressing some of the shortcomings of tilt interaction (Crossan et al. 2008). Subsequent to the user study described in this chapter, several mobile phones were released which incorporate accelerometers, digital compasses and gyroscope sensors. This enables the development of tilt interaction techniques which take advantage of sensor fusion. In Chapter 6, the use of sensor fusion as a means of addressing some of the shortcomings of tilt interaction will be investigated.
Chapter 6: Using Sensor Fusion to Improve Tilt Interaction

6.1 Introduction

In the previous chapter, several enhanced techniques were proposed to address the shortcomings of tilt interaction. One of the most significant shortcomings of tilt interaction is controllability. Several measures were proposed to reduce controllability problems, including the use of attractors to aid in performing selection, visual and vibrotactile feedback to improve the user’s sense of control and sensitivity adaptation to make tilt interaction easier to use while walking. None of these enhancements, however, address the underlying cause of controllability problems in tilt interaction. Most existing tilt interaction techniques have relied on accelerometers, which have known shortcomings when it comes to determining device orientation (Section 3.7.1). The more widespread availability of phones equipped with digital compasses and gyroscopes enables the development of tilt interaction techniques which take advantage of sensor fusion to potentially address the shortcomings of accelerometers. This chapter will investigate the use of sensor fusion in order to determine whether the combination of accelerometer, digital compass and gyroscope data results in a measurable improvement in controllability and efficiency over accelerometer-only tilt interaction (van Tonder and Wesson 2011a).

The chapter will begin with a motivation for the use of sensor fusion to address the controllability shortcomings of tilt interaction. The design of the sensor fusion tilt interaction technique will then be described. Modifications which were made to the MapExplorer prototype are then briefly described. Finally, the results of a comparative user study between the sensor fusion and accelerometer-only tilt interaction implementations are presented and discussed.

---

1 This chapter includes content published in the following paper, which received the best paper award at SAICSIT 2011: Bradley van Tonder and Janet Wesson, The Impact of Sensor Fusion on Tilt Interaction in a Mobile Map-Based Application, In Proceedings of the 2011 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (SAICSIT 2011), 249-258, ACM.
Chapter 6: Using Sensor Fusion to Improve Tilt Interaction

6.2 Motivation

Accelerometer, digital compass and gyroscope sensors were identified in Section 3.7 as the sensors most commonly used to determine mobile phone orientation. Most of the research into the use of motion and tilt interaction over the last few years has revolved around the use of accelerometers to determine device orientation and motion (Kratz et al. 2010, Roudaut, Baglioni and Lecolinet 2009, Büning et al. 2008, Yim et al. 2011). The user studies described in Chapter 4 and Chapter 5 both rely on accelerometer-only tilt interaction implementations.

Accelerometers, digital compasses and gyroscopes all suffer from shortcomings with regard to tilt interaction (Section 3.7). Accelerometers allow tilt angles to be determined using the acceleration detected as a result of changing the orientation of the phone relative to gravity (Tuck 2007). Accelerometers also detect other sources of acceleration, including hand tremor and walking motion, making it difficult to isolate the gravity component of acceleration (Vargha et al. 2009, Nasiri et al. 2009). Digital compass sensors are susceptible to electromagnetic interference from a variety of sources, including speakers and vibration motors (Rohs and Essl 2007). Digital compasses are also inadequate to deal with quick orientation changes (Nasiri et al. 2009). Gyroscope sensors suffer from a shortcoming known as gyroscopic drift, where gradual orientation changes are reported, even when the device is stationary and the reported angles diverge from the true orientation angles (Nasiri et al. 2010).

The use of sensor fusion has been proposed to address the shortcomings of the individual sensors, by combining accelerometer, gyroscope and digital compass data. The use of a gyroscope to detect orientation changes allows the shortcomings of accelerometer input to be addressed. The use of an accelerometer allows us to compensate for the drift problems associated with the use of gyroscope data. The use of digital compass data allows more accurate absolute yaw calculation and allows the use of magnetic north as a second fixed frame of reference (in addition to gravity) relative to which tilt angles can be calculated. In the following section, the design of a sensor fusion tilt interaction approach, which combines accelerometer, gyroscope and digital compass data, is discussed.

6.3 Design of Sensor Fusion Approach

MapExplorer was modified to support tilt interaction employing sensor fusion. This was done in order to investigate whether fusion of gyroscope, accelerometer and digital compass data
results in improved controllability of tilt interaction over an accelerometer-only implementation.

The sensor fusion approach used in MapExplorer was based on a modified complementary filter implementation which was originally developed to fuse gyroscope and accelerometer data in a robot balancing application (Colton 2007). This technique was chosen because of its low computational complexity which makes it suitable for use in a resource-constrained mobile application. Other more complex sensor fusion algorithms exist, such as those based on the use of Kalman filters, but these are more computationally expensive (Sasiadek and Hartana 2000, Drolet, Michaud and Cote 2000).

The original algorithm used only accelerometer and gyroscope data, but this was modified to include compass data. The use of compass data in conjunction with accelerometer data can help improve the accuracy of raw accelerometer data by providing a second fixed external frame of reference (magnetic north) in addition to gravity (Rohs and Essl 2007). Compass data is also useful for providing absolute yaw (rotation about the z axis) data (Vargha et al. 2009). Combining accelerometer and compass data to calculate pitch and roll angles relies on the calculation of a new co-ordinate system based on the Gravity (G) and Compass (C) vectors. The cross product of the gravity and compass vectors yields a third vector \( V \) perpendicular to G and C (Formula 6.1). The cross product of this new vector \( V \) and the gravity vector G yields another vector \( W \) (Formula 6.2).

\[
V = G \times C \quad (6.1)
\]
\[
W = G \times V \quad (6.2)
\]

The three vectors (G, V and W) are orthogonal and define a co-ordinate system relative to which orientation can be calculated (Rohs and Essl 2007). The Android sensor API provides for fusion of accelerometer and compass data (Android 2011).

Gyroscope data is integrated to convert from angular velocity (radians per second) to orientation angles about the three axes (radians, then converted to degrees for easier readability) (Nasiri et al. 2009). The conversion is easily performed using the time interval between measurements to calculate the new orientation using Formula 6.3:

\[
\alpha = \omega \Delta t \quad (6.3)
\]
Where $\alpha$ = orientation angle (pitch, roll, or yaw), $\omega$ = angular velocity (as measured by the gyroscope) and $\Delta t$ = the change in time since the last reading. Accuracy of this timing information is particularly important, as incorrect timing can lead to inaccuracies.

An unfortunate side effect of this conversion process is that the output angle “drifts” over time because of the integration of the gyroscope data (Figure 6.1). In the example shown, the gyroscope data (in grey) can be clearly seen to linearly increase as time goes on, while the fused data (in black) shows a more accurate reflection of the true orientation.

![Figure 6.1. Illustration of gyroscopic drift. The grey line indicates drifting gyroscope-only data, while the black line shows stable fused data.](image)

In order to compensate for this, a high-pass filter is used (original signal – low pass filter = high-pass filter). This allows short term orientation information to pass through, while filtering out long-term drift. As per the original algorithm, the fused digital compass and accelerometer data is low-pass filtered to remove noise. The two outputs are then added together to calculate the angles which are then used to perform panning. A weighted sum is used where the weights add up to 1.

The exact weightings which are employed have to be carefully selected in order to ensure the correct balance between gyroscope (short-term relative orientation changes) and accelerometer and compass data (longer-term absolute orientation). Fused accelerometer and compass data is mainly used to compensate for gyroscopic drift and is therefore given a very low weighting in comparison to the gyroscope data. If this weighting is too low, however, it will be insufficient to correct for gyroscopic drift. If the accelerometer/compass weighting is too high, it will result in unnecessarily slow responsiveness and increased variation in the calculated orientation. The implementation used in the experiment described in this chapter...
made use of weightings of 0.98 and 0.02 for gyroscope and fused accelerometer and digital compass data (Formula 6.4). These weightings were chosen through experimentation to determine weightings which would address the drift problem while still allowing for responsive input. Fusion of accelerometer and compass data was performed using the Android API.

Gyrosopic data is reported relative to the phone’s co-ordinate system, while accelerometer and digital compass data is absolute orientation data (in world co-ordinates). As a result, the fused accelerometer and compass data needs to be subtracted from the neutral starting pitch and roll angles, whereas the gyroscope data can be used directly. The fusion algorithm is illustrated in Figure 6.2.

The formula and weightings used for sensor fusion are shown in Formula 6.4 (Colton 2007):

\[
\text{angle} = \nu(\text{angle} + \text{gyroangle} \Delta t) + \psi(\text{accelerometer} + \text{compass})_{\text{angle}}
\]

Where \(\nu = 0.98\) and \(\psi = 0.02\).

The sensor fusion process results in much smoother orientation data than the use of accelerometer data alone. To illustrate the difference, consider the extreme case shown in Figure 6.3, where sensor data was logged when tilt interaction was used while climbing stairs. The output from sensor fusion maintains a relatively smooth signal, even in this context of use. In contrast, the raw accelerometer output exhibits extreme high-frequency
variation (despite the use of a low-pass filter) as a result of linear acceleration during stair climbing.

![Figure 6.3. Comparison of pitch angle output for accelerometer approach (grey) and sensor fusion approach (black) when using tilt interaction while climbing stairs](image)

### 6.4 Accelerometer-Only Implementation

In order to ensure a meaningful comparison between the accelerometer-only and sensor fusion approaches, the accelerometer-only implementation incorporated both smoothing and discretisation. These measures have previously been shown to mitigate the shortcomings of accelerometer-only tilt interaction implementations. Raw, unfiltered accelerometer data is rarely used on its own. A low-pass filter with a filtering factor of 0.5 was used to eliminate noise from the accelerometer data. This implies that the current filtered value consists of 50% of the current raw value and 50% of previously filtered values (Formula 6.5). This value was selected through experimentation as it offered a good balance between responsiveness and smoothing. The Android API (from version 2.3 onwards) provides a “gravity” virtual sensor type (Android 2011), which also performs low-pass filtering of accelerometer data. A manual approach was preferred to allow for optimal parameter values to ensure responsiveness as well as smoothing. Therefore the acceleration values are calculated according Formula 6.5:

\[
A_t = \lambda A_t + (1 - \lambda)A_{t-1} \quad (6.5)
\]

Where \( A_t \) is the acceleration detected on a particular axis at time \( t \) and \( \lambda = 0.5 \).

Discretisation (as described in Chapters 4 and 5) was used in both the accelerometer-only and sensor fusion implementations described in this chapter.
Both the accelerometer-only and sensor fusion implementations of tilt interaction employed some of the tilt interaction enhancements proposed in the previous chapter. Visual and vibrotactile feedback and attractors were included in both implementations after the positive results shown in the previous experiment. Sensitivity adaptation was not included in order to allow for a direct comparison between the two techniques in a walking context of use without the influence of other variables. Gesture zooming was not included after mixed results with this technique were obtained in the previous user study. An alternative zooming technique was implemented and will be discussed in the following section.

6.5 MapExplorer Modifications

This section discusses modifications that were made to the original implementation of MapExplorer. MapExplorer was ported to Android and the use of tilt zooming was developed as an alternative to the gesture zooming and SDAZ techniques previously evaluated.

6.5.1 Android Implementation

MapExplorer was originally developed in Java ME and used Python for the Series 60 Symbian platform to access sensor data. In order to take advantage of the improved range of sensors available on Android devices, and the more advanced sensor supported provided by the Android API, MapExplorer was ported to the Android platform (Figure 6.4). This conversion process required only minor changes to be made to the original tilt interaction implementation that was used for the previous experiments. Despite the fact that Android uses a different unit of measurement for acceleration (m/s$^2$ as opposed to g’s), the original implementation was independent of unit of measurement and as a result did not need to be changed. The only minor change that was necessary was to invert the sign of the pitch angle calculation, as this is reversed for Android as opposed to Symbian-based Nokia devices. The Android version of MapExplorer was developed and tested on a Samsung Google Nexus S device (running Android 2.3.4) which provides accelerometer, gyroscope and digital compass sensors. The Android implementation proved to be significantly more efficient than the Java ME implementation, running at between 40 and 50 frames per second (fps) (as opposed to the 20 fps achieved on the Nokia N97). Much of this difference can be attributed to hardware differences between the two devices.
Chapter 6: Using Sensor Fusion to Improve Tilt Interaction

Figure 6.4. Android implementation of MapExplorer

All sensors were sampled at the maximum sampling rate allowed by the Android API to ensure smooth and responsive input. A performance comparison of the accelerometer and sensor fusion implementations showed that the additional complexity of the sensor fusion approach did not result in any noticeable negative efficiency effects. MapExplorer was able to run at between 40 and 50 fps in both cases, ensuring very smooth interaction.

MapExplorer was also modified to make use of a portrait screen orientation instead of the landscape orientation used in the user studies described in Chapters 4 and 5. This decision was taken because it was found that portrait orientation provided for easier one handed interaction, while still allowing the user to comfortably reach the screen to activate and deactivate tilt interaction. Observation of participants in the earlier user studies showed that some users found one handed interaction difficult in landscape orientation.

6.5.2 Tilt Zooming

In the previous user study, gesture zooming was used, where discrete zooming operations could be performed using vertical gestures relative to the z-axis. This technique was designed as an alternative to techniques such as Speed-Dependent Automatic Zooming (SDAZ) and Semi-Automatic Zooming (SAZ) (Kratz et al. 2010, Esfambolchilar and Murray-Smith 2004), which link the zoom level to the current panning speed. The response by participants in the previous evaluation to gesture zooming was mixed. Participants complained about gesture recognition errors and some participants found the zooming gestures uncomfortable.
to perform. The design of gesture zooming was also limited in that users were only able to zoom in or out by one zoom level at a time. Furthermore, feedback was not always immediate as loading map images sometimes resulted in slight delays.

Tilt zooming was implemented as an alternative to gesture zooming. Tilt zooming allows zooming to be performed by tilting the phone about the x-axis (adjusting the pitch angle). Tilting the phone towards the user zooms the display in, while tilting the phone away from the user zooms the display out. In order to distinguish zooming input from panning input, the user is required to hold in a button on the left-hand side of the phone while performing zooming operations (Figure 6.5). When the button is depressed, visual confirmation is provided to the user that he/she has entered zooming mode and the current zoom level is indicated using a zoom level indicator on the right hand side of the display. A position control mapping is used, with pitch angles mapped directly onto zoom levels. As the user tilts the phone up or down, the map is scaled to provide immediate feedback to the user about the results of his/her actions. When the zooming button is released, the zoom level is adjusted to the nearest matching zoom level and the application returns to panning mode.

A similar approach has previously been employed on several Android devices to perform zooming in photo and web browser applications. Physical hardware buttons and touching the screen with two fingers have both been employed as clutch mechanisms during zooming. Tilt zooming while the user holds his/her thumb on the screen has also recently been proposed (Hinckley and Song 2011). The use of touch as a means of activating zooming has the disadvantage of occluding the display during zooming operations.
Chapter 6: Using Sensor Fusion to Improve Tilt Interaction

The use of tilt zooming has several advantages over existing zooming techniques employed in tilt-controlled mobile map-based applications. Unlike techniques such as SDAZ and SAZ, which link the zoom level to the panning speed, this approach allows the user full control over the zoom level. Furthermore, this approach allows the user to perform zooming independently of panning operations. In contrast to other motion controlled approaches such as gesture zooming, this approach also allows the user to vary the current zoom level by more than one level at a time.

6.6 Experimental Design

A user study was conducted in order to compare the accelerometer-only implementation with the sensor fusion implementation.

6.6.1 Objectives

The user study was conducted in order to determine whether the theoretical benefits of sensor fusion resulted in significant controllability benefits for tilt interaction in a mobile map-based application (Nasiri et al. 2010). The objectives of the user study were as follows:

- To determine whether sensor fusion facilitated improved perceived controllability over the accelerometer-only implementation.
- To determine whether sensor fusion facilitated improved controllability over the accelerometer-only implementation.
- To determine whether sensor fusion facilitated improved efficiency over the accelerometer-only implementation.

In order achieve the above objectives, several hypotheses will be evaluated. These are outlined in the next section.

6.6.2 Hypotheses

The following hypotheses will be tested in this user study:

- \(H_{0,1}\): Tilt interaction using sensor fusion does not exhibit better perceived controllability than accelerometer-only tilt interaction.
- \(H_{0,2}\): Tilt interaction using sensor fusion does not exhibit better controllability than accelerometer-only tilt interaction.
• $H_{03}$: Tilt interaction using sensor fusion is not more efficient than accelerometer-only tilt interaction.

Each of the above hypotheses can be decomposed into sub-hypotheses by task type and context of use.

### 6.6.3 Metrics

In order to test the above hypotheses, the following metrics will be collected as part of this user study:

- **Efficiency**: Measured using task times. Task times were calculated by logging the time between closing the task instructions and pressing the button to start the next task (bottom right of Figure 6.4).
- **Controllability**: Measured for navigation tasks using the mean deviation from the planned routes (measured using the Euclidean distance between the actual path followed and the planned route) and the number of waypoints missed. For locating tasks, the number of times the cursor entered and left the target POI region and the time between POI region entry and selection were used.
- **Perceived Controllability**: Measured using a questionnaire using seven point interval scales (Appendix G). Participants were asked to identify which technique was easier to control for each task type while seated and walking.

Similarly to the previous experiments, data logging functionality was built into MapExplorer to log task times, tilt angles and actions such as zooming, panning and selecting. This data was used to log and measure the performance metrics outlined above.

### 6.6.4 Participants

A total of 17 participants (6 female, 11 male) participated in the user study. All participants were students in the Department of Computing Sciences between the ages of 20 and 30 (mean age of 24). The participant profile was therefore similar to the participants used in the first two user studies. Four participants were left handed and most reported occasional prior experience with mobile map-based applications.
6.6.5 Tasks

Participants were required to complete two types of tasks. The first task type involved finding and selecting POI icons (referred to as locating tasks). The second task type involved following a planned route from start to finish (referred to as navigating tasks). Routes were split into a number of segments, separated by instruction waypoints. Route segments changed colour to indicate successful completion to encourage users to follow the routes as closely as possible (Figure 6.6). Green indicates successfully completed segments, red unsuccessfully completed segments and blue indicates segments still to be completed. Unlike the user studies described in Chapters 4 and 5, navigating tasks did not require users to plan the routes by selecting start and end points. Instead, pre-planned routes were provided. This was done in order to ensure that the task times and relevant questionnaire responses were not affected by the time taken to find and select the start and end points.

![Figure 6.6. Example of a navigation task in MapExplorer](image)

6.6.6 Independent Variables

A within-subjects approach was used for all test conditions. The following independent variables were considered in the user study:

- **Sensors**: Half the tasks were completed using accelerometer-only input, while the other half were completed using fused accelerometer, digital compass and gyroscope data. The order in which the two configurations were used was counter-balanced to offset any order effects. For ease of reference, these two implementations will be
Chapter 6: Using Sensor Fusion to Improve Tilt Interaction

referred to as the accelerometer implementation and the sensor fusion implementation in the following discussion.

- **Task Type**: Participants were required to perform both locating and navigating tasks. Two similar task sets were used and counter-balanced across participants to offset any learning effects.

- **Context of Use**: Half the tasks were completed while seated and the other half while walking. This was an important variable to be considered in this evaluation, as many of the shortcomings of accelerometer-only tilt implementations are only evident when the user is mobile. Seated tasks were always performed before walking tasks to allow participants to get accustomed to the implementations before the additional complication of walking was introduced.

### 6.6.7 Test Environment

The user study was conducted in a laboratory environment in the Department of Computing Sciences at NMMU. Walking tasks were performed in an indoor corridor with a mixture of artificial and natural lighting.

### 6.6.8 Materials

A Samsung Google Nexus S running Android version 2.3.4 with MapExplorer installed was used in the user study.

### 6.6.9 Procedure

Participants were required to complete an informed consent form and biographical questionnaire before commencing the user study (Appendices A and B). Participants were given an overview of MapExplorer, the task types and tilt interaction before beginning the tasks. Participants were allowed to experiment with the sensor fusion and accelerometer-only implementations prior to starting the corresponding set of tasks. Two tasks of each task type were performed while seated and while walking with each tilt interaction implementation (16 tasks in total). Once participants completed the tasks, they were required to complete the post-task questionnaire (Appendix G).

### 6.7 Results

Performance results are presented first, followed by perceived controllability results. As with the user study in Chapter 5, non-parametric Wilcoxon signed-rank tests were used to compare
mean values for this user study because the relatively small sample size does not allow us to assume normally distributed data.

### 6.7.1 Performance Results

Task times were calculated using logged data of participants’ interaction with MapExplorer. Efficiency data organised by context of use is shown in Table 6.1 (statistically significant results shown in bold). The sensor fusion approach proved more efficient than the accelerometer-only implementation for both seated and walking tasks. This difference was narrowly not significant for seated tasks ($p = 0.076$). For walking tasks, the difference in efficiency was statistically significant ($p = 0.031$), as was the overall difference ($p = 0.031$).

![Image](https://via.placeholder.com/150)

<table>
<thead>
<tr>
<th></th>
<th>Accelerometer</th>
<th>Sensor Fusion</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
</tr>
<tr>
<td>Seated</td>
<td>116.77</td>
<td>37.74</td>
<td>100.10</td>
</tr>
<tr>
<td>Walking</td>
<td>114.02</td>
<td>31.60</td>
<td>94.77</td>
</tr>
<tr>
<td>Total</td>
<td>230.79</td>
<td>78.21</td>
<td>194.87</td>
</tr>
</tbody>
</table>

Table 6.1. Wilcoxon signed-rank test results comparing mean task times (in seconds) for the two implementations ($n = 17$).

Table 6.2 shows the mean task times organised by task type. Sensor fusion was more efficient for both locating and navigating tasks. The difference was statistically significant for locating tasks ($p = 0.031$). The difference for navigating tasks was not statistically significant.

![Image](https://via.placeholder.com/150)

<table>
<thead>
<tr>
<th></th>
<th>Accelerometer</th>
<th>Sensor Fusion</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
</tr>
<tr>
<td>Locating</td>
<td>122.32</td>
<td>53.63</td>
<td>94.48</td>
</tr>
<tr>
<td>Navigating</td>
<td>108.47</td>
<td>36.16</td>
<td>100.39</td>
</tr>
</tbody>
</table>

Table 6.2. Wilcoxon results comparing mean task times (in seconds) organised by task type ($n = 17$)

Statistical analysis of the mean task times was also conducted for the different combinations of task types and contexts of use. The sensor fusion approach was found to be more efficient for locating tasks performed while walking ($\text{Wilcoxon } Z = 2.15, p = 0.031$). The difference in task times for navigating tasks performed while seated was narrowly not statistically significant ($\text{Wilcoxon } Z = 1.92, p = 0.055$).

Accuracy data for the navigating tasks was calculated using log data by determining the mean deviation from the planned route (in pixels). Table 6.3 shows the mean Euclidean distance (in pixels) for different task types and contexts of use.

![Image](https://via.placeholder.com/150)

<table>
<thead>
<tr>
<th></th>
<th>Accelerometer</th>
<th>Sensor Fusion</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
</tr>
<tr>
<td>Seated</td>
<td>120.32</td>
<td>54.56</td>
<td>100.10</td>
</tr>
<tr>
<td>Walking</td>
<td>115.02</td>
<td>36.00</td>
<td>94.77</td>
</tr>
<tr>
<td>Total</td>
<td>230.34</td>
<td>78.05</td>
<td>194.87</td>
</tr>
</tbody>
</table>

Table 6.3. Wilcoxon results comparing mean Euclidean distances (in pixels) organised by task type ($n = 17$)
pixels) between the actual route and the route followed by participants in the user study. In both cases, greater deviations from the planned routes were recorded when completing navigating tasks while walking as opposed to while seated. Accuracy for both seated tasks and walking tasks was very similar for both implementations. Accuracy was slightly better using the accelerometer implementation for seated tasks and slightly better using the sensor fusion implementation for walking tasks. Neither difference was statistically significant. A statistically significant difference was found between seated and walking tasks for both the accelerometer (Wilcoxon Z = 3.62, p < 0.001) and sensor fusion implementations (Wilcoxon Z = 2.48, p = 0.013).

<table>
<thead>
<tr>
<th></th>
<th>Accelerometer</th>
<th>Sensor Fusion</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seated</td>
<td>12.65</td>
<td>13.41</td>
<td>0.91</td>
</tr>
<tr>
<td>Walking</td>
<td>20.94</td>
<td>18.29</td>
<td>1.47</td>
</tr>
</tbody>
</table>

Table 6.3. Wilcoxon results comparing mean accuracy for navigating tasks. Deviation from the actual route reported in pixels (n = 17).

A statistically significant difference was recorded for one of the more complex route-following tasks performed while walking (Figure 6.7). The mean deviation was smaller (better) for this task for the sensor fusion implementation than for the accelerometer implementation (Wilcoxon Z = 2.52, p = 0.01).

Figure 6.7. Comparison of accelerometer (left) and sensor fusion (right) route plots of a routing task performed while walking. The actual route is shown as a thin blue line.
Figure 6.7 shows the plots of the routes followed by participants overlaid on the actual route (shown in blue). The figure shows that participants who used the sensor fusion implementation were able to follow the route more accurately than those using the accelerometer-only implementation. The same trend was evident to a lesser extent in the other navigation tasks performed while walking.

The number of waypoints missed by participants during navigation tasks was also logged. More errors were recorded when using the accelerometer implementation (mean of 1.76 waypoints missed for the two walking tasks) than the sensor fusion implementation (mean of 1.18 errors). This difference was not statistically significant (Wilcoxon Z = 1.02, p = 0.31).

For locating tasks, controllability was measured by counting the number of crossings participants made when attempting to select a POI icon. The number of crossings was defined as the number of times the cursor enters the region of the target icon when performing a selection operation (Xin et al. 2011). Table 6.4 shows the results of this analysis. The results were slightly in favour of sensor fusion, with a more noticeable difference recorded for walking tasks. This difference was not, however, statistically significant. Attractors were used with both implementations, and are likely to have contributed to this difference not being as pronounced as might have been expected.

<table>
<thead>
<tr>
<th></th>
<th>Accelerometer</th>
<th>Sensor Fusion</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
</tr>
<tr>
<td>Seated</td>
<td>1.47</td>
<td>0.41</td>
<td>1.44</td>
</tr>
<tr>
<td>Walking</td>
<td>1.74</td>
<td>0.66</td>
<td>1.56</td>
</tr>
</tbody>
</table>

Table 6.4. Wilcoxon results comparing mean number of crossings per selection (n = 17)

The mean time between first entering the region of a POI icon and actually selecting that icon was also calculated (25 pixels around the icon was regarded as the region of the icon and the area within which the attractor mechanism “magnetised” the cursor onto the icon). The results showed that while seated, the time between entering the POI region and performing selection was almost the same for both the accelerometer and sensor fusion approaches (Table 6.5). While walking, the difference was more noticeable (participants took longer to perform selections when using the accelerometer-only implementation). This difference was statistically significant (p = 0.020).
Chapter 6: Using Sensor Fusion to Improve Tilt Interaction

In order to gain an understanding of some of the results obtained above, sensor output from the two approaches was analysed. Mean variance values were calculated for all tasks for each participant. Figure 6.8 clearly shows the difference between the variance in output for the two implementations. The sensor fusion approach produced a smaller variance in terms of pitch and roll angles (more than 30% smaller in both cases). The difference was statistically significant in both cases (Wilcoxon $Z = 3.72$, $p < 0.001$ in both cases).

This difference is perhaps better illustrated by looking at the output for an individual user for a specific task. Figure 6.9 shows the difference for a user performing a navigating task while walking. The plot clearly shows the greater variability in the accelerometer data (shown in grey) while walking. In contrast, the sensor fusion data (shown in black) exhibits a much smoother trend.

<table>
<thead>
<tr>
<th></th>
<th>Accelerometer</th>
<th>Sensor Fusion</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
</tr>
<tr>
<td>Seated</td>
<td>3.20</td>
<td>1.45</td>
<td>3.11</td>
</tr>
<tr>
<td>Walking</td>
<td>4.63</td>
<td>3.66</td>
<td>2.61</td>
</tr>
</tbody>
</table>

Table 6.5. Wilcoxon results comparing mean time (in seconds) between entering the region of an icon and performing a selection operation ($n = 17$)

Figure 6.8. Comparison of mean variance for pitch and roll angles (where angles were measured in degrees) for sensor fusion and accelerometer implementations. 95% confidence intervals shown ($n = 17$).
Figure 6.9. Comparison of sensor fusion and accelerometer output for a walking navigation task

Given the large difference in terms of variance for the two implementations, a greater difference in terms of controllability was expected. As a result, further analysis of the output of the two approaches was done in order to determine whether the variation in tilt angles translated into a greater variation in latitude and longitude on the map (which could be indicative of greater difficulty in maintaining a steady cursor position). The results are shown in Figure 6.10. The sensor fusion approach produced a smaller mean variance in latitude and longitude, although this difference was only significant for latitude (Wilcoxon Z = 2.25, p = 0.02).
6.7.2 Perceived Controllability Results

Participants were asked to complete a post-test questionnaire asking them to rate the controllability of the two implementations for the different test conditions (Table 6.6). Seven point interval scales were used for all questions (where 1 = Accelerometer and 7 = Sensor Fusion).

<table>
<thead>
<tr>
<th>#</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>While seated, which technique was easier to control for locating tasks?</td>
</tr>
<tr>
<td>2.</td>
<td>While seated, which technique was easier to control for navigating tasks?</td>
</tr>
<tr>
<td>3.</td>
<td>While walking, which technique was easier to control for locating tasks?</td>
</tr>
<tr>
<td>4.</td>
<td>While walking, which technique was easier to control for navigating tasks?</td>
</tr>
</tbody>
</table>

Table 6.6. Questionnaire used to measure perceived controllability for different task types and contexts of use

Perceived controllability results are shown in Table 6.7. The mean ratings were in favour of the sensor fusion approach for all questions. The differences (from the neutral value of 4) were statistically significant for the questions related to walking, with sensor fusion rated higher for both locating (Wilcoxon Z = 2.07, p = 0.038) and navigating tasks (Wilcoxon Z = 1.96, p = 0.049).

<table>
<thead>
<tr>
<th></th>
<th>Preference</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
</tr>
<tr>
<td>Locating + Seated</td>
<td>4.24</td>
<td>1.75</td>
</tr>
<tr>
<td>Locating + Walking</td>
<td>5.12</td>
<td>1.96</td>
</tr>
<tr>
<td>Navigating + Seated</td>
<td>4.53</td>
<td>1.94</td>
</tr>
<tr>
<td>Navigating + Walking</td>
<td>5.06</td>
<td>2.11</td>
</tr>
</tbody>
</table>

Table 6.7. Wilcoxon results comparing mean post-test questionnaire perceived controllability ratings.

Seven point interval scales were used, where 1 = Accelerometer and 7 = Sensor Fusion (n = 17)

Qualitative comments largely supported the ratings given, with most feedback in favour of the sensor fusion implementation. Several participants indicated that they found the sensor fusion implementation to be easier to control, particularly when walking. Participants also found the sensor fusion approach to be smoother. Feedback regarding the tilt zooming technique was very positive with users liking the immediate feedback and intuitive nature of this technique. One participant found that the position of the zooming button was difficult to reach.
6.8 Discussion

In theory, the use of the sensor fusion approach should have resulted in smoother interaction and should have afforded the participants greater control over cursor position and panning speed and direction for walking tasks. While several statistically significant improvements were recorded, it is perhaps most interesting to note that the sensor fusion approach did not result in a statistically significant improvement in controllability for route-following tasks performed while walking. This is in contrast to participants’ perceptions, with a statistically significant perceived controllability improvement reported for the sensor fusion implementation for walking tasks. The accelerometer-only implementation suffered from minor cursor oscillation as a result of walking motion, which may have influenced participants’ ratings.

The lack of a statistically significant controllability improvement can be offset against the strongly significant improvement in efficiency achieved by the sensor fusion implementation. This result is somewhat surprising, given that an identical mapping between tilt angles and panning speed was used for both implementations. One possible explanation could be that participants felt that they had to be more careful using the accelerometer-only implementation and as a result performed the accelerometer tasks more slowly (thereby achieving comparable accuracy, but slower efficiency). This could also serve to explain the discrepancy between the objective and subjective controllability results.

It is also interesting to note that a statistically significant improvement in efficiency was achieved for locating tasks but not for navigating tasks. Analysis of the locating tasks showed that for tasks performed while walking, the time between entering the region of the target icon and selecting that icon was significantly faster using the sensor fusion implementation. This suggests that while walking, participants were able to settle the cursor quicker using the sensor fusion implementation. It is also noteworthy that selection operations took significantly longer when walking as opposed to when seated using the accelerometer implementation, whereas this was not the case using the sensor fusion implementation.

The comparative analysis between the pitch and roll angles calculated using accelerometer data and those calculated using sensor fusion clearly indicates a significantly higher variance in the angles calculated using the accelerometer data (Figures 6.9 and 6.10). It is interesting to note that this trend is not as noticeable when comparing the variation in latitude and longitude (which is more directly indicative of cursor movement and therefore
controllability). A possible explanation can be found in the fact that both implementations made use of discretisation, with angles falling into the same interval having the same effect on panning speed and direction. As a result, much of the variation in the pitch and roll angles calculated using the accelerometer would likely have been smoothed out using this technique. This is also likely to be a reason why the difference in controllability for navigation tasks was not as significant as might have been expected. This would indicate that discretisation proved more effective in mitigating the shortcomings of tilt interaction than the use of smoothing techniques such as the low-pass filter which was also employed. This is particularly interesting because the use of discretisation in the past has largely been limited to position controlled tilt interaction, rather than the rate controlled approach used in this research.

Subjective results indicated only a slight preference for the sensor fusion implementation while seated, but a statistically significant difference for walking tasks. This result is in line with expectations, as in theory there should have been little difference experienced for seated tasks, with the difference being more noticeable for walking tasks. This trend is further supported by the qualitative comments, with several participants pointing out that the difference between the two implementations was more noticeable for walking tasks.

It is also noteworthy that both the accelerometer and sensor fusion approaches performed significantly worse in terms of controllability for navigating tasks when walking than when seated. Sensor fusion is often proposed as a potential solution for the problems of separating linear acceleration (which is particularly prevalent when walking) from gravity in tilt interaction implementations. The results of this experiment suggest that merely employing sensor fusion alone only goes part of the way to solving this problem. Additional measures are necessary to improve the controllability of tilt interaction techniques when walking. Furthermore, it should be borne in mind that the accelerometer-only implementation used in this user study employed a low-pass filter which was likely responsible for minimising the shortcomings of tilt interaction, particularly while walking.

Feedback regarding the tilt zooming technique used in this experiment was positive. Participants found controlling the zoom level using the pitch angle (in conjunction with a hardware clutch button) to be intuitive and were quickly able to master the technique. This feedback suggests that this technique may provide a viable alternative to techniques such as SDAZ and SAZ.
6.9 **Hypothesis Testing**

In Section 6.6.2 several hypotheses were introduced to be tested in this user study. This section revisits these hypotheses in order to determine whether the results obtained allow the null hypotheses to be rejected.

\( H_{0,1} \): *Tilt interaction using sensor fusion does not exhibit better perceived controllability than accelerometer-only tilt interaction.*

The mean ratings for perceived controllability showed that a statistically significant difference was obtained for both navigating and locating tasks while walking (in favour of sensor fusion) (Table 6.7). Statistically significant differences were not recorded for seated tasks. There is therefore insufficient evidence to fully reject the null hypothesis \( H_{0,1} \). There is however, sufficient evidence to reject the sub-hypotheses for both navigating and locating tasks performed while walking (Table 6.8).

<table>
<thead>
<tr>
<th></th>
<th>Seated</th>
<th>Walking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigating</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Locating</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Table 6.8. Hypothesis testing results for perceived controllability**

\( H_{0,2} \): *Tilt interaction using sensor fusion does not exhibit better controllability than accelerometer-only tilt interaction.*

Analysis of log data did not reveal any statistically significant improvement in terms of accuracy for route-following tasks (Table 6.3). Participants were, however, able to settle the cursor to perform selection operations quicker while walking using the sensor fusion implementation. Therefore, while there is insufficient evidence to reject the null hypothesis \( H_{0,2} \), there is sufficient evidence to reject the sub-hypothesis for locating tasks performed while walking (Table 6.9).

<table>
<thead>
<tr>
<th></th>
<th>Seated</th>
<th>Walking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigating</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Locating</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Table 6.9. Hypothesis testing results for controllability**
H₀,₃: Tilt interaction using sensor fusion is not more efficient than accelerometer-only tilt interaction.

Participants completed walking tasks faster using the sensor fusion implementation. This difference was statistically significant (Table 6.1). Analysis by task type showed that the difference was significant for locating tasks, but not for navigating tasks performed while walking. Therefore there is insufficient evidence to reject the null hypothesis H₀,₃. There is, however, sufficient evidence to reject the sub-hypothesis for locating tasks performed while walking (Table 6.10).

<table>
<thead>
<tr>
<th></th>
<th>Seated</th>
<th>Walking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigating</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Locating</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 6.10. Hypothesis testing results for efficiency

6.10 Design Implications

Following on from the design recommendations identified in Section 5.7, the results of the user study allow further insight into the design of tilt interaction techniques. In addition to the three design recommendations proposed in Section 5.7, the following design recommendation is proposed based on the results presented in this chapter:

R₄. Use sensor fusion to improve perceived controllability and efficiency when the user is mobile.

The results of this user study provide empirical evidence that the use of sensor fusion can result in improved perceived controllability and efficiency over accelerometer-only tilt interaction implementations. Some controllability benefits were also achieved, particularly for locating tasks performed while walking. Sensor fusion also did not negatively affect system performance. Given the importance of mobile contexts of use for mobile map-based applications, it is therefore recommended that sensor fusion be employed to calculate device orientation in future tilt interaction implementations.

6.11 Discussion

Previous research into the use of tilt interaction has almost exclusively involved the use of accelerometers (Büring et al. 2008, Cho et al. 2007a). The user study described in this
chapter is the first to provide a detailed comparison of the use of accelerometer-only and sensor fusion implementations of tilt interaction. Furthermore, the sensor fusion implementation described in this chapter shows that sensor fusion can achieve usability improvements using a fairly simple and computationally inexpensive implementation. Seated and walking test conditions were used to show that the benefits of sensor fusion are only apparent when the user is mobile.

6.12 Conclusions

The aim of this chapter was to determine whether the theoretical advantages of sensor fusion over accelerometer-only tilt interaction implementations could be confirmed by measurable efficiency and controllability benefits. A relatively simple and computationally inexpensive sensor fusion algorithm (originally designed for balancing robots) was modified and implemented in a mobile map-based application. The algorithm was able to perform at the same efficiency as an accelerometer-only implementation and was shown to produce smoother output in terms of tilt angles. Furthermore, it was able to greatly reduce the variability in tilt angles when the user is mobile.

A user study was conducted to compare accelerometer-only and sensor fusion implementations of tilt interaction in both seated and walking contexts of use. The results of the user study showed that the sensor fusion approach did indeed provide several improvements. Sensor fusion was shown to offer greater efficiency and perceived controllability than the accelerometer-only implementation for walking tasks.

Accelerometer input was shown to offer comparable performance for seated tasks, but was shown to be inferior in several respects for walking tasks. This experiment provides empirical evidence that the additional complexity of sensor fusion is worthwhile in order to make motion-controlled techniques such as tilt interaction feasible in a mobile context of use. This is particularly important for a domain such as mobile map-based applications, where users are likely to be walking while interacting with the application.

The results of this evaluation show that sensor fusion can be efficiently incorporated into a tilt interaction technique in a mobile map-based application. Furthermore, sensor fusion can achieve significant efficiency and perceived controllability improvements for walking tasks. This is not, however, a complete solution to the controllability problem of tilt interaction.
while walking. Further work remains to determine how the controllability of tilt interaction can be improved to make this form of interaction more suitable for a mobile context of use.

The following chapter will discuss design modifications intended to improve the controllability of tilt interaction. These modifications will seek to address the shortcomings of the zooming techniques evaluated in Chapter 5, and further refine the sensitivity adaptation technique that was proposed.
Chapter 7: Improving the Controllability of Tilt Interaction

7.1 Introduction

This chapter will investigate how the controllability of tilt interaction can be further improved based on the experimental findings from Chapters 4, 5 and 6. Two particular aspects of tilt interaction are investigated, namely zooming techniques and controllability while walking.

In Chapter 5, gesture zooming was compared to SDAZ. Both techniques were shown to suffer from several shortcomings. The use of tilt zooming was proposed in Chapter 6 and showed promising results. In this chapter, tilt zooming will be compared to gesture zooming to determine whether the use of pitch angles to control zooming speed can improve upon the shortcomings of gesture zooming.

Modifications to the original implementation of sensitivity adaptation will also be proposed in this chapter. The original implementation made use of fixed thresholds to identify walking and seated contexts of use. Furthermore, a binary approach to sensitivity adaptation was employed with pre-defined sensitivity levels for seated and walking contexts of use. Sensitivity adaptation will be modified to determine whether the use of a more repeatable and scalable technique for determining the user’s context and dynamic sensitivity adaptation can provide further benefits.

This chapter will begin with a review of current zooming techniques and highlight the shortcomings of each technique identified in the previous user studies. An implementation of rate controlled tilt zooming is then described. Sensitivity adaptation will then be revisited, in order to determine how the shortcomings of the implementation presented in Chapter 5 can be addressed. Finally, the results of a user study evaluating these design modifications are presented and analysed.

7.2 Zooming Techniques

The results of the user studies described in Chapters 4, 5 and 6 provided some insight into the use of different zooming techniques in conjunction with tilt interaction. While the sense of overview provided by SDAZ was regarded as useful for locating POIs, several shortcomings of this technique were also identified. The use of SDAZ was criticised for taking control away from users and was found to be difficult to control while walking. SDAZ was also
described as disorienting (Section 5.5.5). Gesture zooming was proposed as an alternative, but was found to suffer from gesture recognition problems. A position controlled tilt zooming technique was trialled in the user study described in Chapter 6. While participants were generally positive about the tilt zooming technique, some problems were encountered. One participant found the hardware button used as a clutch mechanism difficult to reach (Section 6.7.2). Zooming operations also often ended with the phone tilted away from the user, requiring users to reorient the phone before reactivating tilt panning.

Table 7.1 summarises the advantages and disadvantages of the three zooming techniques.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SDAZ</strong></td>
<td>• No need to switch between panning and zooming</td>
</tr>
<tr>
<td></td>
<td>• Provides sense of overview</td>
</tr>
<tr>
<td></td>
<td>• Lack of user control</td>
</tr>
<tr>
<td></td>
<td>• Difficult to use while walking</td>
</tr>
<tr>
<td></td>
<td>• No zooming independent of panning</td>
</tr>
<tr>
<td><strong>Gesture Zooming</strong></td>
<td>• Intuitive</td>
</tr>
<tr>
<td></td>
<td>• User retains control</td>
</tr>
<tr>
<td></td>
<td>• Recognition errors</td>
</tr>
<tr>
<td></td>
<td>• Discrete due to sensor limitations (single zoom level at a time)</td>
</tr>
<tr>
<td></td>
<td>• Potentially uncomfortable</td>
</tr>
<tr>
<td><strong>Tilt Zooming</strong></td>
<td>• Intuitive</td>
</tr>
<tr>
<td></td>
<td>• Not restricted to one zoom level at a time</td>
</tr>
<tr>
<td></td>
<td>• User retains control</td>
</tr>
<tr>
<td></td>
<td>• Need to switch between panning and zooming modes</td>
</tr>
<tr>
<td></td>
<td>• The need for a clutch mechanism</td>
</tr>
</tbody>
</table>

Table 7.1. Comparison of zooming technique approaches

The following sections describe modifications that were made to the original gesture zooming and tilt zooming implementations described in Chapters 5 and 6. The original implementation of gesture zooming described in Chapter 5 was modified to address the gesture recognition shortcomings encountered during the previous user study. Tilt zooming was modified to implement rate control instead of position control to facilitate easier switching between panning and zooming modes.

### 7.3 Gesture Zooming Modifications

The results of the user study in Chapter 5 showed very mixed results regarding the use of gesture zooming. Gesture zooming was marginally preferred over SDAZ, but users reported struggling with gesture recognition errors. Analysis of the log data from this experiment
revealed a shortcoming in the gesture recognition approach that was employed. Zooming gestures were identified by looking for distinctive acceleration and deceleration patterns indicating rapid movement of the phone forwards or backwards relative to the z-axis (left of Figure 7.1). The order of the acceleration and deceleration peaks in the z-axis acceleration data was used to identify the zooming direction. Zooming gestures were identified by calculating the difference between the lowest and highest z-axis acceleration values in the current time window and the current baseline acceleration. If the sum of these two differences exceeded a pre-determined threshold, a zooming gesture event was fired. In some situations this could lead to incorrect gesture recognition, particularly if the initial acceleration or deceleration was very large. In such cases (right of Figure 7.1), initial acceleration or deceleration is recognised as a complete gesture because the difference from the baseline acceleration is large enough to exceed the defined threshold.

![Correctly identified zooming gesture (left), erroneously identified gesture (right)](image)

Figure 7.1. Correctly identified zooming gesture (left), erroneously identified gesture (right)

The gesture recognition algorithm was modified to address this shortcoming. The algorithm was changed to monitor the median z-axis acceleration value for the last second and check for spikes above and below this median value exceeding a pre-defined threshold. In this way, the algorithm improves on the original implementation by checking for a complete wave pattern, rather than simply for a difference in acceleration, eliminating errors such as those illustrated in Figure 7.1.

A clutch mechanism was introduced to allow gesture zooming to be made more sensitive. The previous implementation attempted gesture zooming without a clutch mechanism, which meant that the threshold for identifying gestures had to be set relatively high to avoid false positive gesture recognition errors. This likely contributed to users feeling uncomfortable when using the gesture zooming technique. Users have a choice between a hardware clutch
button, which makes use of the phone’s volume buttons (either button) and a touch-screen clutch button (Figure 7.2). The clutch button is kept depressed while the user is performing zooming. The touch-screen button was designed to be large enough to easily activate zooming and is semi-transparent to avoid obscuring the map unnecessarily.

Visual feedback is provided to the user in the form of on-screen arrows indicating the mapping between direction and the resultant effect on zoom level (Figure 7.2). When a gesture is correctly identified, the corresponding arrow is highlighted in green to indicate to the user that the gesture has been successfully recognised. The current zoom level is displayed while the clutch button is depressed.

**Figure 7.2.** Visual feedback when performing gesture zooming

### 7.4 Tilt Zooming Modifications

In Chapter 6, a preliminary implementation of tilt zooming was employed during the experiment to compare accelerometer-only tilt interaction with a sensor fusion approach. This tilt zooming implementation used position control to link the current pitch angle to a zoom level. Although this approach was not formally evaluated in the user study, positive comments were received from participants regarding this zooming technique. The most notable shortcomings were that one participant had trouble reaching the button used as a clutch mechanism for performing zooming. In order to compensate for this shortcoming, it was decided to include a large, easy to select touch-screen button as an alternative
mechanism for activating zooming mode. Users are required to keep their finger/thumb on this button for the duration of the zooming interaction. In order to eliminate the need for the user to switch attention between the map and the zoom level indicator on the edge of the display, very short vibration feedback was added to indicate changes in zoom level to the user. Visual feedback, in the form of on-screen arrows, is provided to the user to indicate the mapping between tilt direction and the resultant effect on zoom level (Figure 7.3).

![Image](image.jpg)

**Figure 7.3. Performing tilt zooming in MapExplorer**

The original tilt zooming implementation employed position control. This worked well for controlling the zoom level, but often resulted in zooming gestures being completed with the phone tilted far from the starting position or at an uncomfortable angle for the user. As a result, panning could not be immediately resumed once tilting was completed. The solution employed in the original implementation was to disable panning once tilt zooming was initiated, therefore requiring the user to re-position the phone into a comfortable orientation, before re-enabling panning. To address this shortcoming, a rate control implementation of tilt zooming was implemented.

A similar mapping was used for rate controlled tilt zooming as was employed for panning. Pitch angles of less than or equal to five degrees are ignored to allow the user to maintain a constant zoom level. Pitch angles between five and fifteen degrees either side of the neutral pitch (the pitch angle when tilt zooming is activated) are used to control the speed at which the zoom level changes. This fairly small tilt range was employed to minimise display
visibility problems. Small pitch angles result in slow zoom level changes, with larger pitch angles resulting in faster zoom level changes. As with panning, a linear mapping is used for smaller angles, with a quadratic mapping used for larger tilt angles. This approach is intended to allow for fine control using small tilt angles and more rapid changes with larger tilt angles when quicker zoom level adjustment is required.

### 7.5 Overview

In the earlier user studies that included SDAZ, several positive comments were received regarding the fact that SDAZ helped to provide a sense of overview. Participants found this particularly useful for tasks in which distant POIs needed to be located. Overview functionality is often cited as a fundamental feature of information visualisation applications, allowing users to navigate a large information space effectively (Hornbæk and Hertzum 2011). In order to mitigate the loss of this capability, an overview feature was added to MapExplorer. This overview was implemented as a zoomed out view two zoom levels lower than the current zoom level. The overview is displayed in the top right corner of the display (Figure 7.4). The currently visible portion of the overview is denoted by a blue rectangle in the centre of the overview.

![Figure 7.4. Overview in MapExplorer (top right corner)](image)

It has previously been argued that this form of traditional overview is unsuitable for use on mobile devices, because small screen sizes make detail difficult to distinguish and overviews can occlude large portions of the display (Chittaro 2006). In the intervening five years,
Chapter 7: Improving the Controllability of Tilt Interaction

however, mobile phone displays have improved dramatically both in terms of physical size and screen resolution. Nevertheless, several steps were taken in order to minimise these problems. The display of the overview is linked to the panning speed, and is only displayed when the user is panning at the third panning speed level or higher. This was done in order to avoid the overview obstructing the display when the user is moving slowly, while still making it available to assist the user to locate distant targets on the map. Furthermore, the overview is semi-transparent in order to prevent the user missing any POI icons that might be occluded.

The following section discusses sensitivity adaptation and modifications that were made to the original implementation discussed in Chapter 5.

7.6 Sensitivity Adaptation

The original implementation of sensitivity adaptation described in Chapter 5 yielded promising results. Log data analysis indicated that this approach helped to improve controllability for route-following tasks performed while walking. The original design and implementation was, however, device and platform specific. The thresholds employed to identify walking and seated contexts of use were based on observation of the particular sensor characteristics of the Nokia N97. These thresholds could therefore not be directly transferred to the Android implementation of MapExplorer. The promising results achieved merited further exploration of this technique. In particular, the original implementation offered two significant opportunities for improvement, namely:

- The use of a more classical, repeatable and scalable (capable of identifying more complex contexts of use) learning process to identify the user’s current context; and
- The use of a dynamic adaptation approach, rather than the original static adaptation approach, where the level of adaptation employed takes into account the level of variation in the sensor data.

7.6.1 Training

Several researchers have investigated how accelerometer data can be used to train algorithms to distinguish and classify the user’s current activity (Li et al. 2010, Ravi et al. 2005, Bao and Initille 2004). Sitting, walking, running, cycling, stair climbing and even less common activities such as vacuuming and brushing teeth have been investigated. A wide variety of approaches from decision trees, to Hidden Markov Models and Naïve Bayesian classifiers
have been employed. For this research, the primary objective was to distinguish seated and walking contexts of use. A secondary objective was to employ a learning technique which would scale to allow recognition of more complex contexts of use. A neural network implementation was employed due to the power of this approach for solving classification problems (Zhang 2000, Engelbrecht 2006).

Several features of accelerometer data have been identified as useful training inputs for context of use classifiers, including (Bao and Initille 2004):

- Mean
- Standard Deviation
- Correlation
- Energy

Correlation is calculated between each pair of axes and can be expressed using Formula 7.1:

\[
\text{correlation} \ (x,y) = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y}
\]  

Where \(\text{cov}(x,y)\) is the covariance between accelerometer data measured relative to the x and y axes and \(\sigma_x\) and \(\sigma_y\) refer to the standard deviation along the x and y axes.

The energy of the accelerometer signal measures cyclical patterns in the accelerometer data and requires the calculation of Fast Fourier Transforms (FFTs). While this is useful in distinguishing complex activities, calculating FFTs is a relatively resource intensive task. For the purposes of distinguishing seated and walking tasks, the remaining three inputs (mean, standard deviation and correlation) were considered sufficient. The previous implementation of sensitivity adaptation in Chapter 5 relied only on variance (the square of the standard deviation). Incorporating correlation provides additional information which is useful to distinguish tilt gestures (which also result in standard deviation changes) from increases in standard deviation as a result of walking motion.

Other important variables to consider are the sampling window size to be used and whether or not to employ overlapping sampling windows. Previous research has successfully employed a window size of 256 samples (which corresponds to about 5 seconds at a sampling rate of 50Hz) and a 50% overlap between samples (Bao and Initille 2004, Ravi et al. 2005). This window was shortened to 150 samples (approximately 3 seconds) with 50% overlap.
between sampling windows to allow for a more responsive 1.5 second delay between each context of use classification.

A feed-forward Artificial Neural Network (ANN) was trained to classify the user’s current context of use as either seated or walking. The ANN was given ten inputs, consisting of the mean acceleration for the x, y and z axes during the sample window, the standard deviation for the three axes, the correlation between the three pairs of axes and an additional bias weight. A single hidden layer with five neurons was employed. The hidden layer size was determined through experimentation with varying hidden layer sizes to determine the configuration which would optimise classification accuracy. The ANN produced a single output, with 0 indicating a seated context of use and 1 indicating a walking context of use. A step activation function was used, with values below 0.5 rounded down to 0 and values above 0.5 rounded up to 1. Figure 7.5 shows the network configuration that was employed. The input layer is simplified to improve readability (inputs for different axes are not shown separately and bias weight not shown).

![Artificial Neural Network architecture used to distinguish walking and seated contexts of use. The input layer is simplified for readability.](image)

A genetic algorithm was used to learn the training weights. This approach was favoured over more traditional back propagation algorithms which may get stuck in a local optimum without finding the global optimum solution (Engelbrecht 2006). Genetic algorithms have been successfully used to train neural networks for classification problems and have been
shown to be efficient and comparatively easy to implement (Örkcü and Bal 2011, Gupta and Sexton 1999, Montana and Davis 1989). An initial population of 75 weight configurations was initialised with each weight randomly initialised to between -1 and 1. A cross-over rate of 0.85 and a mutation rate of 0.10 were selected after experimentation to determine the optimal genetic algorithm parameters.

Training data was collected from 9 participants using MapExplorer while seated and while walking. Participants were required to complete eight route-following tasks (four seated and four walking). Walking tasks were performed while walking up and down a 15 metre long corridor. Accelerometer output along all three axes was written to files to be used as training data for the neural network. The raw data was used to calculate mean, standard deviation and correlation values in sample sizes of 150. Walking speeds were recorded for walking tasks and an average walking speed of 2.19 km/h was observed. Walking speeds were fairly tightly grouped, with a minimum average speed of 1.72 km/h and a maximum speed of 2.75 km/h. Participants also performed two additional walking tasks, the first with a dampening factor (the factor by which the sensitivity of tilt interaction is reduced) of 2 and the second with a dampening factor of 3 to determine the appropriate dampening factor to employ. Accuracy results indicated that the lower dampening factor achieved little accuracy improvement, while a more noticeable accuracy improvement was achieved with a dampening factor of 3.

Individual patterns were randomly selected from each participant’s training data to test the accuracy of the ANN after training and optimise parameter settings. The genetic algorithm was trained through 500 generations using 75 randomly initialised training patterns (each containing a potential weight configuration for the ANN). Roulette wheel selection was employed to give fitter individuals a better chance of taking part in reproduction. Cross-over was performed by using a weighted sum (adding up to 1) of each parent’s weights. Mutation was performed by adding a random value in the range -0.5 to 0.5 to approximately 10% of the weights. The training process was repeated multiple times and the ANN’s performance with the test data (separate from the training data) was used to optimise parameter settings.

The result of the training phase was a set of 50 learned weights between the input and hidden layer (10 inputs * 5 hidden layer neurons) and a set of 5 weights between the hidden layer and output layer. The ANN achieved accuracy of 96% on the test set. The weights were transferred to MapExplorer and used to test the accuracy of the learned weights during actual use. Accuracy of greater than 95% was achieved during informal trials. This was further
improved by requiring two successive readings from the ANN to indicate state changes before a switch between states (e.g. seated to walking) is confirmed.

### 7.6.2 Static Sensitivity Adaptation

Static sensitivity adaptation was implemented similarly to the original sensitivity adaptation implementation presented in Chapter 5. Two sensitivity levels were employed, with one sensitivity level used for seated tasks and a reduced sensitivity level used for walking tasks. Sensitivity was decreased by a dampening factor of 3 for walking tasks, based on the preliminary tests that were conducted during the training phase. The original sensitivity adaptation implementation decreased sensitivity by widening the discretisation intervals employed (Section 5.2.4). Static sensitivity adaptation employed a different approach, decreasing sensitivity by dividing the panning speeds calculated using the standard discretisation intervals by a dampening factor of 3. This approach was preferred after analysing log data from the previous user study involving sensitivity adaptation. This analysis revealed that the majority of participants only used tilt angles in the first two discretisation intervals while walking. By adjusting the panning speed instead of the discretisation interval size, users retain finer-grained control over panning speed, although at a reduced level of sensitivity.

![Figure 7.6. Static sensitivity adaptation](image-url)
Sensitivity adaptation was implemented by sampling 150 accelerometer readings at a time (with a 50% overlap between sampling windows). This data was used to calculate mean and standard deviation values for all three axes and correlation values for all three pairs of axes. This data was then fed into the trained ANN (represented as two sets of weights) and used to determine the user’s current context of use. If the user is mobile, a dampening factor of 3.0 is used, otherwise a dampening factor of 1.0 (no dampening) is employed. This process is depicted in Figure 7.6.

### 7.6.3 Dynamic Sensitivity Adaptation

Static sensitivity adaptation employed a binary approach with different pre-defined sensitivity settings for seated tasks and walking tasks. In order to determine whether a more flexible approach could achieve better results, static sensitivity adaptation was modified to make use of dynamic dampening factors. Data recorded during the training phase was used to determine baseline values for standard deviation for the three accelerometer axes while users are seated. Standard deviations were calculated for all three axes in sample windows of 20 observations. This relatively short window (which corresponds to about half a second) was chosen in order to ensure that the adaptation is very responsive to changes in walking speed and other external sources of variability (e.g. stair climbing). Sliding windows were used with one observation discarded each time a new observation was received from the sensors. Mean standard deviations of 0.39, 0.28 and 0.45 were recorded for the x, y and z axes accelerations (where acceleration is measured in metres/second$^2$) for all participants in the training phase.

Dynamic sensitivity adaptation was implemented identically to static sensitivity adaptation, except for the dampening factor employed. Static sensitivity adaptation employs a fixed dampening factor of 3 when the user is mobile. Dynamic sensitivity adaptation calculates the dampening factor when the user is mobile as the maximal ratio between the currently observed acceleration standard deviations and the baseline values calculated from the training data. The ratios are continually calculated for all three axes while the application is in use and the largest ratio is used to calculate the dampening factor. In this way, if the user is walking quickly, greater sensitivity adaptation will take place, whereas if the user is walking slowly, only minor sensitivity adaptation will take place. Table 7.2 illustrates how the dampening factor is calculated using dummy acceleration data.
Chapter 7: Improving the Controllability of Tilt Interaction

<table>
<thead>
<tr>
<th>Acceleration Std. Dev.</th>
<th>Baseline Value</th>
<th>Ratio (Acceleration/Baseline)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-Axis</td>
<td>0.78</td>
<td>0.39</td>
</tr>
<tr>
<td>Y-Axis</td>
<td>0.35</td>
<td>0.28</td>
</tr>
<tr>
<td>Z-Axis</td>
<td>0.65</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 7.2. Example of dampening factor calculation using dynamic adaptation. The largest ratio (x-axis) shown in bold is used as the current dampening factor in this example.

The following section describes a user study which was conducted to compare the modified zooming techniques and sensitivity adaptation implementations.

### 7.7 Experimental Design

A user study was conducted in order to evaluate whether the above changes resulted in an improvement in the controllability of tilt interaction.

#### 7.7.1 Objectives

The objectives of this user study were as follows:

- To comparatively evaluate the gesture and tilt zooming techniques; and
- To determine whether controllability of tilt interaction while walking can be improved by automatically adjusting the sensitivity according to whether the user is seated or walking.

#### 7.7.2 Hypotheses

In order to achieve the above objectives, several hypotheses are proposed below. The results of the user study will be used to determine whether there is sufficient evidence to reject each of the following null hypotheses relating to tilt zooming:

- $H_{0,1Z}$: Tilt zooming does not exhibit better perceived controllability than gesture zooming.
- $H_{0,2Z}$: Tilt zooming is not more efficient than gesture zooming.
- $H_{0,3Z}$: Tilt zooming does not exhibit better perceived workload than gesture zooming.
- $H_{0,4Z}$: Participants do not find the overview feature useful.
- $H_{0,5Z}$: Participants do not prefer tilt zooming over gesture zooming.
The following null hypotheses relating to sensitivity adaptation will be tested:

- $H_{0,1S}$: Static sensitivity adaptation does not exhibit better controllability than no adaptation.
- $H_{0,1D}$: Dynamic sensitivity adaptation does not exhibit better controllability than no adaptation.
- $H_{0,2S}$: Static sensitivity adaptation is not less efficient than no adaptation.
- $H_{0,2D}$: Dynamic sensitivity adaptation is not less efficient than no adaptation.
- $H_{0,3S}$: Static sensitivity adaptation does not exhibit better perceived controllability than no adaptation.
- $H_{0,3D}$: Dynamic sensitivity adaptation does not exhibit better perceived controllability than no adaptation.

Sensitivity adaptation reduces the sensitivity of tilt interaction, which could potentially result in decreased efficiency. The hypotheses relating to the efficiency of the two sensitivity adaptation approaches are therefore designed to test whether there is sufficient evidence to indicate a decrease in efficiency associated with the use of static and dynamic sensitivity adaptation.

### 7.7.3 Metrics

In order to test the above hypotheses and achieve the objectives of this user study, a variety of objective and subjective metrics were recorded. Efficiency data was the only performance metric evaluated for zooming techniques. Efficiency data was collected using logging functionality built into MapExplorer. Similar self-reported metrics were collected as in the user studies described in Chapters 4 and 5. Participants were asked to rate the perceived workload of both zooming techniques using the six workload measures of the NASA-TLX questionnaire (Hart and Staveland 1988). Perceived efficiency, effectiveness, ease of use and controllability metrics were also collected. Participants were also asked to rate the usefulness of the overview function on a 7 point scale. The post-task questionnaire was therefore a combination of the NASA-TLX questionnaire and a modified After-Scenario Questionnaire (ASQ) (Lewis 1991). The post-task questionnaire is provided in Appendix H. Participants were also asked to complete a post-test questionnaire in which they were required to rate their preferred zooming technique while seated, walking and overall (Appendix I). Participants
were also required to identify one positive and one negative aspect of each zooming technique.

Data logging was built into MapExplorer to calculate efficiency and accuracy data for the different sensitivity adaptation settings. Accuracy data was collected to determine how accurately participants were able to control tilt interaction using the different sensitivity adaptation settings. Efficiency data was collected to determine whether sensitivity adaptation negatively affected the participants’ task times. Walking speed was measured to determine whether participants changed their walking speed to compensate for difficulties in controlling tilt interaction (Fitchett and Cockburn 2009). Perceived efficiency, effectiveness, ease of use and controllability metrics were measured using a post-task questionnaire (Appendix J).

7.7.4 Participants

Thirty participants (21 male, 9 female) took part in the user study. All participants were students in the Department of Computing Sciences at the Nelson Mandela Metropolitan University (NMMU). Two participants were left-handed. All participants had at least six years of mobile phone experience. Twenty-two participants reported at least occasional use of mobile map-based applications. Twenty participants had some prior experience with tilt interaction, although in most cases this was limited use of mobile games.

7.7.5 Tasks

The user study was divided into two main sections. The first section involved comparing gesture zooming and tilt zooming. The second section involved comparing three different sensitivity adaptation settings, namely no adaptation, static adaptation and dynamic adaptation. A total of 25 tasks were performed, divided as follows:

- Zooming tasks:
  - 4 seated tasks with gesture zooming;
  - 4 walking tasks with gesture zooming;
  - 4 seated tasks with tilt zooming; and
  - 4 walking tasks with tilt zooming.

- Sensitivity adaptation:
  - 3 navigating tasks with no sensitivity adaptation;
  - 3 navigating tasks with static sensitivity adaptation; and
  - 3 navigating tasks with dynamic sensitivity adaptation.
Chapter 7: Improving the Controllability of Tilt Interaction

The order in which the zooming techniques and sensitivity adaptation settings were used was counterbalanced in order to offset any learning effects. Different task sets were used and counterbalanced across participants in order to avoid any learning effect as a result of the task sets used in conjunction with each technique. For zooming tasks, all participants performed the seated tasks before the walking tasks to allow them to get accustomed to the techniques while seated before commencing with the walking tasks.

Figure 7.7. Example zooming task in MapExplorer

Zooming tasks were designed to ensure that zooming played a significant role in the tasks (Figure 7.7). Tasks either involved zooming in or out, with the last task in each set of four requiring participants to pan to a distant icon and then zoom out to find the nearest POI in a particular category. The overview feature was activated for the last task in each set of four. Suggested zoom levels were included in the task instructions. The navigating tasks used for sensitivity adaptation required the user to follow a pre-planned route on the screen from start to finish. The original route is shown in blue, with successfully completed segments changing to green and unsuccessfully completed segments turning red to encourage participants to stay on the route. Navigating tasks were selected to allow for easy measurement of accuracy data (Figure 7.8).
Chapter 7: Improving the Controllability of Tilt Interaction

7.7.6 Independent Variables

A within-subjects approach was used for all test settings. The following independent variables were changed during the zooming technique portion of the user study:

- **Zooming Technique**: Participants were required to use both gesture zooming and tilt zooming. Both techniques required users to depress a clutch button for the duration of the zooming gesture (either a physical button or a touch-screen button). The order in which the zooming techniques were used was counterbalanced to offset any learning effects.

- **Overview**: The use of an overview in the top right corner of the display was investigated in conjunction with both zooming techniques. The overview was only activated for the last task in each context of use.

- **Context of use**: Participants were required to complete both seated and walking tasks with each zooming technique.

- **Task set**: Two task sets were used which were designed to be as similar as possible. The order in which these were used was counterbalanced across participants.

The following independent variables were changed during the sensitivity adaptation portion of the user study:
Chapter 7: Improving the Controllability of Tilt Interaction

- **Sensitivity adaptation setting:** Participants were required to use three different sensitivity adaptation settings.
  - *No adaptation:* Using the same sensitivity when walking as when seated.
  - *Static sensitivity adaptation:* Using a constant dampening factor when it is determined that the participant is mobile.
  - *Dynamic sensitivity adaptation:* Using a dynamic dampening factor when it is determined that the participant is mobile.

- **Task set:** Three similar task sets were used and counterbalanced across participants.

### 7.7.7 Test Environment

The user study was conducted in a laboratory environment in the Department of Computing Sciences at NMMU. Walking tasks were performed in an indoor corridor with a mixture of artificial and natural lighting.

### 7.7.8 Materials

A Samsung Google Nexus S running Android version 2.3.4 with MapExplorer installed was used in the user study.

### 7.7.9 Procedure

The procedure for the user study was as follows:

1. The participant completed an informed consent form and biographical questionnaire (Appendices A and B).
2. For each zooming technique:
   a. The participant was given an overview of the system and the zooming technique was demonstrated. The participant was then given opportunity to experiment with this technique.
   b. The participant used the zooming technique to perform 4 seated tasks.
   c. The participant completed the NASA-TLX and user satisfaction questionnaires (Appendix H)
   d. The participant used the zooming technique to perform 4 walking tasks.
   e. The participant completed the NASA-TLX and user satisfaction questionnaires.
3. The participant completed the post-test questionnaire rating his/her preferred zooming technique while seated and walking and also provided any general comments (Appendix I).
4. The participant was given an overview of route-following tasks and was allowed to complete one training route-following task.

5. For each sensitivity adaptation setting:
   a. The participant completed three navigation tasks using the sensitivity adaptation setting.
   b. The participant completed the post-task user satisfaction questionnaire (Appendix J). A general comments section was included after the final group of tasks.

7.8 Results: Zooming Techniques

This section presents the results of the first section of the user study, in which gesture zooming was compared to tilt zooming. Objective results are discussed first, followed by subjective results.

7.8.1 Efficiency

Figure 7.9 shows the mean task times for the two zooming techniques. Tilt zooming was more efficient for both seated and walking contexts of use.

![Figure 7.9. Mean task times for gesture and tilt zooming techniques. Error bars show 95% confidence intervals (n=30).](image)

T-tests for statistical significance showed that the difference between the mean task times for the two zooming techniques was statistically significant for both the seated (p = 0.020) and walking (p = 0.047) contexts of use (Table 7.3). The difference between the mean overall task
times was also statistically significant ($p = 0.017$). Cohen’s $d$ values showed a small practical significance in all three cases.

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Tilt</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
</tr>
<tr>
<td>Seated</td>
<td>178.25</td>
<td>45.03</td>
</tr>
<tr>
<td>Walking</td>
<td>156.66</td>
<td>49.43</td>
</tr>
<tr>
<td>Overall</td>
<td>334.91</td>
<td>79.71</td>
</tr>
</tbody>
</table>

Table 7.3. Results of t-tests comparing mean task times (in seconds) for gesture and tilt zooming ($n=30$)

### 7.8.2 Workload

Mean perceived workload ratings for seated tasks are shown in Figure 7.10. Perceived workload was lower (better) for all 6 workload metrics for tilt zooming.

![Figure 7.10. Mean perceived workload ratings for seated tasks. Errors bars show 95% confidence intervals ($n=30$).](image)

T-tests for statistical significance showed that statistically significant differences were recorded between the mean ratings for gesture and tilt zooming for four of the six workload metrics (Table 7.4). Mental demand ($p = 0.009$), physical demand ($p < 0.001$), effort ($p = 0.001$) and frustration ($p = 0.004$) were all found to be significantly lower (better) for tilt zooming than for gesture zooming. Cohen’s $d$ showed medium practical significance for all of these except physical demand, for which a large practical significance was calculated.
Chapter 7: Improving the Controllability of Tilt Interaction

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Tilt</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean  SD</td>
<td>Mean  SD  t (29)  p-value</td>
</tr>
<tr>
<td>Mental Demand</td>
<td>3.37 1.65</td>
<td>2.40 1.22  -2.81  0.009</td>
</tr>
<tr>
<td>Physical Demand</td>
<td>3.53 1.70</td>
<td>2.13 1.14  -5.28 &lt;0.001</td>
</tr>
<tr>
<td>Temporal Demand</td>
<td>2.93 1.51</td>
<td>2.47 1.22  -1.37  0.182</td>
</tr>
<tr>
<td>Performance</td>
<td>2.40 1.22</td>
<td>2.17 1.49  -1.27  0.214</td>
</tr>
<tr>
<td>Effort</td>
<td>3.53 1.74</td>
<td>2.57 1.19  -3.59  0.001</td>
</tr>
<tr>
<td>Frustration</td>
<td>2.83 1.82</td>
<td>1.90 1.21  -3.08  0.004</td>
</tr>
</tbody>
</table>

Table 7.4. Results of t-tests comparing mean perceived workload ratings for seated tasks (n=30)

Figure 7.11 shows the mean perceived workload ratings given by participants after the walking set of tasks. Perceived workload was lower (better) for tilt zooming for all six workload metrics.

![Perceived Workload - Walking Tasks](image)

Figure 7.11. Perceived workload ratings - walking tasks (n=30)

The results of t-tests for statistical significance are shown in Table 7.5. Only the differences for mental demand (p = 0.019) and physical demand (p = 0.002) were statistically significant, with the difference for effort narrowly not statistically significant (p = 0.059). Cohen’s d shows a small practical significance for mental demand and a medium practical significance for physical demand.
Chapter 7: Improving the Controllability of Tilt Interaction

<table>
<thead>
<tr>
<th></th>
<th>Gesture Mean</th>
<th>Tilt Mean</th>
<th>Significance t (29)</th>
<th>p-value</th>
<th>Cohen’s d</th>
<th>Practical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mental Demand</td>
<td>3.30</td>
<td>2.63</td>
<td>-2.48</td>
<td><strong>0.019</strong></td>
<td>-0.45</td>
<td>Small</td>
</tr>
<tr>
<td>Physical Demand</td>
<td>3.50</td>
<td>2.70</td>
<td>-3.45</td>
<td><strong>0.002</strong></td>
<td>-0.63</td>
<td>Medium</td>
</tr>
<tr>
<td>Temporal Demand</td>
<td>2.73</td>
<td>2.63</td>
<td>-0.36</td>
<td>0.725</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>Performance</td>
<td>2.53</td>
<td>2.40</td>
<td>-0.61</td>
<td>0.546</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>Effort</td>
<td>3.37</td>
<td>2.77</td>
<td>-1.96</td>
<td>0.059</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>Frustration</td>
<td>2.63</td>
<td>2.23</td>
<td>-1.53</td>
<td>0.136</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
</tbody>
</table>

Table 7.5. Results of t-tests comparing mean perceived workload ratings for walking tasks (n=30)

7.8.3 User Satisfaction

Figure 7.12 shows the mean user satisfaction ratings for gesture and tilt zooming for seated tasks. User satisfaction was higher for all four measures for tilt zooming.

![User Satisfaction - Seated Tasks](image)

Figure 7.12. Mean user satisfaction ratings for zooming techniques for seated tasks (n=30)

Table 7.6 shows the results of t-tests comparing the mean user satisfaction ratings for seated tasks. The difference in mean values for the two zooming techniques was statistically significant for all four metrics with p-values below 0.001. Cohen’s d values indicated a medium practical significance for effectiveness and large practical significance for efficiency, ease of use and controllability.
### Table 7.6. Results of t-tests comparing mean user satisfaction ratings for zooming tasks while seated (n=30)

<table>
<thead>
<tr>
<th></th>
<th>Gesture</th>
<th>Tilt</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>SD</td>
<td>Mean</td>
</tr>
<tr>
<td>Effectiveness</td>
<td>4.87</td>
<td>1.81</td>
<td>6.27</td>
</tr>
<tr>
<td>Efficiency</td>
<td>4.87</td>
<td>1.81</td>
<td>6.17</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>4.70</td>
<td>1.78</td>
<td>6.43</td>
</tr>
<tr>
<td>Controllability</td>
<td>4.70</td>
<td>1.49</td>
<td>6.07</td>
</tr>
</tbody>
</table>

Figure 7.13 shows the mean user satisfaction ratings for each zooming technique for walking tasks. The mean user satisfaction ratings were greater for tilt zooming for all four user satisfaction metrics. The differences were smaller for walking tasks than for seated tasks. A statistically significant decrease in ease of use was observed for the tilt zooming technique between the seated and walking contexts of use (t(29) = 3.08, p = 0.004).

![User Satisfaction - Walking Tasks](image)

Figure 7.13. Mean user satisfaction ratings for walking tasks (n=30)

Table 7.7 shows the results of t-tests comparing the mean user satisfaction ratings for the two zooming techniques for walking tasks. Statistically significant differences were recorded for effectiveness (p = 0.002), ease of use (p = 0.010) and controllability (p = 0.022). Cohen’s d values showed that the difference in mean ratings for effectiveness and ease of use were of medium practical significance and that the difference for controllability was of small practical significance.
Table 7.7. Results of t-tests comparing mean user satisfaction ratings for zooming techniques for walking tasks (n=30)

Participants were also asked to indicate in the post-task questionnaire whether or not they found the overview feature useful (using a seven point scale where 1 = “not useful” and 7 = “very useful”). The overview was only enabled for the last task in each group of four. Table 7.8 shows the results of t-tests comparing the mean values to the intermediate value of 4. For all four test conditions, the mean rating was greater than 4, and this difference was statistically significant in all four conditions. Prior to correcting the data for counterbalancing it was noticed that the mean rating for the overview increased for each of the four test conditions. Participants therefore tended to give higher ratings as they completed more tasks with the overview.

Table 7.8. Results of t-tests comparing mean overview usefulness ratings to intermediate value of 4 (n=30)

7.8.4 Preferences

Figure 7.14 shows the mean ratings given by participants regarding zooming technique preference. Participants were asked to rate their preferred technique on a 7 point scale (where 1 = gesture zooming and 7 = tilt zooming). The results show a clear preference for tilt zooming for all three questions.
T-tests for statistical significance were conducted to determine whether the mean ratings were significantly different from the neutral rating of 4 (Table 7.9). The results show that the ratings for seated (p < 0.001), walking (p < 0.001) and overall preference (p < 0.001) were all statistically significant. Cohen’s d showed large practical significance for seated tasks and overall and medium practical significance for walking tasks. Participants were also asked to rate their preferred method for activating zooming on a 7 point scale (where 1 = touch screen and 7 = physical button). Participants indicated a statistically significant preference for touch-screen activation (p < 0.001). Cohen’s d indicated a large practical significance for this result.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>5.77</td>
</tr>
<tr>
<td>SD</td>
<td>1.48</td>
</tr>
<tr>
<td>t (29)</td>
<td>6.55</td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Cohen’s d</td>
<td>1.20</td>
</tr>
<tr>
<td>Practical</td>
<td>Large</td>
</tr>
<tr>
<td>Walking</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>5.27</td>
</tr>
<tr>
<td>SD</td>
<td>1.86</td>
</tr>
<tr>
<td>t (29)</td>
<td>3.74</td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Cohen’s d</td>
<td>0.68</td>
</tr>
<tr>
<td>Practical</td>
<td>Medium</td>
</tr>
<tr>
<td>Seated</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>5.67</td>
</tr>
<tr>
<td>SD</td>
<td>1.67</td>
</tr>
<tr>
<td>t (29)</td>
<td>5.47</td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Cohen’s d</td>
<td>1.00</td>
</tr>
<tr>
<td>Practical</td>
<td>Large</td>
</tr>
<tr>
<td>Zooming Activation</td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>2.13</td>
</tr>
<tr>
<td>SD</td>
<td>1.81</td>
</tr>
<tr>
<td>t (29)</td>
<td>-5.64</td>
</tr>
<tr>
<td>p-value</td>
<td>&lt;0.001</td>
</tr>
<tr>
<td>Cohen’s d</td>
<td>1.03</td>
</tr>
<tr>
<td>Practical</td>
<td>Large</td>
</tr>
</tbody>
</table>

Table 7.9. Results of t-tests for statistical significance for user preference ratings (n=30)

### 7.8.5 Qualitative Feedback

Positive feedback regarding gesture zooming focused on the fact that users liked the simplicity of the discrete nature of this technique, where zooming gestures resulted in a single zoom level change at a time. Several participants commented that they found gesture...
zooming easy to control. Participants found that gesture zooming allowed them to control the zoom level without requiring their full attention. Several participants felt that this technique was easier to use while walking than while seated. Table 7.10 summarises the frequency of the main positive themes reported regarding gesture zooming.

<table>
<thead>
<tr>
<th>Theme</th>
<th>n</th>
<th>Example Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete nature</td>
<td>7</td>
<td>“Forces simple one zoom out/in at a time”</td>
</tr>
<tr>
<td>Easy to control</td>
<td>5</td>
<td>“I felt a bit more in control”</td>
</tr>
<tr>
<td>Ease of use while walking</td>
<td>5</td>
<td>“Easy to use while walking”</td>
</tr>
</tbody>
</table>

Table 7.10. Frequency of most common positive themes regarding gesture zooming

Controllability problems were the most frequently cited negative theme regarding gesture zooming (Table 7.11). Several participants stated that they were unsure exactly how far or how quickly to move the phone to activate zooming. Participants also complained that gesture zooming was too slow, particularly when moving through several zoom levels. Gesture zooming was also described as being less responsive. Several participants found gesture zooming difficult to use while seated. Reasons cited for this include the fact that participants found their movement restricted while seated.

<table>
<thead>
<tr>
<th>Theme</th>
<th>n</th>
<th>Example Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllability</td>
<td>9</td>
<td>“Difficult to gauge how much the phone needs to be moved to zoom”</td>
</tr>
<tr>
<td>Slow</td>
<td>6</td>
<td>“Takes too long. Requires too many motions.”</td>
</tr>
<tr>
<td>Difficult to use while seated</td>
<td>4</td>
<td>“It’s not as easy to use when seated and space is limited”</td>
</tr>
</tbody>
</table>

Table 7.11. Frequency of most common negative themes regarding gesture zooming

Positive feedback regarding tilt zooming is summarised in Table 7.12. Participants frequently cited tilt zooming as being “natural” and “smooth”. Participants found the technique intuitive and quick to use. Participants also liked the continuous nature of tilt zooming which allowed them to adjust the zoom level by multiple levels in a single movement. Tilt zooming was also described as easy to control.
Chapter 7: Improving the Controllability of Tilt Interaction

<table>
<thead>
<tr>
<th>Theme</th>
<th>n</th>
<th>Example Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intuitive</td>
<td>8</td>
<td>“Seemed intuitive and simple to use”</td>
</tr>
<tr>
<td>Smoothness/Speed</td>
<td>7</td>
<td>“Smooth, fast response from zooming”</td>
</tr>
<tr>
<td>Controllability</td>
<td>5</td>
<td>“Easy to control zooming as it requires less effort”</td>
</tr>
</tbody>
</table>

Table 7.12. Frequency of the most common positive themes regarding tilt zooming

Frequency information regarding the most common negative themes regarding tilt zooming is summarised in Table 7.13. Six participants did not say anything negative about tilt zooming. Negative comments focused on the fact that tilt zooming was difficult to control when zooming at high speeds and that overshooting problems could occur. Some participants also felt that more effort was required to monitor their current zoom level because of the continuous nature of tilt zooming. Three participants commented that tilt zooming was comparatively difficult to use while walking.

<table>
<thead>
<tr>
<th>Theme</th>
<th>n</th>
<th>Example Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllability at fast speeds</td>
<td>5</td>
<td>“If the phone is tilted too far can zoom too fast and past your desired level”</td>
</tr>
<tr>
<td>Awareness of zoom level</td>
<td>4</td>
<td>“Making sure that you on the right zoom level”</td>
</tr>
<tr>
<td>Difficult to use while walking</td>
<td>3</td>
<td>“It is harder to walk and use this technique”</td>
</tr>
</tbody>
</table>

Table 7.13. Frequency of the most common negative themes regarding tilt zooming

The following section presents the results of the second part of the evaluation, in which different approaches to sensitivity adaptation were compared.

7.9 Results: Sensitivity Adaptation

All participants used three sensitivity adaptation settings, namely no adaptation, static adaptation and dynamic adaptation. The objective metric results are presented first, followed by the subjective results.

7.9.1 Accuracy

Accuracy for the navigation tasks was calculated as the mean Euclidean distance between the planned route and the actual route followed by the user (in pixels). Figure 7.15 shows the mean accuracy for each of the three sensitivity adaptation settings. Greater accuracy was achieved for both static and dynamic adaptation than for the no adaptation setting. Very
similar accuracy was achieved for both adaptation techniques. A repeated measures ANOVA test showed a significant accuracy difference between the three settings ($F = 20.95$, $p < 0.001$). Tukey post-hoc tests were then conducted to compare the individual pairs of settings (Table 7.14). Static adaptation ($p < 0.001$) and dynamic adaptation ($p < 0.001$) were both significantly more accurate than no adaptation. The difference between static and dynamic adaptation was not statistically significant. Cohen’s $d$ showed a large practical significance for the difference between static adaptation and no adaptation and the difference between dynamic adaptation and no adaptation.

![Figure 7.15. Mean deviation from planned routes (in pixels) ($n=30$)](image)

| Static vs. No Adaptation | -5.04 | 5.55 | <0.001 | -0.91 | Large |
| Dynamic vs. No Adaptation | -5.24 | 5.44 | <0.001 | -0.96 | Large |
| Dynamic vs. Static Adaptation | -0.20 | 3.95 | 0.783 | n.a. | n.a. |

Table 7.14. Results of Tukey HSD post-hoc tests comparing accuracy of individual pairs of sensitivity adaptation settings ($n=30$)

Figure 7.16 shows overlaid plots for all participants for one of the routes used during the user study. Due to counterbalancing, ten participants used each sensitivity adaptation setting on each route. The diagrams show the planned route in blue, and the participants’ actual routes in red. The plots clearly illustrate how participants using static and dynamic adaptation were able to more accurately follow the planned routes.
7.9.2 Efficiency

Efficiency was very similar for all three settings (Figure 7.17). Despite the lower sensitivity of the two adaptation techniques only slightly slower task times were reported. Dynamic adaptation was slightly more efficient than static adaptation. A repeated measures ANOVA test revealed no significant difference between the three settings ($F(2,58) = 1.38, p = 0.26$).

Figure 7.17. Mean task times for different sensitivity adaptation settings ($n=30$)
### 7.9.3 User Satisfaction

Participants also completed a post-task user satisfaction questionnaire after using each sensitivity adaptation setting. Static adaptation was rated ahead of dynamic adaptation which in turn was rated ahead of no adaptation for all four user satisfaction measures (Figure 7.18).

![Mean User Satisfaction Ratings - Sensitivity Adaptation](image)

**Figure 7.18. Mean user satisfaction ratings for sensitivity adaptation. 95% confidence intervals shown (n=30).**

Repeated measures ANOVA tests showed significant differences for effectiveness ($F(2,58) = 5.44, p = 0.006$), ease of use ($F(2,58) = 3.18, p = 0.048$) and controllability ($F(2,58) = 5.63, p = 0.005$).

Tukey post-hoc tests were then conducted to compare the mean ratings for effectiveness, ease of use and controllability for the individual pairs of sensitivity adaptation settings. The results showed a significant difference between static adaptation and no adaptation for effectiveness ($p = 0.003$), ease of use ($p = 0.040$) and controllability ($p = 0.002$). Cohen’s $d$ showed a medium practical significance for effectiveness and controllability and a small practical significance for ease of use. The differences between dynamic adaptation and no adaptation were not significant, as were the differences between dynamic adaptation and static adaptation. Tables 7.15 – 7.17 show the Tukey post-hoc test results for effectiveness, ease of use and controllability.
### Chapter 7: Improving the Controllability of Tilt Interaction

#### 7.9.4 Qualitative Feedback

General comments largely mirrored participants’ user satisfaction ratings. Most participants were able to clearly perceive the difference between static adaptation and no adaptation and the difference between dynamic adaptation and no adaptation. Many participants commented that the no adaptation setting was too sensitive and difficult to control. Many participants commented that they liked both static and dynamic adaptation, but disliked no adaptation. While most participants commented that they liked dynamic adaptation, some complained that this technique was not sensitive enough and (incorrectly) thought that extreme tilt angles were required to get the cursor to move faster. A small minority of participants that were very comfortable using tilt interaction preferred the more sensitive no adaptation technique. A few participants commented that they felt that the effects of their tilt actions were difficult to predict when using the dynamic adaptation technique.

#### 7.10 Objective vs. Subjective Data

Mean walking speeds were recorded for the three sensitivity adaptation settings by recording participants total task times and the number of times the 15 metre course was completed.
when performing each set of three navigation tasks. Very similar mean walking speeds were recorded for all three settings, with an average walking speed of 2.31 km/h for no adaptation tasks, 2.38 km/h for static adaptation tasks and 2.37 km/h for dynamic adaptation tasks. Pearson product moment correlation co-efficients were calculated between walking speed and the accuracy achieved using each sensitivity adaptation setting (Figures 7.19 - 7.21). Positive correlation co-efficients of 0.38, 0.05 and 0.44 were recorded for no adaptation, static adaptation and dynamic adaptation. The correlations for the no adaptation and dynamic adaptation settings were statistically significant, indicating that for both settings, faster walking speeds negatively affected accuracy.

![Walking Speed vs. Accuracy: No Adaptation](image)

**Figure 7.19. Correlation between walking speed and accuracy for no adaptation (n=30)**

Closer inspection of the data for the static adaptation condition showed that a group of participants who walked particularly slowly achieved the lowest accuracy. This is somewhat surprising given the fact that slower walking speeds would be expected to result in better accuracy. Analysis of log files showed that for some of these participants (particularly the slowest four), their walking speeds were at times too slow (below 1.5 km/h) to trigger sensitivity adaptation because the system failed to recognise their current context of use as walking. This is likely due to the fact that during the training phase, the slowest walking speed encountered was 1.72 km/h (with a mean speed of 2.18 km/h). Dynamic adaptation was not as affected by this problem, as only one participant recorded a mean speed of below 1.5 km/h for this setting.
Correlations between walking speed and efficiency showed negative correlations for all three sensitivity adaptation settings, with the correlations for static adaptation (-0.45) and dynamic adaptation (-0.34) statistically significant. Participants walking more quickly therefore also completed the navigation tasks more quickly using sensitivity adaptation.

Table 7.18 shows correlations between the subjective ratings given by participants and their actual performance with the different sensitivity adaptation settings. Correlations for
efficiency were small and not statistically significant. For perceived accuracy, all three settings were negatively correlated with actual accuracy data (higher perceived accuracy = lower mean error). This difference was statistically significant for no adaptation and static adaptation, but not for dynamic adaptation. This would indicate that participants did not perceive dynamic adaptation to be as accurate as it actually was.

<table>
<thead>
<tr>
<th>Efficiency: Perceived vs. Actual</th>
<th>Accuracy: Perceived vs. Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Adaptation</td>
<td>-0.18</td>
</tr>
<tr>
<td>Static Adaptation</td>
<td>-0.05</td>
</tr>
<tr>
<td>Dynamic Adaptation</td>
<td>-0.11</td>
</tr>
</tbody>
</table>

Table 7.18. Correlations between perceived and actual sensitivity adaptation efficiency and accuracy \( (n=30) \)

### 7.11 Dynamic Adaptation

In order to understand the differences between the results for static and dynamic adaptation, it is necessary to examine the output from the log files to determine to what degree dynamic adaptation actually adapted the sensitivity of tilt interaction. Analysis of log files showed that on average, dynamic adaptation employed a dampening factor of 2.59, in contrast to static adaptation which employed a constant dampening factor of 3.00. However, as shown in Figure 7.22, dynamic adaptation occasionally resulted in dampening factors of above 3.00. This is because dynamic adaptation was designed to employ a dampening factor relative to the currently observed level of variation in the accelerometer output. Furthermore, Figure 7.22 illustrates that there was a fairly high level of volatility in the dampening factors employed by dynamic adaptation.
7.12 Discussion

The following section will analyse the results of the user study in more detail. Zooming technique results will be discussed first, followed by the sensitivity adaptation results.

7.12.1 Zooming Techniques

Tilt zooming proved to be more efficient than gesture zooming for seated and walking tasks (Table 7.3). This result is perhaps not surprising, given that gesture zooming required multiple discrete zooming gestures to move through multiple zoom levels, whereas tilt zooming allows multiple zoom levels to be traversed with a single gesture. User satisfaction ratings supported this data for seated tasks, with tilt zooming rated as significantly more efficient than gesture zooming (Table 7.6). For walking tasks, however, tilt zooming was only rated slightly more efficient than gesture zooming (Table 7.7). This seems to have been caused by a combination of participants feeling more comfortable with gesture zooming during walking tasks and comparatively less confident with tilt zooming during walking tasks than during seated tasks. Efficiency was also the second most frequently cited negative theme in qualitative feedback regarding gesture zooming, while it was the second most frequently cited positive theme for tilt zooming (Tables 7.11 and 7.12).

Mean ratings for perceived workload were all in favour of tilt zooming, with participants clearly feeling more at ease with this zooming technique (Tables 7.4 and 7.5). More
statistically significant measures were recorded for seated tasks (mental demand, physical demand, effort and frustration) than for walking tasks (mental demand and physical demand). For all six workload metrics, the gap between the two techniques narrowed for walking tasks. The reasons for this seem to have been a combination of participants feeling more comfortable with gesture zooming while walking and participants visibly struggling a little more with tilt zooming while walking.

User satisfaction was significantly in favour of tilt zooming for all four metrics for seated tasks. Mean ratings for effectiveness, efficiency, ease of use and controllability were all significantly higher for tilt zooming (Table 7.6). For walking tasks, all four measures were still in favour of tilt zooming, with the differences for effectiveness, ease of use and controllability statistically significant (Table 7.7). A statistically significant drop in ease of use was recorded for tilt zooming between seated and walking tasks. This would suggest that tilt zooming, like tilt-controlled panning, could also benefit from sensitivity adaptation when the user is mobile. Some participants mentioned that tilt zooming was difficult to control at high speeds, indicating that a linear mapping between pitch angles and zooming speeds may be more appropriate. It is noteworthy that while all four mean user satisfaction ratings for tilt zooming decreased for walking tasks, all four mean ratings increased for gesture zooming. This is supported by the fact that several participants commented that they found gesture zooming to be easier to use while walking, without the space restrictions encountered while seated at a desk (Tables 7.10 and 7.11).

Preference ratings were clearly in favour of tilt zooming for seated and walking tasks, as well as overall (Table 7.9). The preference for tilt zooming was slightly clearer for seated tasks. Out of 30 test participants, only four indicated an overall preference for gesture zooming (with one neutral response), indicating that tilt zooming was clearly perceived as the preferred technique. While a few participants liked the reassuring simplicity of moving discretely one zoom level at a time, most participants preferred the continuous control offered by tilt zooming. The adjectives “natural” and “smooth” were frequently used to describe tilt zooming (Table 7.12). Participants also liked the speed control and ability to easily reverse zooming direction. Participants were able to easily switch between panning and zooming using the tilt zooming technique.

Unlike the user study in Chapter 5, the majority of participants were able to quickly become accustomed to performing gesture zooming. Occasional gesture recognition errors did occur,
but the frequency of these errors was far lower than during the previous user study, indicating that the implementation changes were successful in improving gesture recognition. Several participants felt that the mapping between gesture direction and zooming direction should be reversed, particularly for gesture zooming. As previous research has indicated, this appears to be related to whether users perceive themselves to be moving a physical map, or moving a camera relative to a map.

The use of the overview was positively received by participants. Mean ratings revealed that participants found the overview to be useful in conjunction with both zooming techniques (Table 7.8). It was noticeable that for tasks that required the participants to find a POI far away from their starting position, many participants actively made use of the overview to locate the POI. Participants then switched their attention back to the normal view to select the POI icon once it came into view.

Participants indicated a clear preference for the use of a touch-screen button over a hardware button as a clutch mechanism for activating zooming (Table 7.9). Twenty-seven out of 30 participants (with 3 neutral ratings) indicated a preference for the touch-screen button. This result, however, is probably somewhat misleading, given that the hardware buttons employed (volume buttons) proved difficult or uncomfortable to reach for some participants. A purpose-designed and positioned button might provide very different results.

**7.12.2 Sensitivity Adaptation**

Static and dynamic adaptation were both shown to allow participants to perform navigation tasks significantly more accurately than the standard, no adaptation setting (Table 7.14). A mean accuracy improvement of greater than 40% was achieved for both techniques. Plots of the paths followed by users overlaid on the actual route clearly show that participants struggled to control tilt interaction using the no adaptation setting (Figure 7.16). Plots for both adaptation techniques show that participants were able to follow the routes with a much higher degree of accuracy. Even greater accuracy improvements may have been achieved, were it not for the fact that in a few cases, sensitivity adaptation was not activated for the entire task duration due to participants walking slower than the neural network was trained to recognise. However, at such slow walking speeds, users were less likely to have struggled to control tilt interaction.
Chapter 7: Improving the Controllability of Tilt Interaction

Static and dynamic adaptation both achieved similar mean efficiency to the no adaptation setting (Figure 7.17). This is slightly surprising and encouraging, given that tilt interaction was significantly less sensitive when participants were using both static and dynamic adaptation. This is particularly encouraging in light of the accuracy results and indicates that both static and dynamic adaptation were able to achieve significant accuracy improvements, without negatively affecting efficiency. The fact that participants were not significantly more efficient using the no adaptation setting indicates that they had to exercise more caution and employed smaller tilt angles when using this setting.

User satisfaction ratings largely mirrored the objective data. No significant differences were recorded for perceived efficiency. Perceived controllability ratings showed higher ratings for static and dynamic adaptation compared to no adaptation, although only the difference between no adaptation and static adaptation was statistically significant (Figure 7.18).

User satisfaction ratings reflected a clear order of preference amongst participants for the three different settings (Figure 7.18). Static adaptation was rated ahead of dynamic adaptation for all four metrics (although none of these differences were statistically significant), which in turn was rated ahead of no adaptation for all four metrics. This trend is noteworthy, given that static and dynamic adaptation achieved very similar performance in terms of both accuracy and efficiency (both marginally in favour of dynamic adaptation). Closer inspection of participants’ comments revealed that this could be attributed to the fact that some participants found dynamic adaptation unpredictable because of the changing dampening factors employed. Furthermore, some participants expressed frustration with the high dampening factors employed by dynamic adaptation when walking at high speeds, and some participants incorrectly attempted to use extreme tilt angles to attempt to increase panning speeds in response. Analysis of log data showed that on average dynamic adaptation employed a slightly lower dampening factor of 2.59 (as opposed to 3.00 used by static adaptation), which could also have attributed to the lower ease of use and controllability ratings. Dynamic adaptation could benefit from more gradual changes in dampening factor. This could be achieved through the use of longer sampling periods or the use of an inertia weight to slow changes in dampening factor.

Analysis of mean walking speeds showed that very little difference was observed for the three sensitivity adaptation settings, with participants walking marginally slower when using the no adaptation setting. This could be due to the slightly artificial context of use of a
controlled user study, or simply indicative of the fact that users do not attempt to compensate for tilt inaccuracy by slowing their walking speed. Surprisingly, a stronger positive correlation was recorded between walking speed and mean route-following error for dynamic adaptation than static adaptation, given that dynamic adaptation was intended to compensate more for faster walking speeds, while retaining sensitivity at lower walking speeds (Figures 7.20 and 7.21).

### 7.13 Hypothesis Testing

In Section 7.7.2 several hypotheses to be tested in this user study were outlined. This section revisits these hypotheses in order to determine whether the results obtained provide sufficient evidence for the null hypotheses to be rejected. Hypothesis testing results are summarised in Tables 7.19 (zooming techniques) and 7.20 (sensitivity adaptation).

#### 7.13.1 Zooming Technique

\( H_{0,1Z} \): Tilt zooming does not exhibit better perceived controllability than gesture zooming.

Perceived controllability was significantly better for tilt zooming as opposed to gesture zooming for both seated and walking tasks (Tables 7.6 and 7.7). Qualitative comments regarding the controllability of tilt zooming were also positive (Table 7.12). There is therefore sufficient evidence to reject the null hypothesis \( H_{0,1Z} \) and conclude that tilt zooming exhibited an improvement in perceived controllability over gesture zooming.

\( H_{0,2Z} \): Tilt zooming is not more efficient than gesture zooming.

Tilt zooming was significantly more efficient than gesture zooming for both seated and walking tasks (Table 7.3). Participants also perceived tilt zooming to be significantly more efficient for seated tasks. The difference for walking tasks was not statistically significant. If we consider only objective efficiency, there is therefore sufficient evidence to reject the null hypothesis \( H_{0,2Z} \) and conclude that tilt zooming is more efficient than gesture zooming. For perceived efficiency, however, the null hypothesis can only be rejected for seated tasks.

\( H_{0,3Z} \): Tilt zooming does not exhibit better perceived workload than gesture zooming.

Tilt zooming achieved statistically significant improvements in several aspects of perceived workload. These differences were statistically significant for mental demand, physical demand, effort and frustration for seated tasks, and for mental demand and physical demand
for walking tasks. While there is therefore insufficient evidence to reject the null hypothesis $H_{0.3Z}$ outright, the sub-hypotheses for the above conditions can be rejected.

$H_{0.4Z}$: Participants do not find the overview feature useful.

Participants indicated that they found the overview feature useful in conjunction with both zooming techniques and in both contexts of use (Table 7.8). Participants were also observed to make effective use of the overview in locating distant POIs. There is therefore sufficient evidence to reject the null hypothesis $H_{0.4Z}$ and conclude that participants perceived the overview feature to be useful.

$H_{0.5Z}$: Participants do not prefer tilt zooming over gesture zooming.

Participants indicated a clear preference for tilt zooming for both seated and walking tasks (Table 7.9). There is therefore sufficient evidence to reject the null hypothesis $H_{0.5Z}$ and conclude that participants preferred tilt zooming over gesture zooming in both contexts of use evaluated.

<table>
<thead>
<tr>
<th>#</th>
<th>Null Hypothesis</th>
<th>Reject?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{0.1Z}$</td>
<td>Tilt zooming does not exhibit better perceived controllability than gesture zooming.</td>
<td>✓</td>
</tr>
<tr>
<td>$H_{0.2Z}$</td>
<td>Tilt zooming is not more efficient than gesture zooming.</td>
<td>✓</td>
</tr>
<tr>
<td>$H_{0.3Z}$</td>
<td>Tilt zooming does not exhibit better perceived workload than gesture zooming.</td>
<td>Partly</td>
</tr>
<tr>
<td>$H_{0.4Z}$</td>
<td>Participants do not find the overview feature useful</td>
<td>✓</td>
</tr>
<tr>
<td>$H_{0.5Z}$</td>
<td>Participants do not prefer tilt zooming over gesture zooming</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 7.19. Zooming technique hypothesis testing results

### 7.13.2 Sensitivity Adaptation

$H_{0.1S}$: Static sensitivity adaptation does not exhibit better controllability than no adaptation.

Static sensitivity adaptation was shown to provide a statistically significant improvement in accuracy for navigation tasks over the no adaptation setting (Table 7.14). There is therefore sufficient evidence to reject the null hypothesis $H_{0.1S}$ and conclude that static sensitivity adaptation exhibited better controllability than standard tilt interaction with no adaptation.
**Chapter 7: Improving the Controllability of Tilt Interaction**

$H_{0,1D}$: Dynamic sensitivity adaptation does not exhibit better controllability than no adaptation.

Dynamic sensitivity adaptation was shown to provide a statistically significant improvement in accuracy for navigation tasks over the no adaptation setting (Table 7.14). There is therefore sufficient evidence to reject the null hypothesis $H_{0,1D}$ and conclude that dynamic sensitivity adaptation exhibited better controllability than standard tilt interaction with no adaptation.

$H_{0,2S}$: Static sensitivity adaptation is not less efficient than no adaptation.

The mean task time for static sensitivity adaptation was only marginally slower than for the no adaptation setting (Figure 7.17). This difference was not statistically significant. There is therefore insufficient evidence to reject the null hypothesis $H_{0,2S}$.

$H_{0,2D}$: Dynamic sensitivity adaptation is not less efficient than no adaptation.

The mean task time for dynamic sensitivity adaptation was marginally slower than for the no adaptation setting (Figure 7.17). This difference was not statistically significant. There is therefore insufficient evidence to reject the null hypothesis $H_{0,2D}$.

$H_{0,3S}$: Static sensitivity adaptation does not exhibit better perceived controllability than no adaptation.

Mean ratings for perceived controllability were significantly higher for static sensitivity adaptation than for the no adaptation setting (Table 7.17). There is therefore sufficient evidence to reject the null hypothesis $H_{0,3S}$ and conclude that static sensitivity adaptation exhibits better perceived controllability than no adaptation.

$H_{0,3D}$: Dynamic sensitivity adaptation does not exhibit better perceived controllability than no adaptation.

While mean ratings for perceived controllability were higher for dynamic adaptation than for no adaptation, this difference was not statistically significant (Table 7.17). There is therefore insufficient evidence to reject the null hypothesis $H_{0,3D}$. 

| 195 |
Chapter 7: Improving the Controllability of Tilt Interaction

<table>
<thead>
<tr>
<th>#</th>
<th>Null Hypothesis</th>
<th>Reject?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{0.1S}$</td>
<td>Static sensitivity adaptation does not exhibit better controllability than no adaptation</td>
<td>✓</td>
</tr>
<tr>
<td>$H_{0.1D}$</td>
<td>Dynamic sensitivity adaptation does not exhibit better controllability than no adaptation</td>
<td>✓</td>
</tr>
<tr>
<td>$H_{0.2S}$</td>
<td>Static sensitivity adaptation is not less efficient than no adaptation</td>
<td>✗</td>
</tr>
<tr>
<td>$H_{0.2D}$</td>
<td>Dynamic sensitivity adaptation is not less efficient than no adaptation</td>
<td>✗</td>
</tr>
<tr>
<td>$H_{0.3S}$</td>
<td>Static sensitivity adaptation does not exhibit better perceived controllability than no adaptation</td>
<td>✓</td>
</tr>
<tr>
<td>$H_{0.3D}$</td>
<td>Dynamic sensitivity adaptation does not exhibit better perceived controllability than no adaptation</td>
<td>✗</td>
</tr>
</tbody>
</table>

Table 7.20. Sensitivity adaptation hypothesis testing results

7.14 Design Implications

The results of this user study provide further insight into the design of tilt interaction for mobile map-based applications. Following on from the design recommendations identified in Sections 5.7 and 6.10, the following design recommendations are proposed based on the experimental results presented in this chapter:

R5. Use rate controlled tilt zooming to improve efficiency and perceived controllability.

The results of this user study provide empirical evidence that rate controlled tilt zooming provides superior efficiency and user satisfaction over gesture zooming. Rate controlled tilt zooming was also observed to integrate well with tilt-controlled panning, with participants able to quickly switch between panning and zooming modes using the clutch mechanism.

The results of the user study showed a significant decrease in the perceived ease of use of tilt zooming for walking tasks. Given that tilt zooming employs a rate control mapping between tilt angles and zooming speeds similarly to tilt-controlled panning, the use of sensitivity adaptation could potentially result in similar controllability improvements for tilt zooming.

R6. Provide the user with an overview when panning quickly.

Positive results were obtained in this user study regarding the use of a semi-transparent overview overlaid on the main map when panning quickly. The results of the earlier user studies in which SDAZ was used showed that participants liked the sense of overview that
Chapter 7: Improving the Controllability of Tilt Interaction

SDAZ provided. The use of a separate overview overlaid on the main map was designed to provide similar benefits without the disorientation experienced with SDAZ. Participants rated the overview as useful and were observed to make effective use of the overview when attempting to locate POIs.

*R2 (Updated). Reduce tilt interaction sensitivity by a constant dampening factor when the user is mobile.*

The results of the user study presented in Chapter 5 showed that sensitivity adaptation provided a 35% improvement in accuracy for navigation tasks performed while walking. The results in this chapter showed a greater accuracy improvement for the two versions of sensitivity adaptation implemented (43% and 45%). While these results provide further support for the use of sensitivity adaptation (design implication *R2* in Chapter 5), they also provide further insight into how sensitivity adaptation should be implemented. The use of a constant dampening factor (static sensitivity adaptation) was shown to achieve better perceived ease of use and controllability than the use of a dynamic dampening factor. *R2* is therefore updated to propose the use of a constant dampening factor. Furthermore, the original implementation of sensitivity adaptation adjusted the discretisation intervals employed. The use of a dampening factor in this chapter allows sensitivity adaptation to be implemented independently of discretisation. While the use of a dynamic dampening factor showed promising accuracy results, further refinement of this technique is required. In particular, the use of more gradual changes in dampening factor could potentially improve this technique.

**7.15 Discussion**

The use of tilt interaction to perform zooming has previously been proposed (Hinckley and Song 2011, Büring et al. 2008). The implementation described in this chapter, however, is the first to combine the use of tilt interaction to perform both panning and zooming in a mobile map-based application. The use of rate-controlled tilt interaction was shown to facilitate easy interleaving of panning and zooming operations using tilt interaction. The results of the user study provide interesting and novel insights regarding the use of the different zooming techniques in different contexts of use. Participants reported a significant reduction in the ease of use of tilt zooming while mobile, whereas gesture zooming was reported to be easier to use while walking than while seated. The results of the evaluation show that tilt interaction can be effectively used to support both panning and zooming in a mobile map-based
Chapter 7: Improving the Controllability of Tilt Interaction

application. The use of a semi-transparent overview has also not previously been used in conjunction with tilt interaction. The results of the user study show that the overview proved effective in helping users to locate off-screen POIs.

The comparison between static and dynamic sensitivity adaptation provides insight into the design of tilt interaction for a mobile context of use. Both techniques were shown to offer significant controllability benefits for the user while mobile. The fact that static sensitivity adaptation was preferred over dynamic sensitivity adaptation is somewhat surprising. This result emphasises the importance of predictability in the design of an interaction technique.

7.16 Conclusions

This chapter proposed changes to tilt interaction designed to address controllability shortcomings identified during the user study described in Chapter 5. Gesture zooming was modified to address gesture recognition problems and to provide improved feedback. Tilt zooming was modified to incorporate rate control, allowing users control over panning speed and providing for easier switching between zooming and panning. An overview feature was also designed to aid users in locating distant POIs while minimising the need for zoom level changes.

The original implementation of sensitivity adaptation was modified to make use of a more scalable and repeatable learning technique in the form of an Artificial Neural Network. Training data was collected of users performing navigation tasks while seated and walking and used to train a neural network to identify seated and walking contexts of use based on accelerometer data. Despite fairly limited training data, this approach proved very accurate in correctly classifying the current context of use. Sensitivity adaptation was modified from the original implementation to reduce panning speeds by a pre-defined dampening factor, instead of adjusting the discretisation intervals used. An alternative sensitivity adaptation implementation, called dynamic adaptation, was also developed. Unlike static adaptation, which employs a fixed dampening factor for reducing tilt interaction sensitivity, dynamic adaptation changes the dampening factor in response to changes in accelerometer data variation.

A two-part user study was conducted to evaluate the modifications to tilt zooming and sensitivity adaptation. The first part of the user study was used to compare gesture zooming with tilt zooming in both walking and seated contexts of use. The second part of the user
Chapter 7: Improving the Controllability of Tilt Interaction

study compared no adaptation, static adaptation and dynamic adaptation, requiring users to perform route-following tasks while walking.

The results of the user study showed that participants clearly preferred tilt zooming over gesture zooming. While gesture zooming performed better for walking tasks than for seated tasks, it was still inferior in terms of efficiency and user satisfaction in both contexts of use. Tilt zooming offered better task times, lower perceived workload and greater perceived controllability. Given the objectives of this chapter, the significant increase in perceived controllability is particularly noteworthy. Participants regarded tilt zooming as a natural and smooth form of interaction and were able to efficiently change zoom levels and switch between panning and zooming using this technique. Participants also found the overview feature useful for locating POIs.

Results regarding sensitivity adaptation clearly showed the controllability benefits of this approach. Static and dynamic sensitivity adaptation both resulted in significant route-following accuracy improvements, while achieving similar efficiency to the no adaptation setting. Participants perceived static adaptation to offer significantly better effectiveness, ease of use and controllability over no adaptation. Dynamic adaptation also offered improvements in mean user satisfaction ratings. Unlike static adaptation, however, statistically significant user satisfaction improvements were not recorded. Analysis of qualitative feedback and log data suggest that participants preferred the more predictable approach offered by static adaptation. Dynamic adaptation would possibly benefit from slower changes in dampening factor (possibly by employing a longer sampling period and/or smoothing techniques). Dynamic adaptation could also be improved by employing a limit on the maximum dampening factor, as some participants complained that this technique was occasionally too insensitive.

The results of this user study provide empirical evidence that the proposed design modifications provided both actual and perceived controllability benefits for tilt interaction. Tilt zooming and sensitivity adaptation were both shown to allow participants greater control over tilt interaction.

The results of this user study were used to motivate three additional design recommendations regarding the use of tilt interaction in mobile map-based applications. The use of rate controlled tilt zooming was proposed based on the positive results achieved using this
technique in the user study. Providing an overview overlaid on the map display was also recommended based on positive feedback from participants. Making the overview semi-transparent and only displaying it when panning quickly helped to minimise occlusion problems. The use of static sensitivity adaptation was proposed based on the positive objective and subjective results obtained regarding the use of this technique in the user study.

The following chapter proposes a framework for mobile tilt interaction based on the results of this research. This framework will extend basic tilt interaction by encapsulating the various techniques which were shown to offer benefits for tilt interaction in this research.
Chapter 8: A Mobile Tilt Interaction Framework

8.1 Introduction

This thesis has discussed the development and evaluation of a range of techniques designed to address the shortcomings of tilt interaction. In this chapter, a framework for mobile tilt interaction is proposed, specifically for the domain of mobile map-based applications. This framework is designed to encapsulate the results of this thesis in order to show how the proposed techniques can be incorporated into the design of a mobile tilt interaction technique.

This chapter will begin by reviewing the design implications that were identified throughout this thesis based on the results of the various user studies that were conducted. A framework for mobile tilt interaction is then proposed, showing how the various techniques proposed in this thesis can be integrated into a tilt interaction technique. The framework is presented from both a procedural and an architectural perspective. The procedural view is presented first, outlining the various stages involved in tilt interaction. Each major stage is then discussed in more detail, highlighting the role of each of the techniques proposed in this research. Where appropriate, reference will be made to supporting evidence from the earlier chapters to motivate the various design elements. The architectural view of the framework is then presented. Finally, the implementation and use of the framework as an Android API is discussed. The framework encapsulates all the techniques that were shown to offer benefits (either objective or subjective) during the course of this research. The framework does not seek to prescribe a complete solution to the design of tilt interaction, but rather individual design elements that can be incorporated into tilt interaction as and when they are needed.

8.2 Design Implications Revisited

The results of the user studies in Chapters 5-7 were used to motivate the proposal of several recommendations for the design of tilt interaction for mobile map-based applications. These are summarised in Table 8.1. The framework proposed in this chapter will illustrate how these design recommendations can be implemented in the design of a mobile tilt interaction technique. Note that R2 in Table 8.1 was originally proposed in simpler form in Chapter 5 and updated based on the results presented in Chapter 7.
Table 8.1. Design recommendations for tilt interaction in mobile map-based applications

<table>
<thead>
<tr>
<th>#</th>
<th>Design Recommendation</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Use attractor mechanisms to allow for selection using approximate cursor positioning</td>
<td>5</td>
</tr>
<tr>
<td>R2</td>
<td>Reduce tilt interaction sensitivity by a constant dampening factor when the user is mobile</td>
<td>5, 7</td>
</tr>
<tr>
<td>R3</td>
<td>Use visual and vibrotactile feedback to improve perceived controllability and reduce mental demand</td>
<td>5</td>
</tr>
<tr>
<td>R4</td>
<td>Use sensor fusion to improve perceived controllability and efficiency when the user is mobile</td>
<td>6</td>
</tr>
<tr>
<td>R5</td>
<td>Use rate controlled tilt zooming to improve efficiency and perceived controllability for zooming operations</td>
<td>7</td>
</tr>
<tr>
<td>R6</td>
<td>Provide the user with an overview when panning quickly</td>
<td>7</td>
</tr>
</tbody>
</table>

8.3 MoTIF: A Mobile Tilt Interaction Framework

The Mobile Tilt Interaction Framework (MoTIF) is proposed in this section, extending the basic process of tilt interaction. Tilt interaction requires a continuous cycle between receiving raw sensor data and processing this data to identify and execute meaningful interactions within an application. Figure 8.1 provides a high-level view of the typical operations involved in implementing a tilt interaction technique and is based on the stages of sensor-based interaction identified in Figure 3.1 and extended in Figure 4.1. Figure 8.1 shows the following main stages and operations involved in tilt interaction:

- **Sensor Measurement**: Sensor data is typically accessed asynchronously via a listener interface.
- **Pre-processing Sensor Data**: Pre-processing is performed to smooth out noise in sensor data (particularly accelerometer data).
- **Feature Extraction**: Pre-processed sensor data is then used to calculate pitch and roll angles and to identify discrete gestures.
- **Mapping onto Operations**: An important part of any tilt interaction technique is the relationship between tilt input and the resultant effect in terms of the panning and/or zooming operations that are performed. This could be defined in terms of a functional relationship (e.g. a linear relationship between tilt angles and panning speeds). Alternatively, techniques such as discretisation, where the range of possible tilt angles is mapped onto different panning speeds, can be employed. The mapping between tilt...
angles and panning speeds is then used to calculate the current panning speed and/or zooming speed.

- **Perform Panning**: The horizontal and vertical panning speeds calculated in the previous stage are used to perform panning operations and update the current map position.

- **Perform Zooming**: The zoom level is adjusted using the change in zoom level calculated in the previous stage. In some techniques, such as SDAZ and SAZ, the zoom level may be a function of the current panning speed.

- **Perform Selection**: A wide variety of techniques have been employed in conjunction with tilt interaction to perform selections, including touch input, shake gestures and dwell-time selection.

- **Update Display**: The display is updated to reflect changes in the current position and zoom level and to display information about selected map elements, if appropriate.

![Figure 8.1. Procedural view of the Mobile Tilt Interaction Framework (MoTIF)](image)
This loop is then repeated with new sensor data. Note that while Figure 8.1 shows sensor data being used in a synchronous fashion, most implementations are likely to asynchronously access sensor data at a sampling rate independent of the frame rate through a listener interface. Furthermore, Figure 8.1 shows the process involved when tilt interaction is active. Most tilt interaction implementations provide some means of deactivating tilt interaction to avoid accidental input.

The following section will show how the different techniques developed and evaluated during this research fit into the above overview. The individual high-level stages of tilt interaction depicted in Figure 8.1 will be split into sub-stages, demonstrating how the proposed techniques can be integrated into the tilt interaction process.

### 8.3.1 Sensor Measurement, Pre-Processing and Feature Extraction

Figure 8.2 shows the sensor measurement, pre-processing and feature extraction stages of the MoTIF framework. Aspects shown in green were proposed and/or evaluated during the course of this research. Accelerometer data is first smoothed before being combined with compass and gyroscope data using sensor fusion. As part of this process, integration is performed to convert gyroscope data from angular velocity to orientation angles.

![Figure 8.2. Combined sensor measurement, pre-processing and feature extraction stages of MoTIF](image)

Existing research on tilt interaction has largely focused on the use of accelerometers to facilitate tilt interaction. Digital compasses and gyroscopes have more recently begun to emerge in increasing numbers of mobile phones. These additional sensors have enabled the implementation of sensor-based interaction techniques which take advantage of multiple...
sensors to minimise the limitations of the individual sensors. The results of the comparative user study in Chapter 6 showed that the use of sensor fusion provided significant efficiency and perceived controllability benefits in a walking context of use. Furthermore, these benefits were achieved using a fairly simple and computationally inexpensive sensor fusion algorithm.

Sensor fusion is increasingly being performed at a lower level, exposing the fused results to developers in API format (Microsoft 2011b, Invensense 2010, Apple Inc. 2011a). This could help to remove the burden from designers and developers of tilt interaction techniques and simplify the integration of sensor-based interaction in mobile applications. Nevertheless, the ability to perform sensor fusion manually provides developers with fine control over exactly how tilt interaction behaves and how different sensors’ input is weighted.

### 8.3.2 Mapping Tilt Input onto Panning and Zooming Operations

Figure 8.3 shows a more detailed view of the process of mapping tilt angles onto panning and zooming speeds. Modifications and techniques proposed in this research are highlighted in green.

![Diagram of Mapping between tilt angles and panning and zooming speeds](image)

**Figure 8.3. Mapping between tilt angles and panning and zooming speeds**

The process of mapping tilt angles onto panning speeds frequently makes use of discretisation (Section 3.4). This process divides the range of possible tilt angles into intervals, with each interval associated with a corresponding panning speed. This results in a
step-function mapping between tilt angles and panning speeds. In this research, as with previous research which has employed tilt interaction, the use of a linear mapping for small tilt angles, combined with quadratic speed increases for larger tilt angles, has successfully been employed. Figure 8.3 depicts panning and zooming as taking place in parallel, although given the techniques employed in Chapter 7, panning and zooming would take place sequentially.

Figure 8.3 depicts how rate controlled tilt zooming can be integrated into tilt interaction. Similarly to rate controlled panning, the output of the discretisation process is used to determine the corresponding tilt angle interval for the current pitch angle. Unlike panning, tilt zooming only relies on the pitch angle and not the roll angle. The same set of discretisation intervals can be used for both panning and zooming. Alternatively, as implemented in Chapter 7, separate interval scales can be used for panning and zooming. A separate mapping between angles and zooming speeds is necessary as panning and zooming speeds are measured in different units of measurement. In the implementation in Chapter 7, panning speed is measured in pixels per repaint and zooming speed as the change in drawing scale per repaint.

The results of the comparative user study between gesture zooming and rate controlled tilt zooming in Chapter 7 clearly indicated that tilt zooming was superior in terms of both objective and subjective metrics. Gesture zooming was slightly preferred over SDAZ in the user study discussed in Chapter 5. A position controlled tilt zooming implementation was also discussed in Chapter 6, but not formally evaluated. This technique was shown to suffer from fundamental design shortcomings compared to rate controlled tilt zooming. Therefore, the results of this research indicate that the use of rate controlled tilt zooming is the optimal choice of the zooming techniques tested.

Figure 8.3 suggests that sensitivity adaptation should be performed after both panning and zooming speeds are calculated. The results of Chapter 7 showed clear benefits in terms of accuracy and perceived controllability for the use of static sensitivity adaptation. Dynamic sensitivity adaptation showed significant accuracy improvements for tilt interaction, but without statistically significant perceived controllability benefits. Therefore, static sensitivity adaptation is recommended ahead of dynamic adaptation. The use of sensitivity adaptation was not tested in conjunction with tilt zooming in Chapter 7. However, a statistically significant decrease in perceived ease of use was recorded for tilt zooming for the walking
context of use for this experiment. It is therefore likely that tilt zooming would also benefit from the use of sensitivity adaptation.

Figure 8.3 also shows how the overview function, which was evaluated in Chapter 7, is integrated into MoTIF. The overview is displayed when the panning speed exceeds a pre-determined threshold. In the user study described in Chapter 7, this threshold was reached when the pitch or roll angle fell into at least the third discretisation interval. The overview function was rated as useful by participants in the user study described in Chapter 7 and participants were observed to make effective use of this feature in locating POIs.

### 8.3.3 Performing Panning

Figure 8.4 shows the panning stage of the MoTIF framework. Green blocks show modifications/techniques proposed in this research.

Panning is performed by updating the x and y positions using the horizontal and vertical panning speeds during each repaint. Attractors and vibrotactile feedback were designed to make accurate selection easier (Chapter 5). Visual feedback was designed to provide improved control over panning speed and to reduce mental demand. All three measures were positively received by participants. Quantitative and qualitative feedback from participants
Chapter 8: A Mobile Tilt Interaction Framework

showed that visual and vibrotactile feedback and attractors were all regarded as useful (Chapter 5). Log data showed that attractors helped to achieve more efficient and accurate selection and also helped to reduce overshooting errors.

The visual feedback technique that was tested employed the use of arrows attached to the selection cursor. The length of the arrows is a linear function of the current vertical and horizontal panning speeds. As depicted in Figure 8.4, vibrotactile feedback and attractors both require regular checks to determine whether the current position is within range of any icons. If the current position is within range of an icon and the user is panning at low speed (below the third discretisation level) then panning is performed to draw the cursor towards the icon in question. The implementation employed in this research uses a constant attraction speed, drawing the cursor towards the nearest icon at a speed of one pixel per repaint. This implementation could easily be modified to employ an attractor speed that is relative to the distance between the cursor and the nearest icon. If the cursor is entering the boundary region of an icon for the first time, then vibration feedback is provided to the user.

8.3.4 Performing Zooming

Figure 8.5 shows the zooming stage in the MoTIF framework. The current zoom level is adjusted using the current zooming speed calculated during the previous stage. The current zooming speed and direction are also communicated to the user through the use of visual feedback techniques. This consists of the use of on-screen arrows to indicate zooming direction and a zoom level indicator.

Figure 8.5. Zooming stage in MoTIF
8.3.5 Performing Selection

Figure 8.6 shows an overview of the selection stage in the MoTIF framework. This is largely unchanged, except for the mode of triggering selection. In Chapter 5, the use of dwell-time selection was trialled. This was used in conjunction with touch-screen selection, which was shown to occasionally lead to selection errors in the user study described in Chapter 4. Analysis of the log data showed that participants made more frequent use of dwell-time selection than touch-screen selection. The use of shake gestures to trigger selection was initially rejected and considered infeasible because of the fact that it would cause the cursor to pan away from the currently highlighted icon. However, subsequent to the user study in Chapter 7, shake selection was implemented. Informal trials showed this method of selection to be highly effective, especially when used in conjunction with attractors, which helped keep the cursor on the icon in question. The benefit of selection techniques such as dwell-time selection and shake selection are that these techniques allow a purely one-handed implementation of tilt interaction. Users can also perform selection operations without changing their grip on the phone.

![Figure 8.6. Selection stage in MoTIF](image)

8.3.6 Updating the Display

The final stage in the tilt interaction loop is updating and rendering the display (Figure 8.7). In this step, the map is drawn at the current coordinates and zoom level. Visual feedback appropriate for the current panning speed and direction and the current zoom level and direction is also displayed. If the overview is currently activated (the user is panning at sufficient speed), this overview is rendered on top of the main display, showing a semi-
transparent zoomed out view indicating the currently visible portion of the map. Vibrotactile feedback is provided to the user when entering the boundary region of an icon.

**Figure 8.7.** Updating and rendering the display in MoTIF

### 8.3.7 Supporting Evidence

Table 8.2 indicates the supporting evidence in this thesis (in the form of experimental results) which was used to motivate the inclusion of each of the modified/proposed techniques or design elements included in the MoTIF framework.

<table>
<thead>
<tr>
<th>Design Element</th>
<th>Supporting Evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Fusion</td>
<td>Table 6.1, Table 6.2, Table 6.5, Table 6.7</td>
</tr>
<tr>
<td>Sensitivity Adaptation</td>
<td>Table 5.4, Figure 5.14, Tables 7.14 – 7.17</td>
</tr>
<tr>
<td>Attractors</td>
<td>Tables 5.6, 5.7, 5.8 and 5.12, Figure 5.14</td>
</tr>
<tr>
<td>Vibrotactile Feedback</td>
<td>Tables 5.10 and 5.12</td>
</tr>
<tr>
<td>Visual Feedback</td>
<td>Tables 5.10 and 5.12</td>
</tr>
<tr>
<td>Overview</td>
<td>Table 7.8</td>
</tr>
<tr>
<td>Rate controlled Tilt Zooming</td>
<td>Table 7.3 – 7.7, Table 7.9, Table 7.12</td>
</tr>
<tr>
<td>Dwell-time selection</td>
<td>Table 5.5</td>
</tr>
</tbody>
</table>

**Table 8.2.** Proposed/modified aspects of tilt interaction and supporting evidence

### 8.4 Framework Architecture

The previous section outlined MoTIF from a procedural perspective, demonstrating the stages involved in transitioning from raw sensor data to meaningful interaction operations. In this section, a proposed architecture is presented, demonstrating how MoTIF can be implemented in a mobile map-based application.
Figure 8.8 illustrates the proposed MoTIF architecture. The architecture makes use of three components to implement the techniques proposed in this research, namely the Sensor Data Provider, Context Calculator and Position Listener components. Each of these is now discussed in more detail.

### 8.4.1 Sensor Data Provider

The Sensor Data Provider is used to perform the first four stages identified in Figure 8.1, namely accessing and pre-processing raw sensor data and mapping the calculated orientation data onto panning and zooming speeds. The functions of the Sensor Data Provider can therefore be summarised as follows:

- **Managing sensor measurement:** The Sensor Data Provider is responsible for asynchronously measuring sensor data from the accelerometer, digital compass and gyroscope (where these sensors are available) and managing the connections with these sensors.
• **Pre-processing sensor data:** Smoothing of sensor data (particularly accelerometer data) is performed to remove outliers and noise.

• **Performing sensor fusion:** Sensor fusion is used to calculate the current orientation of the mobile phone using the pre-processed sensor data.

• **Calculating panning speed:** The current pitch and roll angles are mapped onto panning speeds and directions using an appropriate set of discretisation intervals.

• **Calculating zooming speed:** The current pitch angle is mapped onto a zooming speed and direction using an appropriate set of discretisation intervals.

• **Handling discrete events:** The Sensor Data Provider is also responsible for identifying and processing discrete gestures such as the vertical gestures used for gesture zooming and shake gestures.

• **Providing access to sensor data:** The Sensor Data Provider provides a single point of access through which all screens employing tilt and sensor-based interaction are able to request sensor-related data (such as the current orientation).

• **Managing interaction:** The Sensor Data Provider allows tilt interaction to be activated and deactivated and stores the current neutral orientation.

### 8.4.2 Context Calculator

The Context Calculator is used to determine the user’s current context based on raw sensor data. In this research, several features of raw accelerometer data (including mean, standard deviation and covariance) were used to distinguish between walking and seated contexts of use. This could easily be extended to cater for additional contexts of use such as running or climbing stairs. The Context Calculator retrieves raw sensor data via the Sensor Data Provider and uses this data to continually determine the user’s current context. This information is then used by the Sensor Data Provider in determining whether sensitivity adaptation is necessary when calculating panning and zooming speeds.

### 8.4.3 Position Listener

Several of the techniques proposed in this research rely on the current cursor position. Attractors, vibrotactile feedback and dwell-time selection all require the current cursor position to be taken into account. The MoTIF architecture proposes the use of a Position Listener component to enable the use of position-related effects such as attractors. Position Listeners can be notified when the current cursor position matches some criteria (such as when the cursor has passed within range of a selectable icon). Multiple Position Listener
components can be created to enable multiple position-related effects. For example, different effects can be implemented when passing over a POI icon as opposed to when passing over a route marker.

8.5 Framework Implementation

MoTIF was implemented as an API designed for the Android Platform (specifically for Android version 2.3). The design of this API evolved and expanded through the various experiments that were conducted during the course of this research and was used in the implementation of not only MapExplorer, but also a tilt-controlled image browsing application, called PhotoWave (Leonard 2011). Feedback from the design process involved in developing these prototypes was used to modify and improve the API. The MoTIF API is built on top of the existing Android sensor classes, which provide access to raw sensor data and also allow calculation of absolute tilt angles using the raw data.

8.5.1 API Overview

Figure 8.9 shows a simplified class diagram demonstrating the implementation of the MoTIF API in Android. The API provides three classes, SensorDataProvider, ContextCalculator and Settings and two interfaces, SensorScreen and PositionListener. The SensorDataProvider class implements the Sensor Data Provider component of MoTIF. SensorDataProvider uses the Singleton design pattern (Gamma 1995). This ensures that only a single instance of this class can be instantiated to provide a universal point of access to sensor data and maximise efficiency. The SensorDataProvider class allows the developer to specify which sensors to employ (accelerometer, digital compass and/or gyroscope, subject to hardware support from the phone in question). The SensorDataProvider manages interaction with the various sensors via implementations of the Android SensorEventListener interface. Sensor data is retrieved asynchronously, with consumer classes requesting the data as needed. Tilt angles are calculated using various methods, depending on the currently selected sensors. If all three sensors are activated, sensor fusion is employed as outlined in Chapter 6 to calculate the current orientation.

The ContextCalculator class is responsible for continuously calculating the user’s current context. ContextCalculator retrieves raw sensor data from the SensorDataProvider class. This data is then used to asynchronously update the current context of use. The
SensorDataProvider accesses the current context when determining whether sensitivity adaptation is necessary for calculating the current panning speed.

The SensorScreen interface provides the SensorDataProvider with a known interface through which it can interact with the currently active screen supporting tilt interaction. Discrete events such as zooming gestures and shake gestures are communicated via this interface. Only one SensorScreen can be set to consume events from the SensorDataProvider class at a time. The current panning and zooming speeds can be accessed as needed from the SensorDataProvider to perform panning and tilt zooming.

Figure 8.9. Simplified class diagram of application implementing the MoTIF API (without parameters)
The `SensorScreenModel` class in Figure 8.9 is used to store the current state of the `SensorScreen` separately from its visual representation. This allows the state of the screen to be easily maintained should the user switch to other screens.

The `Settings` class stores global system settings related to the various techniques incorporated into MoTIF (e.g. whether the overview feature is currently enabled). The `Settings` class also implements the Singleton design pattern as there is only one shared instance across all screens. Any `SensorScreen` implementation must first check with the current `Settings` instance before employing the relevant techniques.

One or more implementations of the `PositionListener` interface can be associated with any `SensorScreenModel`. `PositionListener` makes use of the Observer design pattern, with the `SensorScreenModel` notifying registered listeners of position changes when appropriate (Gamma 1995). Implementations of `PositionListener` can be used to implement position-dependent techniques such as attractors, vibrotactile feedback and dwell-time selection. `PositionListeners` have an associated range (distance) and type (e.g. POI icon, route markers, or locations on the map).

### 8.5.2 API Implementations

The MoTIF API was used in the implementation of two prototype mobile applications supporting tilt interaction. The first, MapExplorer, has been discussed throughout this thesis and was used in the various user studies that were performed. The second is a prototype photo gallery application employing tilt interaction, called PhotoWave (Figure 8.10).

![PhotoWave photo gallery application using tilt interaction (Leonard 2011)](image)
The full MoTIF framework was employed in the implementation of MapExplorer. PhotoWave made use of the Sensor Data Provider component to implement tilt interaction, including gesture and tilt zooming. PhotoWave did not use the Position Listener and Context Calculator components, as sensitivity adaptation, attractors and vibrotactile feedback were not employed.

8.6 Discussion

The framework proposed in this chapter is a novel contribution to the field of sensor-based interaction. The MoTIF framework demonstrates how the techniques proposed in this research can be successfully incorporated into the design of a tilt interaction technique, both from an architectural and a procedural point of view. The implementation of the framework as an API facilitates the easy re-use of the framework in applications incorporating sensor-based interaction.

8.7 Conclusions

This chapter proposed a framework for the design of tilt interaction in a mobile map-based application. The MoTIF framework incorporates the techniques and design recommendations that were proposed during the course of this research. The framework is described from both a procedural and an architectural perspective. The MoTIF framework splits the tilt interaction process into eight distinct stages. The first three stages involve accessing and pre-processing sensor data and using this data to determine device orientation. The results of Chapter 6, where sensor fusion was shown to offer efficiency and perceived controllability benefits, were used to motivate the inclusion of additional sensors and the use of sensor fusion.

The fourth stage involves the mapping between tilt input and the corresponding panning and zooming operations. MoTIF proposes three additional steps in addition to the classic mapping between tilt angles and panning speed. The use of tilt angles to control zooming speed is proposed, based on the positive results received regarding the use of rate controlled tilt zooming in Chapter 7. The use of an overlaid overview in the top right corner of the display is proposed based on the positive feedback and observations of the use of this approach in Chapter 7. Finally, the use of sensitivity adaptation is proposed, based on the promising results achieved for this technique in Chapter 5 and confirmed in Chapter 7. Sensitivity adaptation is also suggested for performing zooming, as the results of Chapter 7 showed a
statistically significant decrease in ease of use for tilt zooming tasks performed while walking.

The fifth stage involves performing panning. Three additional steps are also proposed in this stage as part of MoTIF. The use of attractors is suggested, based on the results of Chapter 5 which showed that attractors helped to improve selection accuracy and efficiency. Visual and vibrotactile feedback are also proposed, based on positive user feedback regarding the use of these techniques in Chapter 5.

The sixth stage involves performing zooming. The only modification proposed to this stage (in addition to the mapping between pitch angles and zooming speeds proposed in stage two) is the use of visual feedback to indicate the current zooming speed and direction. The seventh stage involves selection, in which dwell-time selection and shake selection are proposed as alternatives to the classic techniques of hardware buttons and touch-screen selection. The final stage updates the display, including displaying the appropriate visual feedback to the user regarding panning and zooming speed and direction. An overview is also displayed if this feature is activated.

The proposed MoTIF architecture makes use of three core components. The Sensor Data Provider manages sensor measurement, pre-processing, feature extraction and mapping tilt interaction and discrete gestures onto panning, zooming and selection operations. The Context Calculator component keeps track of the user’s current context and is used to decide whether sensitivity adaptation is necessary. The Position Listener component allows the implementation of techniques such as attractors, vibrotactile feedback and dwell-time selection which take the current cursor position into account.

The MoTIF framework was implemented as an Android API to facilitate easy reuse of the techniques proposed in this research. The MoTIF API was successfully used in two mobile applications, namely MapExplorer and PhotoWave, a mobile image gallery application supporting tilt interaction.

The following chapter will conclude this thesis and highlight the contributions made by this research.
Chapter 9: Conclusions

9.1 Introduction

This chapter will present conclusions which can be drawn from this research. The results of this research will first be summarised, highlighting the most noteworthy findings. The contribution of this research will then be summarised, highlighting theoretical and practical contributions. Limitations and problems encountered will also be briefly discussed. Finally, opportunities for future work stemming from this research will be identified.

9.2 Summary of Findings

The objectives of this research were:

1. To identify the problems and shortcomings of existing interaction techniques for mobile map-based applications (Chapter 2);
2. To identify and analyse existing sensor-based interaction techniques for interacting with mobile map-based applications (Chapter 3);
3. To compare sensor-based interaction to keypad interaction for mobile map-based applications (Chapter 4);
4. To identify enhanced interaction techniques to address the usability shortcomings of sensor-based interaction (Chapters 5 and 7);
5. To evaluate the proposed techniques against existing techniques (Chapters 5 and 7);
6. To compare sensor-based interaction using sensor fusion with a single-sensor implementation (Chapter 6); and
7. To propose a framework for the design of enhanced sensor-based interaction techniques (Chapter 8).

Tilt interaction was identified as the sensor-based interaction technique most suited to interacting with mobile map-based applications (Section 3.6.6). This section will summarise the main findings of this research in relation to each of the above objectives.

9.2.1 Existing Interaction Techniques

Three key operations for interacting with mobile map-based applications were identified in Section 2.4, namely panning, zooming and selection. Historically, keypad and touch-screen interaction have been used to perform these operations. Several shortcomings were identified
for each of these interaction techniques. Keypad interaction is a binary form of input and therefore does not allow for precise control over panning speed. Furthermore, keypad interaction is more restrictive in terms of the range of motion afforded to users. These two limitations mean that keypad interaction can often be a frustrating form of interaction to use in mobile map-based applications (Section 2.6).

Touch-screen interaction is more intuitive and offers control over panning speed and direction. Touch-screen interaction, however, also has several shortcomings. Two-handed interaction is often required (particularly for zooming gestures), which is often not desirable in a mobile context of use. Touch-screen interaction also results in the display being occluded during interaction. The use of controls such as zoom level sliders, scroll bars and touch-screen buttons also require users to split their attention between the map and the user interface controls and take up valuable screen space. Precise selection is often difficult using touch-screen interaction (Section 2.6).

### 9.2.2 Existing Sensor-Based Interaction Techniques

Five stages were identified in the process of converting raw sensor data into meaningful interaction in the form of panning, zooming and selection operations (Section 3.3). The first two stages involve retrieving raw sensor data and performing pre-processing to remove noise and unwanted variance. The processed data is then used to calculate useful information, such as orientation angles. The extracted information is mapped onto operations such as panning, zooming and selection. Finally, the corresponding operations are performed.

Existing sensor-based interaction techniques were analysed in terms of the typical mobile map-based operations, namely panning, zooming and selection (Section 3.4). The use of tilt interaction to perform panning was found to have been successfully employed in mobile map-based applications, with accelerometers most commonly used to calculate tilt angles (Section 3.4.1). Position control (in which the range of tilt motion is mapped onto the map area) and rate control (in which tilt angles are mapped onto corresponding panning speeds) were identified as the two dominant techniques for mapping tilt angles onto panning operations. Discretisation was also identified as having been successfully employed in conjunction with tilt interaction to improve controllability, with quadratic discretisation proving the most effective. The use of pressure input and cameras to detect phone movement were also identified as alternative approaches to panning.
Several techniques for performing zooming independently of panning were identified (Sections 3.4.2). Tilt interaction has been used to provide the user with continuous control over the current zoom level. Gestures have been used to perform discrete zoom level changes. Pressure input has been used to perform zooming, with the level of pressure used to indicate zooming direction. Phone cameras have been used to detect when the device has been moved up or down to perform zooming. More commonly, tilt interaction implementations have linked the zoom level to the panning speed (Section 3.4.3). Techniques such as SDAZ and SAZ automatically zoom out when panning at high speeds to avoid blurring and provide a sense of overview, and zoom back in when panning at slow speeds to provide a detailed view.

Sensor-based interaction techniques were found to incorporate both traditional and sensor-based methods of selection (Section 3.4.4). Traditional techniques include the use of hardware buttons and touch-screen input to perform selection. Sensor-based techniques include the use of physical gestures and techniques which require the user to settle the cursor over a target icon for a pre-defined time period.

Several benefits of sensor-based interaction techniques were identified in Section 3.5. Sensor-based interaction was found to allow for one-handed interaction with mobile applications. Previous research has shown that users would prefer to interact with mobile map-based applications using only one hand. Sensor-based interaction techniques are frequently described as intuitive forms of interaction and often rely on natural physical gestures. Sensor-based interaction also eliminates occlusion problems associated with touch-screen interaction. In contrast to keypad interaction, sensor-based techniques, such as tilt interaction, provide the ability to control panning speed and direction by adjusting the magnitude of the tilt angles employed.

Five major problems and shortcomings related to the use of sensor-based interaction were identified, namely:

- **Physical limitations**: Sensor-based interaction techniques often restrict how users can hold the phone. Human physical limitations (such as the range of comfortable tilt angles) must also be considered.
- **Toggling input**: Problems are often encountered toggling sensor-based interaction on and off to prevent accidental input.
• **Establishing a neutral angle:** Sensor-based interaction requires a method of establishing a neutral angle relative to which device orientation is calculated.

• **Display visibility:** Display visibility can be problematic with sensor-based interaction techniques which involve tilting and physical movement of the phone. The phone may be pointed away from the user and screen reflections could hamper visibility.

• **Controllability:** Previous research has shown that sensor-based interaction techniques such as tilt interaction and camera-based interaction suffer from controllability problems due to the inexact nature of sensor input.

Tilt interaction was identified as the most suitable sensor-based interaction modality for mobile map-based applications due to its intuitive nature, feasibility given current hardware, superior software platform support and suitability for panning large information spaces (Section 3.6.6).

### 9.2.3 Keypad and Tilt Interaction

The objective of Chapter 4 was to compare tilt interaction to keypad interaction in the domain of mobile map-based applications. This was done in order to determine the shortcomings of tilt interaction in this domain. Keypad and tilt interaction (incorporating SDAZ) were compared in a user study which required users to perform three types of typical mobile map-based tasks, namely locating, navigating and checking tasks. A prototype mobile map-based application, called MapExplorer, was developed and used as a prototyping platform throughout this thesis.

The results of the comparative user study showed that for tasks requiring accurate cursor positioning and selection (locating and checking tasks), participants were more efficient using keypad interaction (Table 4.9). For tasks requiring freedom of movement and speed control (navigation tasks), tilt interaction was slightly more efficient. Subjective ratings showed that for the first group of tasks, perceived workload was higher for tilt interaction (Table 4.3), but by the last group of tasks (requiring similar actions), this trend was reversed (Table 4.5). This indicates that participants became comfortable using tilt interaction fairly quickly. The one workload measure that was consistently worse for tilt interaction was mental demand. Perceived user satisfaction was similar for both techniques, although tilt interaction was perceived to be easier to control and use for navigation tasks (Table 4.4).
Analysis of the results of this user study (including participants’ comments) allowed several shortcomings of tilt interaction to be identified. These shortcomings were:

- **Controllability**: Participants perceived tilt interaction to be difficult to control and struggled to predict the results of panning operations, resulting in overshooting errors.
- **Precise selection**: Participants found it difficult to precisely position the cursor in order to perform selection operations.
- **Mental demand**: Participants found tilt interaction to be more mentally demanding for all three task types.
- **Zooming technique**: Many participants did not like the use of SDAZ, feeling that this technique took control away from them.
- **Practicality**: Several participants raised concerns regarding the practicality of tilt interaction, particularly in mobile contexts of use.

The shortcomings identified provided several opportunities for enhancing tilt interaction.

### 9.2.4 Proposed Techniques

Several techniques were proposed to address the shortcomings of tilt interaction identified in Chapter 4. In particular, the following techniques were proposed:

- **Visual Feedback**: Existing implementations of tilt interaction make only limited use of visual feedback. The user is usually left to judge the current panning speed based on the results of his/her actions and subtle speed changes may be difficult to perceive. The use of arrows attached to the selection cursor to indicate the current panning speed and direction were proposed (Section 5.2.1). Separate horizontal and vertical arrows indicate the horizontal and vertical components of the current panning speed using a linear mapping between speed and arrow length. The positioning of the arrows means that users do not need to shift their attention from the cursor and provides them with a clearly understandable representation of the current panning speed. This form of visual feedback was developed to reduce mental demand and improve the user’s sense of control.

- **Vibrotactile Feedback**: Vibrotactile feedback was proposed to provide non-visual feedback to aid in selection operations (Section 5.2.2). Short vibration pulses are provided to the user when passing over selectable icons. Vibration feedback is only provided the first time the user enters the region surrounding a particular icon and only when panning
at slow speeds, when selection is likely, to avoid excessive vibration feedback. This form of feedback was intended to aid users in performing selection operations and to reduce mental demand, by reducing the need for the user’s full visual attention when performing selection operations.

- **Attractors:** The implementation of attractors in this research was inspired by the successful implementation of this technique in mobile photo browsing application using tilt interaction. Attractors were designed to provide users with assistance in controlling selection operations (Section 5.2.3). When the cursor enters the region surrounding a particular icon, the system aids selection by drawing the cursor towards the icon. The user is therefore only required to approximately position the cursor, with attractors assisting in exact positioning. Attractors are only activated when panning at slow speeds to avoid unwanted effects when not attempting to perform selection.

- **Gesture Zooming:** Gesture zooming was proposed as an alternative to SDAZ, providing users with an explicit means of controlling zooming operations. Gesture zooming employs discrete gestures relative to the z-axis of the phone to perform zooming operations. Gesture zooming was first proposed in Section 5.2.5, although the initial implementation was shown to suffer from gesture recognition problems and mixed results were achieved when compared to SDAZ. Modifications to gesture zooming were proposed in Section 7.3 in order to address the shortcomings identified and the use of a clutch mechanism was implemented to allow the technique to be made more sensitive without causing false positive gesture recognition errors.

- **Dwell-Time Selection:** Dwell-time selection was proposed in Section 5.2.6 to address usability problems regarding the use of touch in conjunction with tilt to perform selections. Participants in the user study described in Chapter 4 were observed to pan away from the targeted icon while reaching to perform touch selections. Dwell-time selection merely requires the user to settle the cursor on a particular icon. If the cursor remains on that icon for longer than a pre-determined interval (2.75 seconds from entering the icon boundary region), then the icon is selected. If the user begins to pan away, dwell-time selection is not triggered. Visual and vibrotactile feedback (in the form of a flashing cursor and vibration pulses) are used to indicate that a selection is about to take place.

- **Sensitivity Adaptation:** Sensitivity adaptation was designed to improve the controllability of tilt interaction while walking and thereby address practicality concerns regarding the
use of tilt interaction. The initial implementation of sensitivity adaptation (Section 5.2.4), used pre-defined acceleration variance thresholds to identify walking and seated contexts of use. When walking was identified, the sensitivity of tilt interaction was reduced by widening the discretisation intervals employed. Sensitivity adaptation was further developed in Chapter 7, with the proposal of static sensitivity adaptation (Section 7.6.2) and dynamic sensitivity adaptation (Section 7.6.3). In contrast to the approach in Chapter 5, static and dynamic sensitivity adaptation were trained to identify walking and seated contexts of use based on weights learned by an Artificial Neural Network after a training phase. These weights were used to determine the influence of the acceleration mean, standard deviation and correlation values in calculating the current context of use. Static adaptation modified tilt interaction sensitivity by decreasing the current panning speed by a fixed dampening factor. Dynamic adaptation employed a dynamic dampening factor linked to the current level of variation in accelerometer data.

- **Tilt zooming:** A position controlled tilt zooming technique was proposed in Section 6.5.2. This approach mapped the pitch angle onto the current zoom level, with a clutch mechanism employed to switch between panning and zooming modes. This technique was used in the user study described in Chapter 6, but was not formally evaluated. Observation of the technique in use, however, revealed that it suffered from a significant shortcoming. Tilting gestures to perform zooming frequently resulted in the phone being tilted to an uncomfortable angle. As a result, the phone had to be re-oriented before panning could resume. Rate controlled tilt zooming was proposed as an alternative in Section 7.4. In this approach, the current pitch angle is mapped onto the current zooming speed. Smaller pitch angles therefore result in slow changes in zoom level with larger pitch angles resulting in rapid changes in zoom level. This technique was intended to offer better controllability than gesture zooming, while integrating better with tilt-controlled panning than position controlled tilt zooming.

- **Overview:** The use of an overview overlaid on the main display was proposed for use in conjunction with gesture and tilt zooming. In order to minimise occlusion problems, this overview was designed to be semi-transparent and is only displayed when panning quickly.

### 9.2.5 Comparison of Existing and Proposed Techniques

Two separate user studies were conducted to evaluate the proposed techniques described in the previous section. In the user study presented in Chapter 5, an enhanced tilt interaction
implementation incorporating visual and vibrotactile feedback, attractors, gesture zooming, sensitivity adaptation and dwell-time selection was compared to a basic tilt interaction implementation. Several positive results were obtained from this user study regarding the proposed techniques.

Visual and vibrotactile feedback both received strongly positive feedback from participants. Participants rated vibrotactile feedback as useful for performing selection operations and visual feedback as useful for controlling panning speed and direction (Table 5.10). Several positive comments were also received regarding both forms of feedback that were provided (Table 5.12).

The use of attractors is likely to have contributed strongly to the statistically significant improvement in ease of selection that was reported (Figure 5.14). Qualitative feedback regarding attractors was also positive, with several participants commenting favourably on the benefits of attractors for the purpose of performing selection operations (Table 5.12). Analysis of log data showed that attractors helped to reduce overshooting errors, improve the accuracy of cursor positioning for selection operations and also helped to reduce the time between entering a POI's boundary region and settling the cursor on the icon to perform a selection operation (Tables 5.6-5.8).

Results regarding participants’ preferred zooming technique were mixed, with participants only slightly preferring gesture zooming over SDAZ (Table 5.9). Qualitative comments revealed usability issues with both zooming techniques. Participants didn’t like the fact that SDAZ took control away from them, while gesture zooming suffered from gesture recognition errors (Tables 5.11 and 5.12). SDAZ was also perceived to be difficult to control while walking.

The enhanced tilt interaction technique, which incorporated sensitivity adaptation, achieved a statistically significant accuracy improvement over the no adaptation setting for navigation tasks performed while walking (Table 5.4). Participants’ perceptions supported this result, with participants regarding the enhanced tilt interaction technique as significantly easier to use while walking as compared to the basic tilt technique (Figure 5.14).

Dwell-time selection was not specifically evaluated, but qualitative comments regarding this technique were positive. Participants were frequently observed to make use of this selection method, even though touch-screen selection was available as an alternative. Analysis of log
data showed that 10 out of 16 participants made more frequent use of dwell-time selection than touch-screen selection (Table 5.5).

The second user study to evaluate the proposed techniques was described in Chapter 7. The comparison between rate controlled tilt zooming and gesture zooming (which was improved from the implementation evaluated in Chapter 5), showed that tilt zooming provided several objective and subjective benefits. Tilt zooming proved more efficient for seated and walking tasks (Table 7.3). Participants reported lower perceived workload ratings for tilt zooming as compared to gesture zooming. Mental and physical demand were lower for both seated and walking tasks (Tables 7.4 and 7.5). User satisfaction was also in favour of tilt zooming for both seated and walking tasks, with an improvement in perceived controllability and ease of use being particularly noteworthy (Table 7.6 and Table 7.7). Participants indicated a clear preference for tilt zooming over gesture zooming, with 25 out of 30 participants indicating tilt zooming as their preferred technique (Table 7.9). Participants found the overview function to be useful when used in conjunction with both interaction techniques (Table 7.8). Participants’ general comments revealed that they found tilt zooming to be natural and intuitive and many participants commented favourably on the smooth, continuous nature of this technique (Table 7.12).

The user study described in Chapter 7 also compared the no adaptation, static sensitivity adaptation and dynamic sensitivity adaptation techniques for route-following tasks performed while walking. The results of the user study showed that both static and dynamic sensitivity adaptation were able to achieve statistically significant accuracy improvements over the no adaptation technique (Table 7.14). The two adaptation techniques achieved similar efficiency results to the no adaptation technique (Figure 7.17). User satisfaction results showed that the static adaptation technique was able to achieve statistically significant perceived effectiveness, ease of use and controllability improvements over the no adaptation technique (Tables 7.15 – 7.17). Dynamic sensitivity adaptation achieved better user satisfaction results than no adaptation, but the differences were not statistically significant. Comments from participants indicated that this was due to some participants finding the effects of dynamic adaptation difficult to predict because of the dynamically changing dampening factor.

### 9.2.6 Accelerometer-Only and Sensor Fusion Tilt Interaction

The use of sensor fusion for performing tilt interaction was investigated in Chapter 6 in order to determine whether incorporating additional sensors could address the controllability
shortcomings of tilt interaction while walking. A simple sensor fusion algorithm, originally
developed for a robot-balancing application, was used to combine accelerometer, digital
compass and gyroscope data in MapExplorer. Accelerometer-only tilt interaction was
compared to the sensor fusion approach using a user study described in Chapter 6.
Participants were required to complete locating and navigating tasks in seated and walking
contexts of use. The aim of the user study was to determine whether the use of sensor fusion
resulted in significant improvements in terms of efficiency and controllability (both perceived
and actual) in a mobile map-based application.

The results of the user study showed that participants were more efficient using the sensor
fusion approach while walking. Overall task times were also quicker for the sensor fusion
technique (Table 6.1). Participants were able to complete locating tasks quicker using the
sensor fusion implementation (Table 6.2). Participants were unable to perform navigation
tasks with more accuracy using sensor fusion (Table 6.3). Sensor fusion resulted in a
statistically significant improvement for locating tasks performed while walking, with a
shorter mean time recorded between entering the boundary region of target icons and
selection operations being performed (Table 6.5).

Analysis of log data recorded during the user study showed that pitch and roll angles
recorded using sensor fusion exhibited significantly less variance than those recorded using
the accelerometer-only implementation (Figure 6.8). This trend wasn’t quite as clear when
comparing the variance in map position (Figure 6.10). This was partly due to the
implementation of discretisation in the accelerometer-only implementation, which would
have smoothed out much of the disturbance as a result of walking motion in the original
signal. A much smoother trend was observed for the orientation angles calculated using the
sensor fusion approach (Figure 6.9).

Participants were asked to indicate which technique (accelerometer-only or sensor fusion)
was easier to control for each task type (locating and navigating) in each context of use
(seated and walking). The mean ratings received indicated a statistically significant
preference for sensor fusion for both task types in a walking context of use (Table 6.7).

### 9.2.7 Proposed Framework

The objective of Chapter 8 was to propose a framework for the design of enhanced tilt
interaction. This framework incorporated six design recommendations for tilt interaction
which were proposed based on the experimental results obtained during this research. These design recommendations are listed in Table 9.1.

<table>
<thead>
<tr>
<th>#</th>
<th>Design Recommendation</th>
<th>Chapter</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Use attractor mechanisms to allow for selection using approximate cursor positioning</td>
<td>5</td>
</tr>
<tr>
<td>R2</td>
<td>Reduce tilt interaction sensitivity by a constant dampening factor when the user is mobile</td>
<td>5, 7</td>
</tr>
<tr>
<td>R3</td>
<td>Use visual and vibrotactile feedback to improve perceived controllability and reduce mental demand</td>
<td>5</td>
</tr>
<tr>
<td>R4</td>
<td>Use sensor fusion to improve perceived controllability and efficiency when the user is mobile</td>
<td>6</td>
</tr>
<tr>
<td>R5</td>
<td>Use rate controlled tilt zooming to improve efficiency and perceived controllability for zooming operations</td>
<td>7</td>
</tr>
<tr>
<td>R6</td>
<td>Provide the user with an overview when panning quickly</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 9.1. Design recommendations for tilt interaction in mobile map-based applications

The framework was described from both a procedural and an architectural perspective. An API facilitating the development of enhanced tilt interaction techniques was also implemented.

The Mobile Tilt Interaction Framework (MoTIF) identifies eight high-level stages in a mobile tilt interaction technique, namely:

- **Sensor Measurement**: Raw sensor data is accessed in an asynchronous fashion.
- **Pre-process Data**: Sensor data is smoothed to eliminate noise and signal disturbance.
- **Feature Extraction**: Sensor fusion is used to convert pre-processed sensor data into pitch and roll angles which can be used for tilt interaction.
- **Map Tilt onto Panning and Zooming Operations**: Pitch and roll angles calculated during the pre-processing stage are translated into panning and zooming speeds. Discretisation is used to map angle intervals onto panning and zooming speeds.
- **Perform Panning**: The current panning speed is used to perform panning operations.
- **Perform Zooming**: The current zooming speed is used to perform zooming operations.
- **Perform Selection**: Selection operations are performed using sensor-based (e.g. shake selection) or traditional techniques (keypad or touch).
• **Update Display:** The display is updated to reflect panning, zooming and selection changes.

The techniques proposed in this research that produced positive results during the user studies were incorporated into MoTIF. The sensor measurement, pre-processing and feature extraction stages were modified to incorporate sensor fusion, based on the positive results of Chapter 6. The mapping between tilt angles and panning and zooming operations was modified to include sensitivity adaptation, rate controlled tilt zooming and the use of an overview. All these features produced positive results in the user studies described in Chapters 5 and 7. The panning stage was modified to include attractors, visual and vibrotactile feedback, which were positively received by participants in the user study described in Chapter 5. Attractors and visual feedback both share a common sub-stage which involves determining the proximity of the cursor to POIs on the map. The only addition to the zooming stage was the use of visual feedback to provide gesture recognition acknowledgement and an indication of the current zoom level (successfully employed in the user study described in Chapter 7). The selection stage was modified to include dwell-time selection as an alternative to touch-screen selection based on the results of Chapter 5. Updating of the display was modified to include the display of visual and vibrotactile feedback for panning and zooming and the display of an overview (if activated).

MoTIF was also described from an architectural perspective. Three components were identified as necessary to implement the techniques proposed in this research. A Sensor Data Provider component was proposed to provide a single point of access to sensor data and to manage the sensor measurement, pre-processing, feature extraction and mapping stages. A Context Calculator component was proposed to continuously calculate the user’s current context for the purpose of sensitivity adaptation. A Position Listener was proposed to implement position-related effects such as attractors, vibrotactile feedback and dwell-time selection.

### 9.3 Summary of Contributions

This research has made the following contributions to the study of sensor-based interaction techniques for mobile map-based applications:

• A prototype mobile map-based application incorporating tilt interaction;
Chapter 9: Conclusions

- Identification of usability shortcomings of existing sensor-based interaction techniques;
- Proposal of enhanced tilt interaction techniques;
- Experimental results regarding the evaluation of the proposed techniques;
- Design recommendations regarding the use of tilt interaction; and
- A framework for the design of enhanced tilt interaction techniques.

A prototype mobile map-based application, called MapExplorer, was developed and used for experimentation purposes throughout this research. This prototype was implemented in both Java ME and Android, and illustrates the feasibility of tilt interaction using currently available hardware and development platforms. The development and use of MapExplorer also illustrated that tilt interaction can effectively support users in completing typical mobile map-based tasks.

The main objective of this research was to enhance current sensor-based interaction techniques, with a particular focus on the domain of mobile map-based applications. In order to enhance sensor-based interaction, it was necessary to identify problems and shortcomings of existing techniques. Through a comparative evaluation between tilt and keypad interaction involving typical map-based tasks, several shortcomings of tilt interaction were identified. Some of these, such as controllability, are well-known and confirm existing research on sensor-based interaction (Rahman et al. 2009). Problems such as high mental demand, concerns regarding practicality in a mobile context of use and the controllability of zooming techniques are less well-known.

This research proposed several techniques to enhance basic sensor-based interaction and address the shortcomings identified. The static and dynamic adaptation techniques are to the best of the author’s knowledge novel to this domain and represent some of the first research to address the usability shortcomings of tilt interaction in a mobile context of use. Several other techniques, such as attractors and the use of a semi-transparent overview, were applied for the first time in conjunction with tilt interaction in a mobile map-based application. Several techniques were also combined in novel ways. The use of gestures to perform zooming in conjunction with tilt-controlled panning was shown to allow for explicit one-handed control over both panning and zooming. The use of tilt to explicitly control both panning and zooming has not previously been employed and experimental results showed this approach to be particularly successful.
The experimental results obtained during this research demonstrate that several of the proposed techniques provided significant benefits for tilt interaction. Sensitivity adaptation was shown to provide controllability benefits for walking tasks without compromising on efficiency. The implication of this is that tilt interaction can be made more sensitive to minimise screen visibility problems and maximise efficiency during seated tasks, with decreased sensitivity facilitating improved accuracy during walking tasks.

A significant emphasis was placed on evaluating zooming techniques used in conjunction with tilt interaction in this research. Experimental results showed that participants liked the sense of overview provided by SDAZ, but disliked the loss of control over the current zooming level. Gesture zooming was first proposed, as this allowed for panning and zooming to be performed separately without the need for a clutch mechanism. Experimental results showed that while this technique was marginally preferred over SDAZ, participants found it uncomfortable and the initial implementation suffered from gesture recognition errors. Furthermore, the need for multiple gestures to traverse multiple zoom levels resulted in a frustrating user experience. Tilt zooming was found to work best using a rate controlled approach to facilitate control over zooming speed and easy switching between panning and zooming modes. The use of a touch-screen clutch control to facilitate zooming was preferred over a physical button, but this may have been as a result of the ergonomics of the device used. In order to mitigate the loss of the overview feature provided by SDAZ, a semi-transparent overview was added when panning quickly. Experimental results showed that participants found this feature useful for locating POIs.

The use of attractors has previously proven successful in conjunction with tilt interaction in the domain of photo-browsing (Cho et al. 2007b). This research extended this technique to aid in the selection of POI icons in a mobile map-based application. Indexing of POIs based on geographic location was used to ensure the efficient operation of this technique. Experimental results showed that the implementation of attractors proved highly effective at improving selection accuracy, reducing overshooting errors and reducing the time between entering an icon’s boundary region and settling the cursor for selection. Subjective feedback was also very positive for the use of this technique. Similar positive results were obtained for the use of visual and vibrotactile feedback, with participants perceiving both to be very useful.
The investigation into the use of sensor fusion was motivated by the rapid increase in the number of mobile phones equipped with a digital compass and a gyroscopic sensor. The comparative user study between accelerometer-only and sensor fusion implementations of tilt interaction is the first of its kind and provides interesting insights into the relative merits of the two approaches. The algorithm used to perform sensor fusion demonstrated an efficient means of incorporating the benefits of multi-sensor input in a mobile application. The results obtained confirm the theoretical benefits of sensor fusion in a mobile context of use (Invensense 2010). The results also show that sensor fusion alone is not a complete solution to the controllability problems of tilt interaction. Additional interventions (such as sensitivity adaptation) are necessary to improve usability in a mobile context of use.

The majority of existing research regarding sensor-based interaction has involved experimentation with users seated at a desk in a lab environment. The inclusion of walking tasks in this research helped to identify problems and benefits related to the use of tilt interaction in a mobile context of use. For example, usability issues regarding the use of SDAZ for walking tasks were identified, while gesture zooming was found to be easier to use for walking tasks than for seated tasks. The benefits of sensor fusion, not noticeable for seated tasks, were clearly evident during walking tasks. Furthermore, tilt zooming suffered from a significant decrease in ease of use during walking tasks as compared to seated tasks, as the effects of walking motion reduced participants’ level of control.

The results of the various user studies that were conducted provided valuable insight into the design of tilt interaction techniques. The results presented in Chapters 5-7 were used to propose several design recommendations for the use of tilt interaction in mobile map-based applications. These recommendations are summarised in Table 8.1. Based on the results of this research, the use of sensitivity adaptation, attractors, visual and vibrotactile feedback, sensor fusion, overviews and tilt zooming can all be recommended.

The proposed MoTIF framework extends the existing stages currently employed in tilt interaction by including those techniques which yielded positive results in this research. The framework helps to outline the specific stages involved in tilt interaction and indicates how the results of this research can be integrated into this process. The framework also proposes an application architecture which allows for the easy implementation of the techniques proposed in this research. Furthermore, the framework was successfully implemented as an Android API and used in the development of two prototype applications. The successful use
of this API indicates that the future development of applications involving sensor-based interaction could benefit from extensions to the standard sensor APIs included in the major mobile development platforms.

**9.4 Limitations and Problems Encountered**

The participant profile used in all the user studies conducted in this research consisted of senior students in the Department of Computing Sciences at NMMU. These students are predominantly in a similar age range (20-29) and are experienced in the use of mobile phones. Many of the participants had previous experience with some form of sensor-based interaction, whether in the form of mobile applications, games or gaming consoles. As a result, the results in this research cannot be generalised beyond this participant profile. However, given that this is the profile of users likely to make use of sensor-based interaction techniques in the future, this participant profile is not unsuitable.

A limitation of this research was that the techniques were implemented on specific mobile devices (a Nokia N97 and a Samsung Nexus S) and particular implementation platforms (Java ME and Android). As a result, specific ergonomic features of these devices may have to some degree influenced the results obtained. Furthermore, issues such as sensor sampling rates and units of measurement were restricted by the data provided by the specific platforms. Performance issues like frame rates, which would have affected user experience, are also dependent on the hardware of the specific phones. Nevertheless, the devices used represent popular smart phones at the time of use and were therefore considered representative of typical sensor-enabled mobile phones.

Gyroscope sensors on mobile phones were introduced into the mass market during the course of this research. Furthermore, development of sensor-based interaction techniques has continued both by researchers and by mobile handset manufacturers. Several manufacturers have recently integrated sensor-based interaction to a limited degree into photo gallery, browser and map applications. Every effort was made to keep up with the latest developments and the user study on sensor fusion was included to address this development in the mobile phone landscape.

**9.5 Recommendations for Future Work**

The results of the user study in Chapter 7 identified that participants found the effects of dynamic adaptation difficult to predict because of rapid changes in the dampening factor
employed. This problem could be addressed by employing either longer sampling windows (during which acceleration variance is likely to be more constant), smoothing or an inertia weight to reduce the rate at which the current dampening factor changes. There was also some indication during the user study in Chapter 7 that the use of personalised dampening factors could be useful, with a few participants with prior tilt interaction experience indicating a preference for the more sensitive setting. The use of a supervised learning approach, where the system learns from users’ changes to the current sensitivity setting in different contexts of use, could be employed. Alternatively, appropriate dampening factors could be learned by observing how accurately users are able to control tilt interaction in different contexts of use.

Sensitivity adaptation could also be extended to consider other contexts of use. The neural network learning approach employed could easily be extended to classify additional contexts such as climbing stairs, running or sitting in a car. Appropriate dampening factors for these contexts would also have to be investigated.

The results of Chapter 7 also indicated that the rate controlled tilt zooming technique suffered from a statistically significant decrease in ease of use for walking tasks as compared to seated tasks. This indicates that this technique could also benefit from the use of sensitivity adaptation in walking contexts of use.

The user study described in Chapter 7 showed very positive results regarding rate controlled tilt zooming as opposed to gesture zooming. The use of a separate, semi-transparent overview was also rated favourably by participants. An interesting extension to this research would be to compare this approach with the Semi-Automatic Zooming technique, which has also been shown to achieve positive results in a mobile map-based application (Kratz et al. 2010).

The relative computational costs of the proposed techniques during the course of this research were not rigorously evaluated. While no obvious performance differences were observed during the various evaluations, more thorough analysis is required to draw definitive conclusions in this regard. The computational requirements of the different techniques are an important consideration in mobile applications, which must share limited resources and battery life.

The experiments conducted during this research were all conducted indoors over short periods of time (typically around 45 minutes to an hour). The use of a longitudinal study,
examining the use of tilt interaction over long time periods and in different contexts of use, would provide further insight into the techniques proposed in this research (Yim et al. 2011). Given that most participants in this research were either novices or inexperienced with tilt interaction, different trends may be observed over prolonged periods of use. Additional usability problems could be uncovered in other contexts of use not considered in this research.

The sensor-based interaction techniques proposed in this research were applied to the domain of mobile map-based applications. Mobile map-based applications have much in common with many other common mobile applications where users must browse two-dimensional spaces at multiple zoom levels. Applications which allow users to explore multimedia collections, browse the internet and interact with documents all require navigation of large information spaces with precise control. It is therefore likely that several of the techniques proposed in this research would be beneficial in other domains. Techniques such as sensitivity adaptation, tilt zooming and attractors all have the potential to make it easier to use sensor-based interaction techniques to interact with a wide range of applications while mobile. An interesting extension of this research would be a comparison between tilt and touch interaction. The use of hybrid interaction techniques incorporating touch and tilt also provides the potential to incorporate the benefits of both techniques (Hinckley and Song 2011). This research largely employed sensor-based interaction techniques independently of more traditional interaction techniques such as touch and keypad interaction. In the design process, however, the need to make use of other modalities such as touch and physical buttons often became apparent. Touch interaction was used to toggle tilt interaction and both touch and physical buttons were used as a clutch mechanism to allow the use of sensor-based interaction for more than one type of operation. It is unlikely that sensor-based interaction techniques such as tilt interaction will take over from touch interaction as the dominant mobile interaction technique. Tilt interaction, however, have obvious intuitive benefits for domains that require browsing large information spaces, such as mobile map-based applications. By minimising the shortcomings of sensor-based interaction using techniques such as those proposed in this research, sensor-based interaction could mature into more widespread use.
References


APPLE INC. (2011a). iOS Developer Library - Motion Events [Online].


# Appendix A: Informed Consent Form

## NELSON MANDELA METROPOLITAN UNIVERSITY

### INFORMATION AND INFORMED CONSENT FORM

<table>
<thead>
<tr>
<th>RESEARCHER’S DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title of the research project</strong></td>
</tr>
<tr>
<td><strong>Reference number</strong></td>
</tr>
<tr>
<td><strong>Principal investigator</strong></td>
</tr>
<tr>
<td><strong>Contact telephone number</strong></td>
</tr>
</tbody>
</table>

## A. DECLARATION BY OR ON BEHALF OF PARTICIPANT

<table>
<thead>
<tr>
<th>I, the participant and the undersigned (full names)</th>
</tr>
</thead>
</table>

## A.1 HEREBY CONFIRM AS FOLLOWS:

| I, the participant, was invited to participate in the above-mentioned research project that is being undertaken by Bradley van Tonder from Department of Computing Sciences of the Nelson Mandela Metropolitan University. |

## THE FOLLOWING ASPECTS HAVE BEEN EXPLAINED TO ME, THE PARTICIPANT:

<table>
<thead>
<tr>
<th><strong>2.1 Aim:</strong></th>
<th>The investigators are studying interaction techniques for mobile map-based applications. The information will be used to/for research purposes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2.2 Confidentiality:</strong></td>
<td>My identity will not be revealed in any discussion, description or scientific publications by the investigators.</td>
</tr>
<tr>
<td><strong>2.3 Access to findings:</strong></td>
<td>Participants can access findings on request.</td>
</tr>
<tr>
<td><strong>2.4 Voluntary participation / refusal / discontinuation:</strong></td>
<td>My participation is voluntary YES NO My decision whether or not to participate will in no way affect my present or future care / employment / lifestyle TRUE FALSE</td>
</tr>
</tbody>
</table>

| 3. | No pressure was exerted on me to consent to participation and I understand that I may withdraw at any stage without penalisation. |
4. Participation in this study will not result in any additional cost to myself.

A.2 I HEREBY VOLUNTARILY CONSENT TO PARTICIPATE IN THE ABOVE-MENTIONED PROJECT:

<table>
<thead>
<tr>
<th>Signed/confirmed at</th>
<th>on</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Signature of witness:

Full name of witness:

Signature
Appendix B: Pre-Test Biographical Questionnaire

Nelson Mandela Metropolitan University
Department of Computing Sciences
This questionnaire is part of research towards a PhD in Computer Science and Information Systems

<table>
<thead>
<tr>
<th>Pre-Test Questionnaire</th>
<th>Biographical Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Gender</td>
<td>Male</td>
</tr>
<tr>
<td>2. Dominant hand</td>
<td>Right</td>
</tr>
<tr>
<td>3. Age</td>
<td>18-19</td>
</tr>
<tr>
<td>4. Mobile phone experience</td>
<td>years</td>
</tr>
<tr>
<td>5. How often do you use map-based software on a mobile phone?</td>
<td>Never</td>
</tr>
<tr>
<td>6. Do you have experience using tilt input (e.g. tilt-controlled mobile applications)?</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Appendix C: Post-Task Questionnaire (Chapter 4)

_Nelson Mandela Metropolitan University_  
_Department of Computing Sciences_

This questionnaire is part of research towards a PhD in Computer Science and Information Systems

<table>
<thead>
<tr>
<th>Post-Task Questionnaire</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Section A: Perceived Workload</strong></td>
</tr>
<tr>
<td>1. Mental Demand: How mentally demanding was this set of tasks?</td>
</tr>
<tr>
<td>Very Low</td>
</tr>
<tr>
<td>2. Physical Demand: How physically demanding was this set of tasks?</td>
</tr>
<tr>
<td>Very Low</td>
</tr>
<tr>
<td>3. Temporally Demand: How hurried or rushed was the pace of this set of tasks?</td>
</tr>
<tr>
<td>Very Low</td>
</tr>
<tr>
<td>4. Performance: How successful were you in accomplishing what you were asked to do?</td>
</tr>
<tr>
<td>Perfect</td>
</tr>
<tr>
<td>5. Effort: How hard did you have to work to accomplish your level of performance?</td>
</tr>
<tr>
<td>Very Low</td>
</tr>
<tr>
<td>6. Frustration: How insecure, discouraged, irritated, stressed and annoyed were you?</td>
</tr>
<tr>
<td>Very Low</td>
</tr>
<tr>
<td><strong>Section B: User Satisfaction</strong></td>
</tr>
<tr>
<td>1. I am satisfied with the ease of completing this set of tasks</td>
</tr>
<tr>
<td>Strongly Disagree</td>
</tr>
<tr>
<td>2. I am satisfied with the amount of time it took to complete this set of tasks</td>
</tr>
<tr>
<td>Strongly Disagree</td>
</tr>
<tr>
<td>3. I am satisfied that the interaction technique was easy to control for this set of tasks</td>
</tr>
<tr>
<td>Strongly Disagree</td>
</tr>
<tr>
<td>4. I am satisfied that the interaction technique was easy to use for this set of tasks</td>
</tr>
<tr>
<td>Strongly Disagree</td>
</tr>
</tbody>
</table>
Appendix D: Post-Test Questionnaire (Chapter 4)

Nelson Mandela Metropolitan University
Department of Computing Sciences

This questionnaire is part of research towards a PhD in Computer Science and Information Systems

<table>
<thead>
<tr>
<th>Section A: Preferred Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Which interaction technique did you prefer?</td>
</tr>
<tr>
<td>Tilt</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section B: Tilt Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tilt interaction was easy to control</td>
</tr>
<tr>
<td>Strongly disagree</td>
</tr>
<tr>
<td>Tilt interaction allowed me to quickly complete tasks</td>
</tr>
<tr>
<td>Strongly disagree</td>
</tr>
<tr>
<td>Tilt interaction allows me to correctly complete tasks</td>
</tr>
<tr>
<td>Strongly disagree</td>
</tr>
<tr>
<td>Tilt interaction was easy to use</td>
</tr>
<tr>
<td>Strongly disagree</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section C: Keypad Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keypad interaction was easy to control</td>
</tr>
<tr>
<td>Strongly disagree</td>
</tr>
<tr>
<td>Keypad interaction allowed me to quickly complete tasks</td>
</tr>
<tr>
<td>Strongly disagree</td>
</tr>
<tr>
<td>Keypad interaction allows me to correctly complete tasks</td>
</tr>
<tr>
<td>Strongly disagree</td>
</tr>
<tr>
<td>Keypad interaction was easy to use</td>
</tr>
<tr>
<td>Strongly disagree</td>
</tr>
</tbody>
</table>

Section D: General Feedback
## Appendix D: Post-Test Questionnaire (Chapter 4)

1. **What is the most positive aspect of keypad interaction?**

2. **What is the most negative aspect of keypad interaction?**

3. **What is the most positive aspect of tilt interaction?**

4. **What is the most negative aspect of tilt interaction?**
Appendix E: Post-Task Questionnaire (Chapter 5)

*Nelson Mandela Metropolitan University*  
*Department of Computing Sciences*

This questionnaire is part of research towards a PhD in Computer Science and Information Systems

## Post-Task Questionnaire

### Section A: Perceived Workload

<table>
<thead>
<tr>
<th></th>
<th>Mental Demand: How mentally demanding was this set of tasks?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Very Low 1 2 3 4 5 6 7 Very High</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Physical Demand: How physically demanding was this set of tasks?</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Temporally Demand: How hurried or rushed was the pace of this set of tasks?</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Performance: How successful were you in accomplishing what you were asked to do?</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Effort: How hard did you have to work to accomplish your level of performance?</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Frustration: How insecure, discouraged, irritated, stressed and annoyed were you?</td>
<td></td>
</tr>
</tbody>
</table>

### Section B: User Satisfaction

<table>
<thead>
<tr>
<th></th>
<th>I am satisfied with the ease of completing this set of tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Strongly Disagree 1 2 3 4 5 6 7 Strongly Agree</td>
</tr>
<tr>
<td>2</td>
<td>I am satisfied with the amount of time it took to complete this set of tasks</td>
</tr>
<tr>
<td>3</td>
<td>I am satisfied that the interaction technique was easy to control</td>
</tr>
<tr>
<td>4</td>
<td>I am satisfied that the interaction technique was easy to use</td>
</tr>
<tr>
<td>5</td>
<td>I am satisfied that accurate selection was easy using this interaction technique</td>
</tr>
<tr>
<td>6</td>
<td>I am satisfied that the interaction technique was easy to use while walking</td>
</tr>
</tbody>
</table>
Appendix F: Post-Test Questionnaire (Chapter 5)

Nelson Mandela Metropolitan University  
Department of Computing Sciences

This questionnaire is part of research towards a PhD in Computer Science and Information Systems

<table>
<thead>
<tr>
<th>Post-Test Questionnaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please note:</td>
</tr>
<tr>
<td>“Technique 1” refers to the first interaction technique you used</td>
</tr>
<tr>
<td>“Technique 2” refers to the second interaction technique you used</td>
</tr>
</tbody>
</table>

Put a cross in the appropriate block

### Section A: Preferred Technique

1. Which interaction technique did you prefer while seated?

<table>
<thead>
<tr>
<th>Technique 1</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Technique 2</th>
</tr>
</thead>
</table>

2. Which interaction technique did you prefer while walking?

<table>
<thead>
<tr>
<th>Technique 1</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Technique 2</th>
</tr>
</thead>
</table>

3. Which zooming technique did you prefer?

<table>
<thead>
<tr>
<th>Technique 1</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Technique 2</th>
</tr>
</thead>
</table>

4. State reasons for your preferences

### Section B: Feedback

5. Vibration feedback was useful for performing selections

<table>
<thead>
<tr>
<th>Strongly disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Strongly agree</th>
</tr>
</thead>
</table>

6. Visual feedback (arrows indicating speed and direction) was useful for controlling panning speed and direction

<table>
<thead>
<tr>
<th>Strongly disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Strongly agree</th>
</tr>
</thead>
</table>

### Section C: General

1. State one positive aspect of technique 1

---

Nelson Mandela Metropolitan University

for tomorrow
2. State one negative aspect of technique 1

3. State one positive aspect of technique 2

4. State one negative aspect of technique 2
Appendix G: Post-Test Questionnaire (Chapter 6)

Nelson Mandela Metropolitan University
Department of Computing Sciences

This questionnaire is part of research towards a PhD in Computer Science and Information Systems

<table>
<thead>
<tr>
<th>Post-Task Questionnaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>Put a cross in the appropriate block</td>
</tr>
</tbody>
</table>

**Section A: Controllability**

1. While **seated**, which technique was easier to control for **locating tasks**?
   
   | Technique 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Technique 2 |

2. While **seated**, which technique was easier to control for **navigating tasks**?
   
   | Technique 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Technique 2 |

3. While **walking**, which technique was easier to control for **locating tasks**?
   
   | Technique 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Technique 2 |

4. While **walking**, which technique was easier to control for **navigating tasks**?
   
   | Technique 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Technique 2 |

**Section B: General Comments.** Provide any general comments below.
Appendix H: Post-Task Questionnaire – Zooming Techniques (Chapter 7)

Nelson Mandela Metropolitan University
Department of Computing Sciences
This questionnaire is part of research towards a PhD in Computer Science and Information Systems

Post-Task Questionnaire

<table>
<thead>
<tr>
<th>Section A: Perceived Workload</th>
<th>Very Low</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Very High</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Mental Demand: How mentally demanding was this zooming technique?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Physical Demand: How physically demanding was this zooming technique?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Temporally Demand: How hurried or rushed was the pace of this zooming technique?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Performance: How successful were you in accomplishing what you were asked to do?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Effort: How hard did you have to work to accomplish your level of performance?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Frustration: How insecure, discouraged, irritated, stressed and annoyed were you?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section B: User Satisfaction</th>
<th>Strongly Disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Strongly Agree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I am satisfied with the ease of completing the tasks using this zooming technique</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. I am satisfied with the amount of time it took to complete the tasks using this zooming technique</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. I am satisfied that this zooming technique was easy to use for these tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. I was able to accurately control this zooming technique for these tasks</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Section C: Overview

1. I found the overview in the top-right corner (last task) useful

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Strongly Agree</th>
</tr>
</thead>
</table>

---
Appendix I: Post-Test Questionnaire – Zooming Techniques (Chapter 7)

Nelson Mandela Metropolitan University
Department of Computing Sciences
This questionnaire is part of research towards a PhD in Computer Science and Information Systems

<table>
<thead>
<tr>
<th>Post-Test Questionnaire</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please note:</td>
</tr>
<tr>
<td>“Technique 1” refers to the first zooming technique you used</td>
</tr>
<tr>
<td>“Technique 2” refers to the second zooming technique you used</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section A: Preferred Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Which zooming technique did you prefer while seated?</td>
</tr>
<tr>
<td>Technique 1</td>
</tr>
<tr>
<td>2. Which zooming technique did you prefer while walking?</td>
</tr>
<tr>
<td>Technique 1</td>
</tr>
<tr>
<td>3. Which zooming technique did you prefer overall?</td>
</tr>
<tr>
<td>Technique 1</td>
</tr>
<tr>
<td>4. Which method of activating zooming did you prefer?</td>
</tr>
<tr>
<td>Touch-Screen Button</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section B: General</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. State one positive aspect of technique 1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>2. State one negative aspect of technique 1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>3. State one positive aspect of technique 2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>4. State one negative aspect of technique 2</td>
</tr>
</tbody>
</table>
Appendix J: Post-Task Questionnaire – Sensitivity Adaptation (Chapter 7)

Nelson Mandela Metropolitan University
Department of Computing Sciences
This questionnaire is part of research towards a PhD in Computer Science and Information Systems

<table>
<thead>
<tr>
<th>Section A: User Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I am satisfied with the ease of completing the tasks</td>
</tr>
<tr>
<td>Strongly Disagree</td>
</tr>
<tr>
<td>2. I am satisfied with the amount of time it took to complete this set of tasks</td>
</tr>
<tr>
<td>Strongly Disagree</td>
</tr>
<tr>
<td>3. I am satisfied that tilt interaction was easy to use for this set of tasks</td>
</tr>
<tr>
<td>Strongly Disagree</td>
</tr>
<tr>
<td>4. I was able to accurately control tilt interaction for this set of tasks</td>
</tr>
<tr>
<td>Strongly Disagree</td>
</tr>
</tbody>
</table>