The Design of a New Musical Glove: A Live Performance Approach.

Giuseppe Torre
Digital Media & Arts Research Centre
Faculty of Science and Engineering
Department of Computer Science and Information Systems
University of Limerick

Submitted to the University of Limerick for the degree of
Philosophiae Doctor (PhD) 2013
1. Supervisor: Dr. Kerry Hagan  
   Digital Media & Arts Research Centre  
   University of Limerick  
   Ireland

2. Supervisor: Dr. Mikael Fernström  
   Interaction Design Centre  
   University of Limerick  
   Ireland

1. Reviewer: Dr. Dominic Murcott  
   Trinity Laban Conservatoire of Music  
   London  
   United Kingdom

2. Reviewer: Dr. Paul Stapleton  
   Sonic Arts Research Centre  
   Queen’s University Belfast  
   Northern Ireland

1. Chair: Dr. Chris Exton  
   Dept. of Computer Science and Information Systems  
   University of Limerick  
   Ireland

Day of the defense: 17th December 2012

Signature from head of PhD committee:
Abstract

A live performance using novel technologies is a highly complex system in which anthropological, sociological, psychological, musicological and technical issues are heavily involved. The New Interfaces for Musical Expression (NIME) community has presented a new approach to music performance often heavily technologically mediated while outputting a great deal of new digital instruments since 2001. Within this broad research field, important issues such as hardware interface design, mapping strategies, skilled performance and compositional approaches have been considered. Many NIME practitioners have explored the development of ‘gestural controllers’ in the hope of achieving natural and intimate interaction while also designing clear interactions between performer’s gesture and sound from an audience perspective. This thesis expands on this notion through the consideration of the possibilities for enhancing the audience engagement and understanding of the underlaying structures and mechanics of the live performance.

To this end, a newly developed data glove named Pointing-at is developed. A number of live performances in which the data glove is used are presented and discussed. The analysis of both the theoretical and practical elements of the research formed the basis for the development of an approach to the design of nuanced gestural performative actions that are both visually and sonically legible to audience members. In that regard, the use of metaphors that are coherent to the theme of the performance have been found to be a useful tool that can enhance both the performer and audience experience of the live performance.
Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This paper has not previously been presented in identical or similar form to any other Irish or foreign examination board.

The thesis work was conducted from 2007 to 2012 under the supervision of Dr. Kerry Hagan and Dr. Mikael Fernström at University of Limerick.

Limerick, 2013
Acknowledgements

I am indebted to many people for the development of this research. They all have played a fundamental role in this long journey. My warmest and most sincere thanks go to:

My supervisors: Dr. Kerry Hagan and Dr. Mikael Fernström for their guidance, support and patience. My previous supervisor Dr. Annette Aboulafia for helping me understand what PhD research means.

My performance partners: Robert Sazdov and Dorota Konczweska.

My colleagues in the CSIS Department and Ms. Annette McElligott for moral and financial support. My third unofficial ‘supervisor’ Nicholas Ward for all the interesting and grumpy debates had and advice given. Anthony Quinn for proof-reading this dissertation.

The SFI-funded Tyndall National Access Program (NAP60) and all the Tyndall team participating to the Celeritas project: Philip Angove, Javier Torres, Brendan O’Flynn. The IDC people involved in the Celeritas Project: Stephen Shirley, Marc McLoughlin, Conor Higgins and Colm McGettrick.

The NIME Community for stimulating many of the reflections in this dissertation.

To Caitriona Kinsella and my friends. All of you. Distant or close (geographically): thanks.

My family: Maurizio, Liliana, Zio Giovanni e Zia Giusy for just being who they are.

My parents Diego and Albina for just being who they were.
... ‘Every one who is sure of his mind, or proud of his office, or anxious about his duty assumes a tragic mask. He deputes it to be himself and transfers to it almost all his vanity. While still alive and subject, like all existing things, to the undermining flux of his own substance, he has crystallized his soul into an idea, and more in pride than in sorrow he has offered up his life...by conscience into loyalties and duties, and we become “persons” or masks’. Art, truth and death turn everything to marble. (Santayana 1922 pp. 133-134)
Contents

List of Figures vii

1 Introduction 1
  1.1 Aims and Objectives ............................................. 4
  1.2 Methodology ....................................................... 5
  1.3 Research Contribution ........................................... 7
  1.4 Thesis Outline ...................................................... 8

2 NIME Live Performance Ecology - Composer and Audience Perspectives 11
  2.1 NIME Live Performance Ecology .................................. 11
    2.1.1 Gestures in Live Performances ................................. 13
    2.1.2 Constraints .................................................... 15
    2.1.3 Affordances .................................................... 18
    2.1.4 Instrument or Controller? ..................................... 20
    2.1.5 Mapping - Composer and Audience Perspectives ............. 23
    2.1.6 Functions and Frames .......................................... 27
  2.2 Summary ............................................................. 29

3 Survey of Data Gloves and their use in Live Performances 31
  3.1 Glove Devices ....................................................... 31
  3.2 Glove Devices for Music Performances ............................ 40
  3.3 Data gloves - Analysis of Live Performances ...................... 55
    3.3.1 The Hands ...................................................... 55
    3.3.1.1 Discussion .................................................. 57
## CONTENTS

3.3.2 Lady’s Glove ........................................... 59
    3.3.2.1 Discussion ..................................... 63
3.3.3 Collide ............................................. 63
    3.3.3.1 Discussion ..................................... 65
3.3.4 Imogen Heap’s data glove ................................ 65
    3.3.4.1 Discussion ..................................... 70
3.4 Summary ............................................. 72

4 Development of Pointing-at Data Glove ............................ 73
    4.1 History and Partners involved ........................... 73
    4.2 Pointing-at Glove Design ................................. 75
    4.3 Hardware Specifications .................................. 76
    4.4 UART settings and MaxMSP Data Input ....................... 79
    4.5 Numerical Evaluation .................................... 82
        4.5.1 Sensors’ Offset and Resolution ....................... 82
        4.5.2 Bending Sensor .................................... 84
        4.5.3 Attitude Estimation - TRIAD ALGORITHM ............... 85
            4.5.3.1 Accuracy Considerations ....................... 89
    4.6 Max Sensor Interface .................................... 90
        4.6.1 Sensor Input Section ................................ 90
        4.6.2 Calibration ........................................ 90
        4.6.3 On Tour Section .................................... 91
        4.6.4 Orientation Visualisation Section ..................... 94
        4.6.5 Calculated Values Section ............................ 94
    4.7 Summary ............................................. 95

5 Performances ........................................... 97
    5.1 Live Performances ...................................... 97
    5.2 Preliminary Works ...................................... 98
        5.2.1 Discussion and Lessons Learned ....................... 101
    5.3 3D-Dj Application ...................................... 103
        5.3.1 Application Features - v.1 ......................... 103
        5.3.2 Application Features - v.2 ......................... 105
        5.3.3 Discussion ....................................... 107
CONTENTS

5.3.4 Lessons Learned .................................. 111

5.4 Rex Levitates and Pointing-at ....................... 112

5.4.1 Application Features .............................. 112

5.4.2 Discussion ........................................ 113

5.4.3 Lessons Learned .................................. 118

5.5 Molitva .................................................. 119

5.5.1 Sound Design and Score Macro-Structure .......... 119

5.5.2 Mapping Strategies ................................. 122

5.5.3 Discussion ........................................ 125

5.5.4 Lessons Learned .................................. 128

5.6 Mani ..................................................... 129

5.6.1 Sound Design and Score Macro-Structure .......... 130

5.6.2 Mapping Strategies and Pointing-at Score Part .... 131

5.6.3 Discussion ........................................ 138

5.6.4 Lessons Learned .................................. 142

5.7 ‘AGORA’ ................................................. 143

5.7.1 Concept ............................................ 143

5.7.2 Development ....................................... 146

5.7.2.1 Technical Specifications ......................... 146

5.7.2.2 Part A ........................................ 146

5.7.2.3 Part B ........................................ 150

5.7.2.4 Part C ........................................ 153

5.7.2.5 Score Macro Structure - Length and Structure of the Whole Performance ................ 153

5.7.3 Discussion ........................................ 158

5.7.4 Lessons Learned .................................. 162

5.8 Summary ............................................... 163

6 Conclusions and Future Directions ..................... 165

6.1 Summary .............................................. 165

6.2 Conclusions .......................................... 166

6.3 Contributions ......................................... 173

6.4 Future Work .......................................... 174
CONTENTS

Appendix A: WIMU Specifications 175
Appendix B: TRIAD Algorithm numerical example 181
Appendix C: Driver 184
Appendix D: Mote External and Driver 197
Appendix E: TRIAD Algorithm Code 205
Appendix F: AHRS MAX Library 208
Appendix G: AGORÁ 211
References 213
# List of Figures

2.1 Gestures .................................................. 16
2.2 Mapping Transparency ................................. 25
2.3 Functions and Frames .................................. 28

3.1 Communication Device ................................. 32
3.2 Sayre Glove ............................................. 32
3.3 Digital Data Entry Glove ......................... 33
3.4 VPL DataGlove .......................................... 34
3.5 VIEW .................................................. 35
3.6 Virtual Cockpit ......................................... 36
3.7 Power Glove ............................................ 37
3.8 Virtuality ............................................... 37
3.9 CyberGlove .............................................. 38
3.10 CyberGlove III ......................................... 39
3.11 Some data gloves ..................................... 41
3.12 Data Glove ............................................ 42
3.13 Data glove ............................................. 43
3.14 SoniMime ............................................... 44
3.15 SoniMime ............................................... 44
3.16 VIFE data glove ..................................... 45
3.17 HandySinger Glove ................................. 46
3.18 Beat Boxing ........................................... 47
3.19 MusicGlove ............................................. 48
3.20 Mims .................................................. 49
3.21 VAMP glove ........................................... 50
## LIST OF FIGURES

3.22 ForTouch ................................................. 51
3.23 P5 Glove ................................................. 52
3.24 Human-Conductor Interface ................................. 53
3.25 Beatwear ................................................. 54
3.26 eShofar Instrument ....................................... 54
3.27 The Hands - Performance ................................ 56
3.28 The Hands - Performance ................................ 57
3.29 The Hands - NIME’03 Performance ....................... 59
3.30 The Lady’s Glove ........................................ 60
3.31 The Lady’s Glove ........................................ 61
3.32 Letitia Sonami and the Lady’s Glove ..................... 62
3.33 Collide ................................................... 64
3.34 The 5DT glove ............................................. 66
3.35 Imogen Heap’s Gloves ..................................... 66
3.36 Imogen Heap’s Gesture ................................... 68
3.37 Imogen Heap’s Gestures .................................. 68
3.38 Imogen Heap’s Gestures .................................. 69
3.39 Imogen Heap’s Gestures .................................. 70
3.40 Imogen Heap’s Gesture ................................... 71
4.1 Celeritas Body Suite ....................................... 74
4.2 The earliest version of the data glove ..................... 75
4.3 Pointing-at glove design .................................. 77
4.4 Pointing-at glove design .................................. 77
4.5 Pointing-at glove ......................................... 78
4.6 Pointing-at glove ......................................... 78
4.7 25mm Wireless Inertial Measurement Unit (WIMU) .... 80
4.8 25mm WIMU ............................................... 80
4.9 Packet Length ............................................. 81
4.10 Max 5 mote patch ....................................... 82
4.11 Max 5 patch ............................................... 83
4.12 Max 5 patch ............................................... 84
4.13 Bend sensor patch. ........................................ 85
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.14</td>
<td>Orthogonal Axis System</td>
<td>86</td>
</tr>
<tr>
<td>4.15</td>
<td>Pitch, Yaw and Roll movements.</td>
<td>86</td>
</tr>
<tr>
<td>4.16</td>
<td>Max 5 patch</td>
<td>91</td>
</tr>
<tr>
<td>4.17</td>
<td><em>calibration</em> subpatch.</td>
<td>92</td>
</tr>
<tr>
<td>4.18</td>
<td>Calibration procedure</td>
<td>92</td>
</tr>
<tr>
<td>4.19</td>
<td>NOAA webpage</td>
<td>93</td>
</tr>
<tr>
<td>4.20</td>
<td>Calibration procedure</td>
<td>94</td>
</tr>
<tr>
<td>5.1</td>
<td>Preliminary Works</td>
<td>99</td>
</tr>
<tr>
<td>5.2</td>
<td>Preliminary Works</td>
<td>100</td>
</tr>
<tr>
<td>5.3</td>
<td>Preliminary Works</td>
<td>100</td>
</tr>
<tr>
<td>5.4</td>
<td>Performer’s Virtual Sphere</td>
<td>104</td>
</tr>
<tr>
<td>5.5</td>
<td>3D-Dj App v.1</td>
<td>104</td>
</tr>
<tr>
<td>5.6</td>
<td>3D-Dj App v.1</td>
<td>105</td>
</tr>
<tr>
<td>5.7</td>
<td>3D-Dj App v.2</td>
<td>106</td>
</tr>
<tr>
<td>5.8</td>
<td>3D-Dj App v.2</td>
<td>107</td>
</tr>
<tr>
<td>5.9</td>
<td>Demo at NIME 2007 (Genova, Italy)</td>
<td>108</td>
</tr>
<tr>
<td>5.10</td>
<td>3D-Dj Frames</td>
<td>111</td>
</tr>
<tr>
<td>5.11</td>
<td><em>Rex Levitates</em> - Science Gallery 2008 (Dublin - Ireland)</td>
<td>112</td>
</tr>
<tr>
<td>5.12</td>
<td><em>Rex Levitates</em></td>
<td>114</td>
</tr>
<tr>
<td>5.13</td>
<td>Embodied Generative Music</td>
<td>117</td>
</tr>
<tr>
<td>5.14</td>
<td><em>Molitva</em> - RE:New 2009 (Copenhagen - Denmark)</td>
<td>120</td>
</tr>
<tr>
<td>5.15</td>
<td><em>Molitva</em> - Macro Structure</td>
<td>121</td>
</tr>
<tr>
<td>5.16</td>
<td><em>Molitva</em> - Gesture 1</td>
<td>122</td>
</tr>
<tr>
<td>5.17</td>
<td><em>Molitva</em> - Gesture 2</td>
<td>123</td>
</tr>
<tr>
<td>5.18</td>
<td><em>Molitva</em> - Gesture 3</td>
<td>124</td>
</tr>
<tr>
<td>5.19</td>
<td><em>Mani</em> - NIME 2010 (Sydney - Australia)</td>
<td>129</td>
</tr>
<tr>
<td>5.20</td>
<td><em>Mani</em> - Macro Structure</td>
<td>130</td>
</tr>
<tr>
<td>5.21</td>
<td><em>Mani</em> - Setup</td>
<td>132</td>
</tr>
<tr>
<td>5.22</td>
<td><em>Mani</em> - Max patch.</td>
<td>133</td>
</tr>
<tr>
<td>5.23</td>
<td><em>Mani</em> - Section Speech A score</td>
<td>133</td>
</tr>
<tr>
<td>5.24</td>
<td><em>Mani</em> - Gesture 1</td>
<td>134</td>
</tr>
<tr>
<td>5.25</td>
<td><em>Mani</em> - Section Speech A - part 2 score</td>
<td>135</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.26</td>
<td><em>Mani</em> - Gesture 2</td>
<td>136</td>
</tr>
<tr>
<td>5.27</td>
<td><em>Mani</em> - Section overlap score</td>
<td>137</td>
</tr>
<tr>
<td>5.28</td>
<td><em>Mani</em> - Gesture 3</td>
<td>137</td>
</tr>
<tr>
<td>5.29</td>
<td><em>Mani</em> - Section Finale score</td>
<td>138</td>
</tr>
<tr>
<td>5.30</td>
<td><em>Mani</em> - Gesture 4</td>
<td>139</td>
</tr>
<tr>
<td>5.31</td>
<td><em>Agorá</em> - CataRT</td>
<td>144</td>
</tr>
<tr>
<td>5.32</td>
<td>MSA Visual - Video screenshot</td>
<td>147</td>
</tr>
<tr>
<td>5.33</td>
<td><em>Agorá</em> - Part A</td>
<td>149</td>
</tr>
<tr>
<td>5.34</td>
<td><em>Agorá</em> - CataRT</td>
<td>150</td>
</tr>
<tr>
<td>5.35</td>
<td><em>Agorá</em> - Part B</td>
<td>152</td>
</tr>
<tr>
<td>5.36</td>
<td><em>Agorá</em> - Macro Structure</td>
<td>155</td>
</tr>
<tr>
<td>5.37</td>
<td><em>Agorá</em> - Structure of Part A</td>
<td>155</td>
</tr>
<tr>
<td>5.38</td>
<td><em>Agorá</em> - Structure of Part B</td>
<td>157</td>
</tr>
<tr>
<td>5.39</td>
<td><em>Agorá</em> - Structure of Part C</td>
<td>157</td>
</tr>
<tr>
<td>1</td>
<td>25mm WIMU</td>
<td>176</td>
</tr>
<tr>
<td>2</td>
<td>25mm WIMU</td>
<td>177</td>
</tr>
<tr>
<td>3</td>
<td>25mm</td>
<td>178</td>
</tr>
</tbody>
</table>
1

Introduction

This thesis presents and discusses the development of a series of live music performances that include the use of a data-glove musical device. In doing so, it addresses the relationship between glove-based controllers, musical output and audience understanding/engagement in a live electronics performance scenario. The performances are designed in order to strive for the communicative efficiency of both the novel musical controller and mapping strategies adopted by the performer/composer to facilitate performance readability at the receiver (i.e. the audience).

A live performance is a human activity that has been widely investigated in the fields of anthropology, ethnomusicology, sociology, psychology and linguistics. In its basic form, it is sufficient to state that a live performance reflects the characteristics of a communication system in which a performer represents the transmitter and an audience the receiver. In a live music performance, music represents the message in its bare state. However, due to the undefined nature of any art form, it is also important to consider that the audience will often employ the help of secondary clues such as performers’ gestures, stage design and light to better understand the message or pseudo-message. In addition to this, the musician’s ability to master their instrument, perceived also through the seemingly effortless use of it, is an important feature for the proper understanding and evaluation of the overall showmanship act.

In traditional instrumental practice, the relationship between performer and instrument is straightforward in that the audience will inevitably associate the
receiving sound as coming from a given performer/instrument. This knowledge, ‘acquired through massive experience of sound-sources in general and music performances in particular’, is described by Godøy as ecological knowledge (Godøy 2010, p.106). The evaluation of their ability to control the instrument can be done on the basis of multiple factors such as touch, speed, intonation and errors (Fyans and Gurevich 2011, Fyans et al. 2009) which relates to the ability of controlling (sometimes failing to control) and investigating the almost infinite nuances that belong to the instrument. In live electronic concerts, where the performer uses devices such as a laptop and/or novel musical controllers, this relation becomes confused. The audience has little information on the degree of control the performer has on the generated musical output or even what gesture can be considered the cause for a given sound. When in the presence of multiple performers, things can become even more confused (e.g. a laptop orchestra).

Arguably, this could be traced to the fact that, contrary to a traditional musical instrument, the computer has no specific sound that could make it identifiable. Through the use of a controller, almost every possible sound or cluster of sounds can be achieved at performance time. In addition, a traditional instrument is both a controller and a sound outputting device. In contrast, what is considered an instrument in the New Interfaces for Musical Expression (NIME) community (2012), is composed of two autonomous devices: a controller and a computer.

The discussion is therefore informed by the numerous threads of debate currently active in the NIME community, which is responsible for the outputting of a large variety of musical interfaces (NIME 2012, TIEM 2010). The musical instrument presented here adds, therefore, to the long list of data glove devices developed by both the NIME community and the industry since the second half of the 20th century. The development of the vast majority of these devices (whether data gloves or other) has taken an ad hoc approach where the controller has been thought, developed and used exclusively according to the performer/composer needs. This approach, focusing on the relationship between the performer and their instrument, has pushed the discussion toward HCI issues. Orio, et al., have, for example, stated that:

...the counterpart of this creativity is the lack of commonly accepted
methodologies for the evaluation of existing developments, which pre-
vents from the comparison of different controllers and from the evalu-
ation of their performances in different musical contexts. (Orio et al.
2001 p. 1)

From this, Orio, et al., suggest that tools from the field of Human Computer
Interaction (HCI) could be borrowed to have a more systematic rather than id-
iosyncratic approach for the development of music controller and/or instruments.
In that regard, some attempts have been made in which the tendency appears
to be the evaluation of a large set of digital instruments employable in a variety
of scenario (Card et al. 1991; Vertegaal and Eaglestone 1996). More recently
Marquez-Borbon, et al. (2011, p. 373), have argued in defense ‘of the merits of
designing purpose-built devices for experimental context’. Still, the relationship
between gestures and musical intentions have been mainly investigated on the
side of simple cause and effect process (mapping) where only the performer and
its instrument are taken into account. Only a few very recent studies have focused
on the interpretative problem from the perspective of audiences (Barbosa et al.

However, as previously mentioned, a live music performance presents the basic
characteristics of a communication system in which both performer and audience
are involved. Emotional and cognitive process as well as many other elements are
somehow to be added to the equation ‘performance’ in the interest of a better
and wholly-comprehensive compositional approach.

Thus, the way the performer interprets and links the gestural input data of
their own controller to a generator and sound output device (i.e. computer),
a process known as mapping, becomes crucial in giving the audience enough
clues on what is happening at any given time during the performance. However,
we need to take into account that the mapping, being mainly a programming
problem, is always hidden to the audience and it may reveal itself only through
those gestures that enable the performer to control their instrument. Once the
cause/effect mechanism between gesture and sound becomes clear, the audience
is able to speculate on the device’s properties and limitations. This could then
1. INTRODUCTION

offer the audience an additional element to judge the performer’s skill or more generally the overall performance.

Thus, by including the element ‘audience’ in the performance ‘equation’, the discussion is able to move from a pseudo-Human Computer Interaction (HCI) domain, which specifically deals with the interaction between the performer and their instrument, to a wider scenario which involve elements of psychoacoustics, cognitive psychology, anthropology, sociology and aesthetics.

1.1 Aims and Objectives

The main question that this dissertation addresses is how to develop, compose for and perform with, newly developed digital instruments that focus on nuanced gestural performative actions that are both visually and sonically legible to audience members.

In order to address this question, this work focuses on the following two issues in the context of glove-based devices:

- Firstly, how do we design instruments that have high levels of controllability as required by the performer in order to control fine nuances of the sound producing mechanism. This is also described by Moore as the control intimacy that is the match between the variety of musically desirable sounds produced and the psycho-physiological capabilities of a practiced performer (Moore 1988 [p. 21]).

- Secondly, how do we provide the audience with a clear understanding of the relationship between the performers gestures and the sounds they create in a manner that is nuanced and non-trivial?

To such an extent, the development of a data-glove musical interface - named Pointing-at - which translates a performer’s hand gestures to musical output, is documented. The sensor technology used by the Pointing-at data glove is limited to a cluster of accelerometers, magnetometers and gyroscopes placed on the hand’s dorsum and a single bending sensor placed along the index finger only. In addition, the data coming from the single bending sensor is used solely as binary
1.2 Methodology

In order to address this question a practice-led approach was taken. An iterative process of development, composition, performance and reflection was undertaken. The knowledge produced through this process was contextualized through discussions of both the pertinent literature surrounding the NIME community, and the performance and design work of other relevant practitioners. In order to support this discussion an audio-visual documentary approach was utilised.

The development of the performances helps to iteratively develop an understanding of how the main question can be addressed in the authors performance practice.

The focus is on the NIME literature and specifically on glove-based performances. In that regard, a NIME music performance could be discussed with tools from musicology by analysing the elements that make a music composition (e.g. phrases, harmonies, structure, dynamic, methods, musical influences and references etc.). However, this approach to the analysis was only partially implemented, as its usage would not strictly address the main question this work addresses.

As stated, the primary concerns of this investigation are those of the design of gestural performative actions that enhance the perceived control intimacy from...
the performer perspective, and the clarity of the gesture to sound relationship in terms of causality from the audience perspective.

As the main subject of this dissertation is glove-based performances, a review and analysis of existing glove based performance was first undertaken. This focused on the work of Wasseil, Rovan, Sonami and Heap. A limitation in the analysis was found to be the lack of detailed material on the function of the glove based systems in the context of the live performance (i.e. how it was used). This necessitated the use of an admittedly speculative approach with regard to glove functions that emphasized the audiences perception. This, however, lead to useful outcomes in terms of informing possible avenues of enquiry in the authors own practice development.

The iterative process of development, composition, performance and reflection of the authors own performances helped frame a series of elements for the analysis, based on a discussion around some of the issues addressed in the pertinent NIME literature. The following concepts were therefore chosen to frame the analysis:

**Affordance**

The concept of affordance, permeated of defined attributes and significances from the field of cognitive psychology and HCI, is here more closely related to Godøy’s idea of ‘gestural affordances of musical sound which rests on the assumption that musical sound is a transducer of source-information made of both the actions that go into producing the sound, e.g. hitting, stroking, blowing, bowing and the material properties of the sound source, e.g plates, strings’ (Godøy 2010, p. 106). Thus in here, the modes of employment of the data-glove suggest the sound producing mechanisms while the characteristics of the outcomes (i.e. the sound generated) describes the mapping strategies or algorithms employed. The term affordance, intended then as the gestural affordance of musical glove based interactions, offers a mean to describe gestures in a NIME glove based performance scenario.

**Constraints**

Constraints, divided in technological and cultural, provide a frame for discussing the limitation the performer/composer will have to deal with when ap-
proaching the development of a performance. More in detail, the technological constrains refer to the limitations imposed by the sensor technology in use by the glove, the numerical evaluation allowed with the data available, mapping strategies and so on. The cultural constraints refer instead to the choices needed when approaching the development of a NIME performance such as the performer and audience’s cultural background, genre and theme.

**Sound Design**

The attention paid to the sound generating mechanisms leads the discussion into the reasons behind the choice of the sounds for the performance and the way they are organised and presented. For the reason exposed above however, the discussion will never offer a complete compositional analysis carried with the pertinent tools offered by the field of musicology but will rather offer a way to contextualise the choice for the sound used in the performance.

**Mapping Strategies**

The analysis of the performances approach is also described in terms of the mapping strategies employed. This offers an idea of how sounds are generated via the glove-based interaction of the performer. Mapping strategies are investigated and evaluated in relation to the main question addressed in this dissertation.

**Functions and Frames**

Lastly, Emmersons terminology of functions and frames is useful to describe and categorise specific elements of the live performance. In particular, it helps to define and describe those parts of the performance in which the perceivable audio output is the result of the performer’s interaction with their glove or generated via an automated computer event.

### 1.3 Research Contribution

The audio-visual documentary that presents the process described above, alongside the authors critical reflections, is the primary research contributions. The
1. INTRODUCTION

critical reflections then help to make the knowledge embodied in the creative process available to others.

The dissertation also documents the development of the tools that enabled this research. These are:

- Pointing-at glove, a new glove musical interface for the control of sounds in a live music performance. The device, named Pointing-at, is the result of a collaborative effort between the Tyndall National Institute of Cork and the University of Limerick.

- AHRS Max library, a new MaxMSP Library that has been named AHRS (Attitude Heading Reference System). The library contains a set of externals able to conduct a series of vector mathematical operations and their conversion to several formats. Between these, an external for the calculation of the object orientation in the physical space via accelerometers and magnetometers values has been developed. The external is based around the algorithm known as TRIAD (Bar-Itzhack and Harman 1996). The algorithm is an extremely efficient method for calculating the orientation of an object hosting any sensor cluster enabling six Degrees of Freedom (DOF). Thus, it can be utilised for other systems too.

They are not to be considered as research contributions in their own right but rather as a documentation that helped the author (and possibly others) in the research process.

1.4 Thesis Outline

The remaining chapters of this dissertation are as follows:

Chapter Two is an introductory discussion to some of the key issues surrounding a NIME performance. The concepts introduced are affordances, cultural and technological constraints, gestures, mapping strategies, control intimacy, functions and frames which are discussed from both a composer/performer and audience perspective.
Chapter Three is a historical review of data gloves developed from 1962 to present days. The last part of the chapter is dedicated to the analysis of some performances in which a data glove is used and to the description of its functionalities.

Chapter Four presents the development of the Pointing-at glove. It includes a description of the hardware, software and numerical methods employed for the successful retrieval of data.

Chapter Five describes and discusses the portfolio of performances created for the newly developed data glove. A description of the mapping strategies and key compositional issues encountered is offered. An analysis of each performance is also included.

Chapter Six draws conclusions and evaluates the performances. Lastly, it suggests possible future works.

A series of documents have been included in the Appendix section of this dissertation. These are:

- Appendix A outlines in detail the sensor cluster specifications and schematics of the Pointing-at glove.

- Appendix B presents a numerical example for the TRIAD algorithm outlined in Chapter Four.

- Appendix C includes the internal technical report written by Shirley (2006). The report describes the working mechanisms of the driver enabling the communication between the sensor cluster and the host computer.

- Appendix D presents the C++ code for the custom driver enabling communication between the sensor cluster and the host computer.

- Appendix E presents the C++ code for the ‘ahrs_triad_mote’ Max/MSP external (part of the developed AHRS Max/MSP Library).

- Appendix F is a list of all the externals developed and that are part of the AHRS Library.
1. INTRODUCTION

• Appendix G references all the music used as sound source in the development of the AGORÁ performance.

Attached to this dissertation is a CD containing the following items:

• Pointing-at: It includes the Attitude Heading Reference System (AHRS) Library which is a set of newly developed MaxMSP objects for the numerical transformation and evaluation of common operations used in 3D and 4D maths. This Library also contains the C++ code for the TRIAD algorithm and the ‘mote’ object that enabled the communication between the data-glove and the main interface. The folder also contains all the sensor node and basestation codes as well as the main Pointing-at Max 5 software interface.

• Glove design: a history photo book picturing the development of the Pointing-at glove suit (photos by Cillian O’Sullivan).

• Performances.Software: It contains all the documentation regarding the practice-led research (i.e. performances) such as Software, video footage, photos and programme notes.

• PhD-Book: It contains a Pdf version of the dissertation.

Also attached to this dissertation is a DVD containing the following items:

• A video documentary presenting a reflective narrative of the artistic processes involved for the development of the presented performances.

• All the video documentation available about the performances developed and performed.
2

NIME Live Performance Ecology
- Composer and Audience Perspectives

2.1 NIME Live Performance Ecology

An attribute often used for the description of a music performances that includes the use of electronic devices by the performer is the one of ‘live’. In that regard, Emmerson (2007) defines ‘Live’ as an act that involves:

...the presence of a human performer who takes decision and/or actions during a performance which change the real sounding nature of the music;...who produces sound mechanically; or which produces sounds on electronic substitutes for mechanical instrument using similar physical gestural input; ...who does not mechanically cause the sound, yet may cause, form or influence it through electronically mediated interfaces under their immediate control. (ibid p.90)

Interestingly, the word ‘live’ is seldom used when referring to a dance or theatrical performance as the ‘here and now’ presence and actions of the performer are assumed to be an implicit connotation of it. On the contrary, when referring to an electroacoustic music performances, the live attribute seems needed in order
2. NIME LIVE PERFORMANCE ECOLOGY - COMPOSER AND AUDIENCE PERSPECTIVES

to stress and assert (despite any reasonable doubt the audience may have) the real-time interaction and responsibility of the performer over the music material presented. Indeed, performer’s intentions are not necessarily interpreted correctly by the audience\footnote{It is not far-fetched to say that in laptop performance, the audience is required to nearly blindly believe in the live interaction elements of the performance.} as the rapid granularisation of music genres and growth of both hardware and software tools in the twentieth century have diversified the attitude of both performers and audience towards the ritual of the live performance which now seems to have lost its dialogical attributes. The vast output of the NIME community and its large amount of newly developed electronic musical instruments have enabled both composer and performer to devise new control metaphors which have inevitably led and added to the ongoing experimentation and discussion on possible ways of approaching the creation of a live music performance. This may, however, further complicate the interpretational work for the audience. As stated by Orio, et al., ‘the counterpart of this creativity is the lack of commonly accepted methodologies for the evaluation of existing developments, which prevents from the comparison of different controllers and from the evaluation of their performances in different musical contexts’ (Orio et al. 2001 [p.1]).

It is easy to state that this variety of tools and stylistic approaches has largely widened the possibilities offered to composers. However, as explained later, ‘unlimited’ freedom is not necessarily synonymous with creativity.

Addressing the NIME community, Perez, et al., (2007) distinguish between an \textit{instrument driven} and a \textit{composition driven} approach for the creation of a live performance involving the use of digital devices. The former has informed much of the music literature output by the NIME community and has the disadvantage of separating the instrument’s design process from the composition to allow for greater versatility and adaptability of the device itself for different compositions and performers. The advantage is instead that new mechanisms of interaction are explored, and this can lead to a higher degree of freedom and possibility to ‘explore’ the device during rehearsal and performance time. In the latter approach, the aesthetical ideas and decisions form the basis of the compositional discourse. The device is ultimately chosen, from many available, according to the
requirements of the given composition. The main advantage of this approach is that the device is exclusively functional to the artwork and therefore the attention is placed on the aesthetic rather than technical (the device) issues. On the other hand, the device capabilities may not be fully explored as the artwork may not require it. This is particularly relevant in the case of solo live performances.

However, these distinctions exist only within the system formed by the composer/performer and the newly developed instrument. In line with what was described earlier, we have seen that a performance is made of at least two fundamental elements: a performer and an audience. Expanding on this idea enables us to add on to the list of elements that play an important role on the development of a live NIME performance.

Most importantly, the acknowledgment of the audience presence defocuses the attention from problematics regarding the solely user-device interaction in favor of a more artistic orientated discussion in which the final aim is the communication of a message or psuedo-message to someone. As a consequence, the composer will be led to reason around a series of strategies that could facilitate the reception of the given messages by an audience. Only then the highlight of the performers’ ability (virtuosity) or the enhancement of the artistic message becomes an unnecessarily exclusive choice.

2.1.1 Gestures in Live Performances

The term ‘gesture’, derived from the Latin ‘gerere’, originally referred to the use of body posture and movements to enhance communication in oratory practices. The term has been adopted in a variety of disciplines such as linguistics, psychology, Human Computer Interaction (HCI), musicology, dance and figurative art. Although many disciplines use the word gesture from a perspective that makes it relevant to the issues of the accounted discipline, almost everyone agrees on the fact that gesture is human movement that carries some sort of meaning interpreted by a receiver. Thus, it is also through gestures that the audience can better decode a live performance.

A classification of different gesture in HCI is given by Cadoz (1994)\footnote{Quoted in Mulder (1996)} with
2. NIME LIVE PERFORMANCE ECOLOGY - COMPOSER AND AUDIENCE PERSPECTIVES

respect to hand-gestures. Cadoz distinguishes three different hand movements according to their function:

- **Semiotic**: Dependent on the cultural experience and used to carry meaningful information.

- **Ergotic**: associated with the ability of humans to manipulate the physical or augmented world\(^1\).

- **Epistemic**: learning experience through tactile and haptic exploration.

Zhao (2001) and McNeill (2000) classify gestures slightly differently and see three other categories:

- **Communication**: used mainly in disciplines such as linguistics. Gesture, here, is a hand or facial movement that can enrich the spoken word or be a language on its own as in the case of sign language used by hearing impaired people.

- **Control**: used in Human Computer Interaction (HCI). Gesture is considered here to be a movement that enables interaction with a computer and/or an interactive system.

- **Metaphor**: used mainly in the field of cognitive psychology and music. Gesture is carrying a metaphoric meaning that need to be interpreted subjectively by the receiver.

The above three groups are not mutually exclusive and it is possible to perceive how each of them represents a constituent element of a live electronic music performance. Based on the work of Delande (1988) and the Cadoz-Wanderley’s functional analysis (Cadoz and Wanderley 2000), Jensenius (2007) goes further by classifying metaphoric gestures that are used in a live music performances:

- **Sound Producing**: gestures that are responsible for producing sound (e.g. fingers on the fret board of a guitar)

\(^1\)‘Augmented’ is added by the author of this dissertation
• *Communicative:* could be used to communicate with a co-performer or audience. It also refers to the use of gesture to express emotional intention.

• *Sound-Facilitating* gestures that help producing (or marking) sounds. (e.g. shaking head to help maintaining the tempo).

• *Sound-Accompanying:* gestures that are made in response to music (tapping feet, clapping hands or dancers’ movements).

Each of the above gestures play a different role in a live performance. A sound-producing gesture could be defined as the primary cause of the sounding output, while the remaining three categories could be classified as *ancillary* gestures. They do not have a direct role in the sound producing mechanism but undeniably help both performer and audience to communicate and together interpret the performer’s intentions.

Figure 2.1 shows the relationship between the above two lists and illustrate how a live music performance can be seen as existing within a highly complex system in which many of the interpretations of the word ‘gesture’ coexist at the same time.

More importantly, if gestures are a key element of both oratory practices and live performances we could also say, using and agreeing with Manfred Bierwisch’s words, that the difference lies in that language, exhibiting a logical form, aims to ‘say something’ while music, being mainly gestural, aims to ‘demonstrate something’  

\[ \text{(Bierwisch 1979)} \]

### 2.1.2 Constraints

Many studies, as well as the empirical experience of senior artists, prove that creativity is not originated by unrestrained freedom. For example, Boden (2004) writes that the process of defining constraints is at the basis of the compositional process. It is only through their exploration that something new can be created. We could consider a constraint to be the cultural environment in which the composer lives, their ideas, ideologies, preferred music genres, instruments, software

---

1 Quoted in Godøy (2010, p. 69).
2. NIME LIVE PERFORMANCE ECOLOGY - COMPOSER AND AUDIENCE PERSPECTIVES

Figure 2.1: Gestures - Gestural control, meaning and communication

and electronic hardware such as physical controllers and computers but also sound sources, musical material and algorithmic and/or stylistic rules (Wishart 2010). In relation to NIME performances, Gurevich, et al., have stressed the importance of constraints for the development of style which develops via the performer’s ‘exploration, problem solving and operation with constraints’ imposed by the instrument (Gurevich et al. 2012, p. 23-24). Pearce and Wiggins (2002) compare the act of composing to a problem-solving process and classify constraints into stylistic constraints such as the choice for the music genre; internal constraints, such as the need to ensure that the overall elements of the composition are balanced and coherent to some principles; and external constraints, such as the need to ensure the playability of the composition and its readability from the audience. In the same paper, Pearce and Wiggins draw five hypotheses with regards to the ‘functional characteristics of the cognitive process which support creativity in musical composition’ (ibid. [p. 1]). Interestingly, only one of these hypotheses focusses exclusively on the composer. The remaining four take into account the audience and their expectations. The core idea is that the compositional mechanisms adopted must adhere to the body of shared experiences and mental
2.1 NIME Live Performance Ecology

schemata held by the audience and that agreement on the aesthetical judgement is only possible when these requirements are met. This is valid for both the composer when approaching a composition that aims to be original and the audience to make its judgement. With particular reference to the field of digital music, Magnusson (2010) defines constraints into three categories: *subjective constraints* as referring to the cultural background and ideology of the composer, *objective constraints* represent the physical limitations of tools in use whether software or hardware and *cultural constraints* as the complex relation between technology and ideas and how they reciprocally inform and affect each other. From the comparison of the previous two classifications, it emerges that Pearce and Wiggins’ analysis more closely reflects what is described in Perez, et al., as ‘composition driven’ approach. In Magnusson, it is the technological element to be preponderant, making it closer to the ‘technological driven’ one. We could also include the Pearce and Wiggins’ stylistic and internal constraints into the Magnusson ‘subjective’ constraints as exclusively depending on the composer’s will. What are considered ‘external’ constraints could instead be distributed between what Magnusson calls ‘objective’ and ‘cultural’ constraints: dependent on external factors not controllable by the composer but that they need to take into consideration and carefully evaluate. In light of these consideration, the proposed constraints’ classification presented in this work consists of only two macro-groups:

- **cultural constraints**: such as performer and audience’s cultural background which includes music genre choice/preferences, stylistic choices, coherent presentation and understanding of the elements included in the score, gestures.

- **technological constraints**: such as selection of the instrumentation, devices, software, hardware, playability, venues, mapping strategies and control intimacy.

In the proposed classification, the **cultural constraints** include both the stylistic, internal and external constraints from Pearce and Wiggins and the subjective constraints from Magnusson while, at the same time, acknowledging the audience
2. NIME LIVE PERFORMANCE ECOLOGY - COMPOSER AND AUDIENCE PERSPECTIVES

roles and influence in the decision making process from the composer perspective. Indeed, as soon as the composer has committed to a particular genre and/or style, they are inevitably committing to a particular audience which is assumed to know, with different degrees of expertise, the ‘rules’ of the game.

The technological constraints here are similar to the objective constraints mentioned by Magnusson and extend on that idea including others practical elements such as, for example, budget and venue available, feasibility, time constraints, workforce required and so on. In that sense, the practical constraints refer to both the physical limitations of the hardware and software in use. In addition, it refers to the practical issues the composers face before any selection of the hardware and software and the elements that will inform their choice. Magnusson’s cultural constraints are not considered here as they represent only a way of use and interaction of the presented classes and not a class on its own.

2.1.3 Affordances

In Chapter One of this thesis, we have introduced the term of ecological knowledge to describe the natural mechanism that enables the audience to associate a given sound to its instrument in music (live or not) that employs traditional instruments exclusively. A key concept in this field is that of affordances. Informed by the work of phenomenologists such as Husserl and Heiddeger, Gibson, a well-known cognitive psychologist and pioneer in the field of ecological psychology, introduces the concept of affordance to describe the properties of the relation between the environment and the agent (Gibson 1979). For example what describes the relationship between the environment and the animal (agent) is referred as affordance. As Gibson states:

An affordance is neither an objective property nor a subjective property; or it is both if you like. An affordance cuts across the dichotomy of subjective / objective and helps us to understand its inadequacy. It is equally a fact of the environment and a fact of behavior. It is both physical and psychical, yet neither. An affordance points both ways, to the environment and to the observer. (Ibid. p. 129)
2.1 NIME Live Performance Ecology

An affordance, therefore, is *not* a property of the environment but, rather, a feature of the whole situation in which the agent plays an active and fundamental part in the construction of meaning. Furthermore, Chemero suggests that events are *changes in the layout of affordances in the animal-environment system*, therefore stating that the dynamic changes of the affordances status of the system represent the key element to enable the agent to perceive ‘events’ (Chemero 2003 [p. 192]).

The ontological issues of affordance have also been introduced in the field of HCI by the seminal work of Norman (1988, 1999). Here, the environment is represented by the interface and the agent by the user. A major difference in Norman’s theory of affordance is that affordance can be dependent on the cultural background and previous experience of the user. The active knowledge of the interface is, therefore, gained through comparison with the previous experience of the world (Soegaard, 2012).

In that regard, it is also important to discuss Windsor’s interpretation of affordance in relation to acousmatic music and its listening audience which is represented by ‘*the dynamic relationship between a perceiving, acting organism and its environment*’ (Windsor 1994 [p. 11]). The environment is intended here as the real world object the sound is referring to (something referred to as sound object) (ibid.). In that sense, both performer and audience have a common ground on which to form the basis of their compositional (in the case of the composer) and interpretational (for the audience) process. Thus, it might seem that one of the key ingredients to a successful acousmatic composition is the audience’s ability to identify *sounding objects*’ equivalent in the real world. We could call this previous experience of the real world. Atkins (1999) further develops this

---

1. A good video explaining the concept of affordance in HCI is at Normann (1994).
2. This ability seems to be strictly related to individual cultural background, expertise, mood and other psycho-social factors. Godoy (2010) describes this as the gestural affordance of musical sounds. Similarly, the physical process of translating sound in real gestures has been described as the *perception-action cycle* (ibid.). In the same paper, Godoy observes that, if the link between perception and action has been widely demonstrated, less attention has been paid to the temporal aspect of this process which could eventually highlight how the mind is segmenting an apparently continuous stream of information.
3. ‘Successful’ is intended here as accessible to a wider audience. A very useful resource for this debate is Landy (2007).
interpretation suggesting that the listener will perceive the affordance of a given
sounding object by first comparing it to similar sounds that have been previously
heard in other compositions and only after comparing to the composition itself.

With regards to music performances using newly developed digital devices, we
can see how affordances are present both from an HCI and musicological point of
view. In a live performance, two parallel affordance systems are working simult-
aneously but often not ‘in tune’. There is a first subsystem, whose properties
are described by the relationship between the interface and the performer, and
a second macro-system which is instead described by the relationship between
the performer (the environment) and the audience (the agent). In the former
system, HCI plays a major role but, as previously described, compositional ap-
proaches can be affected. Norman’s definition of affordance can be valid here. For
the latter, however, it would be more proper to recall Gibson’s original definition
of an affordance which sees environment and agent as both (or neither) subject
and object, and as mutually validating the system. Indeed, according to Windsor
and Atkins both performer and audience are active parties of the same system
(i.e. the live performance).

2.1.4 Instrument or Controller?

NIME means New Interfaces for Musical Expression. The core idea here is to
develop new tools that would facilitate the user/performer to connect to (interface
with) the vast realm of sounds. The community often refers to these tools as
‘instruments’ or less often as ‘controllers’. However, it should be noted that
these two words are not synonyms and are quite often misused in the NIME
community. Indeed, most of these new interfaces for musical expression developed
by the community are arguably controllers rather than instruments. The word
‘instrument’ recalls a similarity to a traditional musical instrument (acoustic or
electronic) and thus a stricter relation to the music domain. On the contrary,
the word ‘controller’ more closely relates to the field of HCI. This distinction
clearly separates control from expressivity as noted in Dobrian and Koppelman

The roles of performer/composer and audience are inverted during the compositional phase.
The composer becomes the agent that relates to an imaginary audience.
The reasons for preferring the word ‘instrument’ over ‘controller’ is that NIME’s main research interests are (or intend to be) balanced in favor of the artistic purposes rather than the interactive properties. It should be also noted that in the NIME community the instrument is often intended to be formed by the device, the computer and their mapping (Rovan et al. 1997). However, this is only correct to a certain extent. MIDI controllers are indeed named ‘controller’ because their main function is to control parameters on an external software or hardware which is usually referred as the instrument. What is generally referred as a MIDI instrument is often instead a traditional instrument such as violin, piano, guitar which enables MIDI data transfer (i.e., we have a MIDI Violin, MIDI Guitar etc.). Ultimately, a group of bending sensors, potentiometers and/or accelerometers would interface with a computer in the same way (maybe using a different protocol) of a MIDI controller. In other cases, such as the instruments developed by Otso Lahdeojaa (2008) and the iTouch Guitar (Green 2008) where the controller part is attached to a more traditional instrument (i.e. the guitar), the preferred term is the one of ‘augmented instrument’.

A controller is instead often designed free from any sound design constraints; indeed, it has no sound on its own and it merely controls some external sound-producing device (e.g. a computer). This observation, however, is not sufficient. In fact, we could consider the neck, string and bridge of a guitar as the ‘controller’ and in combination with the guitar’s body the ‘instrument’. Similarly, a single bending sensor (the controller), when connected to a computer and generating a specific set of sounds, would make an instrument.

Cook (2004) describes the dichotomy between the performer’s sound producing interface and the sound generator mechanisms in terms of controller and generator. According to Cook, the major flaws in this paradigm are, in most cases, the lack of an haptic feedback mechanism, the lack of fidelity in the connection between the controller and the sound generator mechanism and and the lack of any sense that sound comes from the instrument (controller) itself (ibid. p. 316). These issues are then tackled with the practical development of new interfaces. Indeed, the word ‘remutualizing’ used by Cook suggests that the development of new instrument should follow the traditional workflow in which the ‘design used

1Note the simultaneous use of the words controller and instrument in the referred quote.
to be the result of mutual evolution of performer and craft, and that, with care, designers can reintroduce this symbiosis in our modern electronic instruments’ (ibid. p. 315). In that regard, the development of the spherical radiating speaker represents an interesting way to approach and solve the traditional dislocation in space between instrument and speakers (usually away from it). This device, originally implemented for the development of the Bowed Sensor Speaker Array (BoSSA) [Trueman 2000; Trueman and Cook 1999], offers the performer a better way of feeling the instrument by greatly diminishing the distance between the speakers and the performer. Thus, ultimately, it increases the perceivable control intimacy. In addition, since the device allows for the sound to originate from the performer location in space (the spherical speaker is held in proximity of the performer’s body), it creates both a stronger connection between performer and sound from an audience perspective and a reference to traditional instruments.

The development of new interfaces for musical expression have seen many other paradigms of interaction. Another issue is that, traditionally speaking, we could define an instrument as a mechanism able to only generate one specific set of sounds (so, a flute can only sound as a flute). The problem seems indeed dictated by the differentiation between the interface and the sound generator units in place. These considerations almost justify the confusion in the use of the two words. Yet, it may all depend on the uses the user made of the device. Indeed, if a series of sensors enables the control of a virtual synth or a physical modeled instrument, we could call the sensor-computer system an ‘instrument’. When the same device controls loops, starts, ends, volumes, effects and other elements of the composition, we could more easily refer to the device as a controller, rather than an instrument. It controls macro elements of a composition created in the software. From this perspective, it is then the mapping strategies in place that would enable the differentiation between an instrument and a controller. But this does not seem to be sufficient due to the fact that one of the most exciting attributes of every NIME tool is the possibility to use the instrument to produce or reproduce almost an infinite sets of sounds. This prevents the construction of an organic musical literature for the given instrument and, thus, prevents the formation of an ecological knowledge shared by both audience and performers. The mapping problem (discussed in section 2.1.5) seems, then, the
cause for this confusion beyond any possible technique implemented. It is also true that many researchers in the community now prefer to talk about digital musical instruments in order to stress the underlying difference between analogue mechanism of interaction and software (thus digital) mediated ones.

However, the problem could reside elsewhere. The Oxford Dictionary gives the following definitions for the word instrument:

- a tool or implement, especially one for precision work
- a measuring device used to gauge the level, position, speed, etc. of something
- (also musical instrument) an object or device for producing musical sounds
- a means of pursuing an aim

The word itself comes from the Latin word *instruere* which means ‘to arrange’. If so, it is clear that it is not a problem of terminology in itself but rather the specific meaning associate with the word which restricts its interpretation according to its historical heritage. If the word ‘instrument’ is intended as any tool complying with the characteristics above mentioned, the word has all the connotations to fully describe the long list of newly developed musical interfaces discussed here.

### 2.1.5 Mapping - Composer and Audience Perspectives

The possibilities offered to the composer for connecting a given set of data to a given set of musical parameters are almost limitless. This problem is known as ‘mapping’. Many studies have described this issue in terms of the number of dimensions between the input (data from the sensor/s) and the output (control parameter/s). The recurrent terminology is *one-to-one*, *one-to-many*, *many-to-one* and *many-to-many* ([Fels et al., 2002](#)). However, the link between input and output is not necessarily a direct one. Other studies have outlined how sets of data could be mapped via interpolation of given presets ([Goudeseune, 2002](#)).
Despite the many techniques available to the composer, it emerges that the problem is not limited to the sensor software interaction only. Since the gestures are used in a live scenario, the clarity of the cause/effect mechanisms are sought for the benefit of both the performer and the audience. Gestures (and we are referring here to gestures responsible for the sound producing mechanism only), represent a visual stimuli that, when coupled with the current auditory scene, can give an insight into the instrument’s controls, mapping techniques and the performer’s virtuosity. The mapping problem is two-fold: on one side its aims are to offer coherent and refined methods of interaction for the performer so that the mastering of the instrument can be developed over time. On the other hand, the cause/effect links between gesture and sound need to be clear to an audience in order to assess the functionalities of the instrument and the expertise of the performer using it. The main issue preventing the solution to these two problems is the ‘secrecy’ of the mapping that, being a programming problem only, is hidden to the audience (and performer when using the instrument for the first time).

Fels, et al., suggest that ‘the expressivity of an instrument is dependent on the transparency of the mapping for both the player and the audience’ [Fels et al. 2002 [p. 113]]. In other words, transparency is intended as a quality of the mapping techniques in use which clarifies the relationship between the input (gesture) and the output (sound). Hence, this quality is assessed on the basis of how well the audience can understand the interaction between the performer’s gesture and the audible output. To give consistency to the model, it is also assumed that both performer and spectators share the same literature (to use Fels words).

In addition, Fels, et al., (ibid.) address the relationship between the complexity of the mapping strategies versus the audience and performer understanding with the graphical plot divided in four quadrants as depicted in Figure 2.2.

Traditional instruments belong to the TT quadrant as the mechanism of interaction is known to both players and audience due to the shared ecological knowledge. Diametrically opposite is the OO quadrant for which the mapping strategies in place are unclear to both performer and audience. The OT quadrants describes an instrument in which the mapping strategies are clear to an
2.1 NIME Live Performance Ecology

![Figure 2.2: Mapping Transparency](image)

**Figure 2.2: Mapping Transparency** - Mapping transparencies versus audience and performer understanding. [Source: (Fels et al. 2002, p.116)]

Audience but that greatly limits the interaction of the performer or that does not allow a great deal of predictive results. Conversely, the TO quadrant refers to all the devices for which the mapping strategies in place are clear and which offer a high level of control intimacy to the performer but do not enable a full understanding of the underlying working mechanisms to the audience.

From an audience centered perspective, this issue is further discussed by Fayan, et al. (2009, 2010), for which the attention should be shifted on the mechanisms through which the audience gain understanding of the performative actions. The term *audience’s mental model* (ibid.) grounds the discussion in the field of HCI and cognitive psychology. The suggested model sees the judgement on the performer’s skill as derived from ‘a continuous proximity of the spectators understanding of the performers intention and the spectators understanding of the result’ (Fyans et al. 2009 [p. 172]). Domain knowledge and contextual experience are also a major factor of the level to which the spectator can form an accurate understanding of the performative interaction. However, their studies have also demonstrated that the presence of high domain knowledge and experience (i.e. ‘expert’ spectators) did not always guarantee that spectator would form an accurate model of the interaction.
2. NIME LIVE PERFORMANCE ECOLOGY - COMPOSER AND AUDIENCE PERSPECTIVES

Another consideration regards the *specificity* of gestures. Indeed, it should be noted that the mapping strategies implemented in one performance could be used in a different manner in another one. Thus, gestures and sound output are performance specific. The specificity of the mapping requires interpreting the live performance as also a pedagogic session in which the audience can be ‘taught’ and/or actively learn about the new instrument within the contest and theme of a live performance. Simple strategies such as the use of musical and gestural repetitions throughout the performance can therefore be very useful to ease the audience’s interpretative work. Thus, as the use of repetition can enhance and/or reinforce a given melody, phrase or theme, the same can be applied to a gesture and its resulting output. Moreover, the mapping and the score for the performance should be written to highlight the instrument’s affordance properties and constraints within the the broader context of the live performance and should take into consideration the mixture of both live elements (i.e. events that are controlled live by the performer) and automated systems (e.g. programs autonomously generating random numbers or automated curves of values over given intervals of time). Indeed, the ability to differentiate live and automated elements of the performance is also an important aspect to take into account in order to clarify the underlying mechanism to an audience and can, at times, also help highlighting the mapping strategies in place for the live elements.

The mapping processes should, however, strive for the establishment of a certain coherency between what we see and what we hear (cause/effect link). This coherency between multiple sensor stimuli (sound and vision) is also pointed out in the filed of cognitive psychology. The use of appropriate metaphors is a valid approach. In that regard, Hunt and Wanderley speak of *three-layer mapping* in which the intermediate layer, between the sensor input and the sound engine, is indeed represented by the mapping metaphors that link meaningful performance parameters to meaningful artistic parameters. Arfib, et al. use a similar three-layer model in which gestures and sounds are mediated by psychoacoustic elements that, in turn, are transformed by an artificial neural network whose parameters are shaped by the input (gesture) and output (sound).
Contextualizing the gestures and their mappings around the theme of the performance can also be an important strategy that opens to a new interpretation of the mapping problem in a key that does not exclusively concern the performer-instrument interaction in relation to the control intimacy perceived but widens it to the overall performance assets in order to facilitate the understanding from an audience perspective. Many elements play an important role in a live music performance and, as just outlined, these elements are not restricted to the auditory domain only. Scene, performers’ presence, lights, performers’ gestures, performance theme and stylistic choices are all elements that need to be taken into account for the development of a readable live performance.

2.1.6 Functions and Frames

Emmerson provides a clear classification of the elements describing this relationship in terms of local and field functions. The description of these two functions is reported below using Emmerson’s words (2007, p. 92):

- **Local** controls and functions seek to extend (but not to break) the perceived relation of human performer action to sounding result.

- **Field** functions create a context, a landscape or an environment within which local activity may be found.

The terminology used by Emmerson is extremely relevant for the description of a NIME performance. With local activity we can indeed describe the area in which the perceivable output is the result of the performer interaction with their instrument. Conversely, with field we can describe any sonic element that it is not affected directly by the performer. This can include prerecorded sounds as well as automated score\(^1\) events or any other algorithm for which the perceivable performer interaction is lost. The two listed functions are described by the attributes of real and imaginary. In that regard, a ‘local real’ function is the one that sees the performer’s direct manipulation of the sound producing mechanism.

\(^1\)In this dissertation the word ‘score’, rather than relating to the traditional music score, is intended as set of instructions for both the computer and the performer. An interesting discussion in that regard is offered in (Lambert 2003, p. 230 - 233).
while a ‘local imaginary’ would be one in which the performer triggers events such as sound loops prepared in advance. Similarly, real and imaginary field functions describe the semantic characteristics of the sounds used in relation to the live performer.

The displacement of these functions is given in terms of frames (Figure 2.3), a concept familiar to sociologists for which it describes where and when a given performative action is taking place and that here is interpreted in a similar manner.

![Figure 2.3: Functions and Frames - Local/Field Frames as depicted in (Emmerson 2007, p. 98).](image)

While the subsets of landscape, arena, stage and event are concerned specifically with the sound material presented in the performance, the picture clearly describes areas of separation between performer and audience. Indeed, the local and field functions, marking the limit between live interaction and pre recorded material or automated systems, are latched onto specific frames that reflect the traditional displacement of performer and audience in a venue. As stated by Emmerson, and as we will see later in this dissertation, these frames are all but fixed depending on the compositional methods employed, artistic results sought, listening and experiencing modalities as well as technological mediated issues.
2.2 Summary

This chapter began with an overview of some of the issues surrounding a NIME live performance. Particular attention has been paid to the concept of affordances, cultural and technological constraints, gestures, mapping strategies, control intimacy, functions and frames from both a composer/performer and audience perspective. This study has also helped to build an appropriate terminology that will be used in the discussion of the performances presented in the following chapters.
2. NIME LIVE PERFORMANCE ECOLOGY - COMPOSER AND AUDIENCE PERSPECTIVES
Survey of Data Gloves and their use in Live Performances

3.1 Glove Devices

The history of glove-based input devices can be traced as back as the early 1960’s with the so called ‘Communication Device’ developed by IBM’s engineers Robert Seibel and Nathaniel Rochester (Seibel and Rochester 1962). The device, developed during the Cold War to allow computer operators to typewrite in airborne high-acceleration aircrafts, is a computer keyboard fitted inside a glove (Figure 3.1). The first continuous glove controller reported in the literature was the Sayre Glove, developed in the 1977 at the University of Illinois in Chicago by Rich Sayre, Tom Defanti and Daniel Sandin while working on a project for the National Endowment for the Arts (Figure 3.2) (Rauterberg 2012; The Ohio State University 2012). This early glove used flexible tubes (that were not fiber optic) mounted on the top the hand’s fingers on which a light source at one end was beamed onto a photocell at the other. As each tube was bent by the fingers’ movements the amount of light received by the photocell varied thus varying the voltage. The glove was used to control a simple set of sliders and did not provide gesture tracking (Sturman and Zeltzer 1994).

The sensor technology used in the Sayre Glove is a predecessor of the Gary Grimes’s ‘Digital Data Entry Glove Interface Device’ developed at the Bell Telephone Laboratories (Figure 3.3 (a)) (Grimes 1983). In addition to a series of flex
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

Figure 3.1: Communication Device - Robert Seibel and Nathaniel Rochester: U.S. Patent 3,022,878. [Source: Seibel and Rochester 1962, p.1]

Figure 3.2: Sayre Glove - An archive image of the data-glove from Sturman and Zeltzer. [Source: Sturman and Zeltzer 1994, p. 32]
3.1 Glove Devices

sensors to retrieve the amount of bending of the fingers, the glove had sensors that detected the contact between different portions of the hand and a sensor to measure the inclination of the hand with respect to the gravity vector. The aim of the inventor was to offer ‘the user who is familiar with hand language, but has no typing or other keyboard skills, to input information to a machine end, at the same time, to communicate visually with another hearing impaired person’ (ibid. p.14)(Figure 3.3 (b)). Significant improvements in the development of data-gloves were made in the early eighties in the field of Virtual Reality (VR). The optical flex bending sensor patented in 1985 by Zimmerman (1985) enabled not only the retrieval of the amount of bending of the finger but also the direction of bending. The inventor employed this new technology for the development

Figure 3.3: Digital Data Entry Glove - (a) A drawing of the Digital Data Entry Glove (b) the hand-sign language enabled by the glove. [Source: (Grimes 1983 p. 9,12)]
of the VPL Dataglove and the Z-Glove (Figure 3.4) (Zimmerman et al. 1986). Both data gloves enabled also the tracking of the orientation and positioning of the hand. The VPL DataGlove achieved this by means of sensors using low frequency magnetic fields to measure six degrees of freedom while the Z-Glove used two ultrasonic transducers attached to opposite sides of the metacarpal thus making it less expensive compared to the former. Zimmerman goes even further mounting a series of piezoceramic benders underneath each finger in order to give to the user some rudimental tactile feedback. The applications for the DataGlove were many. In his paper, Zimmerman states that the glove can be used as ‘a gesture recognition device, a clinical tool for evaluating hand function, a three-dimensional hand model controller, an interface to a visual programming language, a music and sound synthesis controller, a finger spelling interpreter, and a computer-generated object manipulator’ (ibid. p.191).

Figure 3.4: VPL DataGlove - The VPL DataGlove connected to an Apple Computer for the manipulation of 3D virtual objects via hand gestures. [Source: [Zimmerman et al. 1986 p. 189]]

The Zimmerman’s optical flex sensor technology and some of the features
3.1 Glove Devices

of his DataGlove were then used in one of the earliest Virtual Reality systems developed in the 1980’s. Indeed, in 1985 NASA researcher Scott Fisher, in collaboration with Zimmeran and his VPL Research Lab, developed the Virtual Interface Environment Workstation (VIEW) where, in combination with a head mounted display, a data glove was used in order to enable the manipulation of virtual objects in the fully immersive three dimensional virtual space by means of hand gestures (Artmuseum.net 2000a,b,c; Humanoidity 2009). In 1987, British Aerospace developed an immersive car simulator named the ‘Virtual Cockpit’ in which a data glove with vibrotactile feedback was also used (Figure 3.6)(Seipel 2004).

At the end of the 1980’s, electronic components were becoming significantly cheaper and the game industry made an effort to devise new and cheaper Virtual Reality systems for the wider audience. The Power Glove, originally developed by Abrams-Gentile Entertainment and then acquired by Mattel for the Nintendo Entertainment System, was release in the 1989 (Figure 3.7)(1Up.com 2006). The Power Glove is a low resolution data glove device inspired to Zimmerman’s Data Glove. It tracked the flexion of thumb, index, middle and ring finger, the roll of
Figure 3.6: Virtual Cockpit - British Aerospace fully immersive virtual reality system developed in 1987. [Source: (Seipel 2004, p. 9)]

the hand along the x, y and z axis between +/- 45 degrees and the hand’s distance from a receiver (usually mounted on top of the TV screen) with a precision of 1/4 inch. This finger flexion was retrieved by using a particular kind of strain gauge sensors and resistive-ink flex sensors, based around the patented work of Edward E. Simmons in 1946 (Simmons 1946). This solution represented a much cheaper alternative to the optical flex sensors used by Zimmerman. Position and orientation (roll only) were tracked by means of acoustic trackers (Angelfire 1999).

Later in 1991, W-Industries (then renamed Virtuality Inc.) presented its first virtual reality game system named Virtuality. Together with a proprietary head mounted display, a data glove using optical flex sensors for the hand’s fingers and three magnetic field trackers for the hand’s orientation was also developed and named ‘Space Glove’ (Figure 3.8)(ibid.).

In the same year, Virtual Technology released another data glove named Cyberglove at a retail price of 9,800 USD (Virtual Technologies 1994)(Figure 3.9). The cost was, however, justified by the amount of technology in use. The Cyber-Glove was made of up to twenty-two sensors which included ‘two bending sensors on each finger, four abduction sensors (Kramer 1996), sensors measuring thumb
3.1 Glove Devices

Figure 3.7: Power Glove - Mattel’s data glove for the Nintendo Entertainment System released in 1989. [Source: Munchbach 2010]

Figure 3.8: Virtuality - The Virtuality system and its Space Glove developed by Virtuality Inc. in 1991. [Source: Virtuality n.d.]
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

crossover, palm arch, wrist flexion, wrist abduction and the flexion of the distal joints on the four fingers’ (Angelfire 1999). In 1993, this technology was used for the development of Glove-Talk, a system devised to interface with a speech synthesizer based on neural networks (Fels 2012; Fels and Hilton 1993a,b,c; Fels and Hinton 1993) that represented a greater improvement from previous gesture to speech synthesis experimentations such as the Talking Glove (Kramer and Leifer 1988). The CyberGlove, now owned by CyberGlove Systems, today is released in two versions, both providing wireless communication with a host computer: the CyberGlove II and CyberGlove III (Figure 3.10) (CyberGlove Systems 2012a).

Figure 3.9: CyberGlove - The CyberGlove controlling the Silicon Graphic workstation. [Source: (Virtual Technologies 1994)]

The list below tries to summarize those data-gloves developed by the industry from 1986 to present days for commercial purpose. Therefore, it excludes all the data gloves developed for research purposes only.

- (1986) VPL DataGlove - VPL Research Inc.
- (1987) DataGlove mod.4 - Greenleaf Medical Systems/VPL Research Inc.
- (1989) Powerglove - Mattel / Nintendo
- (1990) GLAD-IN-ART glove - AITek s.r.l.
- (1991) Cyberglove - Virtual Technologies

38
3.1 Glove Devices

![CyberGlove III](image)

Figure 3.10: CyberGlove III - A photo of the latest version of the CyberGlove III. Photo from cyberglovesystems.com. [Source: CyberGlove Systems (2012)]

- (1993) Pinch Glove - Fakespace Laboratories
- (1995) 5DT Dataglove - Fifth Dimension Technologies
- (1997) Humanglove - Humanware Srl
- (1998) TouchGlove - Infusion Systems
- (2007) DG5 VHand - DG Tech Engineering Solutions
- (2008) TouchGlove-R v1.7 - Infusion Systems
- (2009) Shapehand - Shapehand
- (2009) StrinGlove - Teiken Limited
- (2010) Cyberglove II - Cyberglove Systems
- (2011) 5DT Data Glove 5 Ultra - Fifth Dimension Technologies
- (2012) Cyberglove III - Cyberglove Systems

3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

It is interesting to note how the enthusiasm of the industry peaked around the early 1990’s, which is the time in which VR systems were greatly advertised and praised by the press. All these variety of data gloves devices have been employed for many purposes such as sign language interpretation, medicine and rehabilitation, virtual reality systems for game applications and pilot training, 3D modelling, motion analysis, robot control, data mining and entertainment industry. Prices of data gloves vary, of course, according to the number and quality of the sensors in use, cloth material for the glove and development costs.

3.2 Glove Devices for Music Performances

With respect to NIME music performances, several data gloves have been developed. Each data glove enables different force tracking (e.g. finger bending, pressure, touch, acceleration, rotational movements, position and attitude). Therefore, developers’ choices have been mainly influenced by mapping intentions and budget restrictions. However, while these devices have been developed for live music performance purposes, very little, if none, record and documentation demonstrating the use of them in a live scenario is available. What follows is a survey of the most relevant data gloves developed for live music performances by the NIME community.

Scanglove

The Scanglove, developed by Kessous and Arfib in 2003, explores the use of bimanual control mapping strategies for sound synthesis manipulation. It is made of two gloves: a 5DT data glove form Fifth Dimension Technologies for the right hand and custom made data glove for the left hand (Figure 3.12) (Kessous and Arfib 2003).

The flex sensor data coming from the 5DT glove is mapped to different pitches using ‘mimophony’, a MAX/Msp application based on the gestural sign language used by corsican singers to guide other musicians (Gualtieri et al. 2008). The custom made data glove is instead fitted with pressure sensors on the first and second phalanx of the index finger while the thumb is acting upon them. These continuous parameters are used to trigger the note, selected by the hand wearing
3.2 Glove Devices for Music Performances

Figure 3.11: Some data gloves - (a) Dexterous Handmaster [Worsnop 2007]; (b) Pinch Glove [Pleps 1993]; (c) DT5 Glove Ultra [Fifth Dimension Technologies 2011]; (d) P5 glove [Virtual Realities 2012b]; (e) Shapehand Glove. [Virtual Realities 2012c]
Figure 3.12: Data Glove - (a) the 5DT glove for the right hand; (b) the ‘home made’ glove for the left hand. [Source: (Kessous and Arfib 2003, p. 143)]

the 5DT glove, on a high or low octave according to which sensor the thumb acts upon. Also, the continuous data coming from the two flex sensors placed on the middle and little finger is used to control global damping and a comb filter effect respectively.

The authors used this system to control parameters in Voicer, a music software that produces vowel-like sounds. Although no public material was found, the authors also claimed to have used the system in a traditional rock band. However, the system seems to provide means of exploration for the performer which can, with time and practice, increase their level of expertise. This could be considered as a positive characteristic of the device because it enables different level of expertise by its users (beginner, medium, advanced, virtuoso). From the audience perspective, the performer’s ability in moving both fingers’ hands, can highlight the level of mastery achieved by the performer.

Pointing Fingers

Couturier and Arfib developed a data glove to create a multi-touch interactive screen so to enable the performer to control a music software via hand gestures (Figure 3.13) (Couturier and Arfib 2003).

The data glove was developed in order to retrieve when and where in the screen the user was touching a normal monitor. In order to achieve this, two switch sensors placed at the fingertip of both index finger and thumb enable the
3.2 Glove Devices for Music Performances

Figure 3.13: Data glove - The data glove developed by Couturier and Arfib. [Source: (Couturier and Arfib 2003, p. 185)]

software to know when the user touches the screen. A *bird base* from the *flock of birds* system, mounted on top of the first phalanx of the index finger and thumb, enables the retrieval of the location in which the contact was made (Ascension Technology Group 2012). The system represents a rudimentary and inexpensive way of creating a multitouch screen in a time in which only single-touch screens, at the a very high cost, were available.

**SoniMime**

SoniMime is data glove system consisting of two gloves each mounting a 3D accelerometer sensor for the tracking of the acceleration forces along the x, y and z axes. The accelerometers are connected to an ATMEL microprocessor communicating via OSC to Pd. (Figure 3.14) (Fox and Carlile 2005).

The data retrieved from the pair of accelerometers is used to control a tri-stimulus timbre model synthesizer outputting vowel-like sounds. The x and z tilt values of the right hand are mapped to the weight of the partial component of the given sound. By tilting the hand all the way to the right only the fundamental frequency will be heard as all the amplitudes for partials will be ‘0’. Conversely by tilting the hand all the way to the left the amplitude for the highest partial will be the loudest, and so on for all the middle values retrieved by the sensor. The same mapping model is applied to the left hand. This time though the x-tilt
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

Figure 3.14: SoniMime - The data glove SoniMime. [Source: (Fox and Carlile 2005, p. 242)]

Figure 3.15: SoniMime - Hand gestures mapped to the tri-stimulus timbre model. [Source: (Fox and Carlile 2005, p. 243)]
3.2 Glove Devices for Music Performances

d-controls the fundamental frequency in the range between 50Hz and 615Hz while the z-tilt value controls the overall amplitude of the resulting sound (Figure 3.15).

**VIFE**

The VIFE glove was developed for the installation named Virtual Interface to Feel Emotions, alpha v.01, an interactive audio-visual work by Rodríguez and Rodríguez (2005) (Figure 3.16).

![Figure 3.16: VIFE data glove - A screenshot of the VIFE glove. [Source: (Rodríguez and Rodríguez 2005, p. 253)]](image)

The glove is made of four pressure sensors mounted on only four fingers. Very little information has been found on this work. However, it seems that the pressure exerted by the user is mapped to ‘some’ sounds selected within a library of six ‘forms’ made of the following sounds: ‘Archanoid (Base loop), Campanas (Bass loop), Mask (Percussion loop), Titan (Voices loop), Arachnid (Melody loop), Orbital (Matrix loop)’ (ibid., p. 252).

**HandySinger**

The HandySinger is a data glove made up of seven flex sensors and two pressure sensors. The flex sensors are placed on the rear and palm side of the index, middle and thumb finger (plus one on the palm side of the ring finger). Pressure sensors are placed at the top of the thumb and middle finger (Figure 3.17).
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

(Yonezawa et al. 2005). The peculiar characteristic of this data glove is that it is in the shape of a hand-puppet. Indeed, the aim of the researcher is to enhance the performative aspects of the instruments. The bend sensors control the expression of a live singing voice. In that sense, the glove is used as MIDI effect controller. The voice is modulated in order to achieve three different expressions: normal (no expression), ‘dark’ (entirely like interior tongue vowel), ‘whisper (including more white noises)’ and ‘wet (entirely nasal voice)’ (ibid., p. 122). The pressure (touch) sensors are triggered by a member of the audience who can, therefore, feel and interact with the singing voice, too. Thus, the performer is not only the person wearing the glove but also the second person stroking the puppet. The aim here is to enhance performer’s expressions in human-to-human communication. The personification of the hand-puppet interface, a penguin, enables both performer and audience to ‘connect’ with the puppet; for the performer, the penguin will represent a kind of of performance partner through which expression and feelings are enhanced and mirrored, while for the audience an additional element that conveys expressive information about the performance. The mapping metaphors developed in order to achieve this seem relevant.

Figure 3.17: HandySinger Glove - A screenshot of the HandySinger glove. [Source: (Yonezawa et al. 2005 p. 123)]
3.2 Glove Devices for Music Performances

Beat Boxing

![Beat Boxing glove](image)

**Figure 3.18: Beat Boxing** - A screenshot of the Beat Boxing glove. [Source: (Lugo and Jack 2005, p. 246)]

Beat Boxing is another example of data glove developed in order to highlight the expressivity of hand gestures. In this case the data glove is a pair of boxing gloves fitted with accelerometers which detect hit or punches and pressure sensors near the thumb fingers for fine control. While not much detail is given about the mapping strategies adopted and no real live performance has been documented, it is important to highlight the data mapping metaphor of the data coming from the data glove which is a pair of boxing gloves whose use were, indeed, to detect a hit (Lugo and Jack 2005).

MusicGlove

The *MusicGlove* (Hayafuchi and Suzuki 2008) (Figure 3.19) consists of one three-axis accelerometer, four bending sensors, one microprocessor, a bluetooth module that enables wireless communication and a battery pack.

The MusicGlove enables the control of the devised audio/video application (implemented using both MaxMSP and Eyesweb) on three different levels:

- *Track Controls*: Play, Pause, Volume Up/Down, Next/Previous Track are triggered by bending the index finger, making a fist, rotating wrist up/down and pointing right/left respectively.
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

Figure 3.19: MusicGlove - A screenshot of the MusicGlove. [Source: (Hayafuchi and Suzuki 2008, p. 242)]

- **Sound Control**: Fast Forward, Fast Rewind, Tempo Up/Down and Scratch functions are triggered by rotating wrist right/left (rolling), up/down (pitching) and simulating scratch motion respectively.

- **Searching Tracks**: Searching and Shuffling are triggered by the hand motion and acceleration values respectively.

However, the software implemented here does not allow the user to re-map the input data. In fact, the device has been mainly devised to be used as a simple controller for a large audio/video library, and this justifies the rigidity of the mapping scheme in place. Although no document shows which HCI principles have been taken into account, this seems to be dictated by a user perspective approach in order to achieve playability.

**Mims**

Mims is an interactive system that provides visual feedback for those sounds that have been manipulated via a glove controller (Figure 3.20) [Kanda et al. 2009]. The overall system consists of a projector, a glove device, a vision sensor and a wireless microphone. The glove device consists of a three-axis accelerometer and infrared (IR) sensor that enables manipulation of small live recorded chunks.
3.2 Glove Devices for Music Performances

Figure 3.20: Mims - A screenshot of the Mims glove. [Source: Kanda et al. 2009, p. 46]

of voice. The main idea is to enable audiences to better understand performer mapping strategies at performance time and in real time via visual feedback. It clarifies the performer’s gestures and their link to the manipulated audio output. The shift of attention from a strictly user point of view to the one of the audience is the main connection to the topic investigated in this dissertation.

Vamp

Vamp (Jessop 2009) is a data glove that extends to the user’s shoulder. It is outfitted with one flex sensor over the elbow measuring the elbow’s bending and one flex sensor on the wrist. Also, an accelerometer is placed at the top of the forearm, and it is used to retrieve downbeat movements of the arm. Lastly, a pressure sensor has been attached to the index finger.

The mapping strategies in place see the data from the pressure sensor controlling the freeze effect for the voice. The accelerometers’ data is used to find peaks in movements and map these to the downbeat for the loop in use. The bending sensor on the elbow controls the dynamics (crescendo and decrescendo). Lastly, the bending sensor on wrist controls the chorus effect. While the mapping strategies in place offer a certain degree of interaction for the performer, a negative aspect of it is represented by the high level of abstraction of the mapping metaphors in use which could make the system not intuitive from an audience perspective.
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

Figure 3.21: VAMP glove - A screenshot of the VAMP data glove. [Source: Jessop 2009, p. 257]

ForTouch - Cyberglove II and TouchGlove

ForTouch is a glove based application for live performances developed by Sydney Fels, Robert Pritchard and Allison Lenters (Fels et al. 2009). The performer wears a Cyberglove II (CyberGlove Systems, 2012b) in their left hand and a TouchGlove (Infusion Systems, 2010) in their right hand. The system is mainly based around the Glove Talk II system which enables the real-time translation of hand gesture to speech (Fels 2012). With regard to the aesthetic approaches investigated by the performer and the performance, the ForTouch authors state:

Our libretto uses the current vocabulary of our single performer and the subtext of the piece is the singers discovery and teaching of the system. During the performance the singer uses the ForTouch and her own voice and often the same words occur sequentially or simultaneously. This primes the ear for comprehension and supports the connection between the system and the body (Fels et al. 2009, p. 275).

The system works around the communicative properties of gestures described
3.2 Glove Devices for Music Performances

![Figure 3.22: ForTouch - Cyberglove II (left). Marguerite Witvoet performing with ForTouch (right). [Sources: (Jessop 2009, p. 274) (left); (Virtual Realities 2012a) (right)]](image)

...in Zhao (2001) and McNeill (2000). More importantly, the work stresses the repeatedness of movements and consequent output to prime comprehension, which can be considered as a simple but important technique to facilitate the cognitive processes involved in a live performance on the audience’s end.

Other researchers have suggested the use of generic data gloves in their systems. Naef and Collicott (2006) envisage the use of a data glove for a virtual reality interface that enables several users to collaboratively spatialize sound in 3D. Similarly Wozniewski, et al. (2006), designed a framework for immersive spatial audio-visual performance. Schacher (2007) presents a series of tools, and between them a data glove, that could be used to map gesture to sound source location.

Outside the NIME realm many other data gloves for live music performances have been developed too. The P5-Glove (Virtual Realities 2012b), originally developed as a controller for game consoles, quickly became an interesting research tool for other applications such as a music controller for which a variety of music applications have been developed (Figure3.23) (Bencina 2010). David Litke’s glove application is of particular relevance for this thesis (Litke 2007). The main characteristic of the application is that it enables the deconstruction and manipulation of audio loops through gestural control. More precisely, through
Figure 3.23: P5 Glove - A screenshot of the P5 glove. [Source: Virtual Realities 2012b]

simple hand movement in the three dimensional space, the user is able to browse each region of space to which a different quality of sound has been assigned. In this way, the performer is ‘able to mentally construct a three dimensional sonic space that can be navigated’ (ibid. p. 205). The application also enables computer keyboard input to differentiate between continuous (glove) and discrete (keyboard) commands. The user can also re-map the input data to any parameter given in the software, thus allowing a much greater freedom of choice to exploit in different live performances.

The Human-Conductor is a digital interface that analyses the conductor’s gestures via CCD camera, tracking the conductor’s baton and the VPL data glove from VPL Research tracking the other hand movements (Figure 3.24(a)) (Morita et al. 1991). The data received from both hands is used to control a pre-recorded score through a MIDI synthesiser. The hand governing the baton controls the tempo of the music score while the VPL data glove is used to translate hand gestures in sequences of conducting gestures, which reflect the grammar of conducting (for example a silencing gesture as in Figure 3.24(b). The controllable MIDI parameters through hand gestures are: Volume, Vibrato, Portamento, Expressivo, Hold (damper), Timbre change.
3.2 Glove Devices for Music Performances

---

**Figure 3.24: Human-Conductor Interface** - (a) A screenshot of the VPL data glove used in the Human-Conductor interface; (b) the Pianissimo gesture in the grammar of conducting. [Source: Morita et al. 1991, p. 49,50]

The **Beatware** is a data glove made up of ten pressure sensors mounted on the fingertip of each finger and a bending sensor on the middle finger. After selecting the drum samples and drum kits, the user is able to trigger the samples by hitting their fingertips and changing the drum kit by bending the flex sensor placed on the middle finger (Figure 3.25) (Lindros and Eriksson 2001).

**eShofar** is an electronic ram’s horn played through a I-Cube data glove (Touch Glove by Infusion Systems). The pressure exerted on the horn by the performer’s fingers is mapped to a series of filters, harmonizer and granular synthesis effects (Figure 3.26) (Gluck 2005).

The list is surely not exhaustive, but it gives an idea of the great effort made by numerous researchers in the development of wearable systems for the tracking of hand gestures.
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

Figure 3.25: Beatwear - A Screenshot of the Beatwear glove. [Source: (Lindros and Eriksson 2001, p. 2)]

Figure 3.26: eShofar Instrument - Bob Gluck playing his eShofar. [Source: (Gluck 2005)]
3.3 Data gloves - Analysis of Live Performances

3.3.1 The Hands

The Hands is a pair of data gloves developed by Michel Waisvisz at STEIM Lab in Amsterdam in 1984 ([Crackle.org] n.d.). While The Hands was developed and refined over two decades, the basic system is made of a small keyboard on the player’s hands, accelerometers to measure the tilt of the hands, pressure sensors triggered by the thumbs, an ultrasound distance sensor that calculates the distance between the two hands and a microphone placed on the glove for the left hand. The Hands connects to a box secured around the back of the performer hosting a STEIM Sensorlab, a microcomputer that converts the incoming analogue data from the data gloves into MIDI messages and sends them to the host computer for further sound processing and manipulation ([STEIM] 2012).

Michel Waisvisz used The Hands in many performances. While there is little documentation available concerning the hardware, the mapping strategies adopted and the sound creation process, some video-documents available online offer the possibility to speculate on the range of possibilities offered by the data gloves. The STEIM VHS Archive has, for example, recorded a full concert of Michel Waisvisz performing with The Hands ([STEIM] 2008; [STEIMITUBE] 2008). This performance starts with a live recording of the audience’s applause that welcomes the performer. The recording lasts a few seconds and fills up a buffer. After that, the performer is able to manipulate the created audio loop by hand gestures. It is difficult to state what movement is the cause for the given processed results, but it seems to be controlled by the tilting of both hands. At 1’22” the performer hits the microphone mounted on his left glove on a mic stand to record a percussive sound that will be added to the ongoing loop. The sonic material presented in the composition increases in complexity over time. However, it is still difficult to create a clear mental picture of what movement is the cause for the heard sound. Given the density of the material, it is plausible to think that this is the result of both live and pre-recorded sounds. At 5’20” there is a rest that then alternates with short excerpts from the sonic

---

1Date and place unknown.
material previously exposed. In here, the close-up shot allows us to see that the reintroduction of the sonic material is controlled by the keyboard mounted on both hands. This represents the Coda of the performance, marking the end of the first composition.

Figure 3.27: The Hands - Performance - (a) Michel asking for a louder applause while holding his left hand up to facilitate the recording from the mic installed on it; (b) Michel tapping the glove on a mic stand; (c) Michel moving fingers on the keyboards mounted on the glove in the Coda section. [Video screenshots sources: [STEIMTUBE 2008a][b]]

The second composition opens with the performer singing a note on the mic mounted on the left hand glove. After the first note has filled up the buffer more notes are overdubbed. This creates a choir effect. Subsequently, the performers move away from the stage in an area that, in the recording available, is not well lighted, thus not allowing the viewer to see the action properly (7’53” - 8’30”). Then, he starts moving his arms around. A flanging pitch shifting effect applied to the ongoing audio loop is the result of this movement. After that, the fingers seem to trigger a prerecorded ”Ah” sound. The keyboard seems to be working as a MIDI sampler as a series of differently pitched ‘ahh’ sounds are introduced. The same happens again soon after but with another voice sound ‘woooohh’ soon followed by other noise-like sounds. However, it remains unclear how the performer switches between different samples. The arm moving gesture (hands tilting) is then repeated at 0’31” (Part 2). It also possible that the distance between the two hands is mapped to some effect parameter. These gestures are then used until the end of the performance (3’29” Part 2). The dynamic of the
composition here is similar to the first composition, as it sees an alternation of dense and quiet moments in a sort of waving motion.

The third composition starts with pre-recorded voice material presented at a different pitch and morphed in a quite complex manner. The performer’s gestural vocabulary here is very articulated. From the video it appears that the keyboards, the tilting and the distance of both hands are used simultaneously to control the voices.

![Figure 3.28: The Hands - Performance](a) Michel shouting at his mic (b) Michel recording a bell. [Video screenshots sources: (STEIMTUBE 2008 a, b)]

The fourth and last composition starts with a live recording of a bell again using the left hand glove. Tilting and distance between both hands manipulate the source material. The keyboard seems to be employed as a pitch shifter while, at the same time, triggering pre-recorded sound files. At 2’59” (part 3) a reverb effect is introduced and it appears to be triggered by the thumb of the right hand acting upon a pressure sensor. At the end of the performance, it seems that the overall volume is controlled by the tilting angle of the hands that when tilted down will fade to zero.

### 3.3.1.1 Discussion

It is difficult to establish a clear cause/effect link between gesture and sound with the available material. It is also worth noting that the speculations done here were possible only after repeated views of the performance itself. It is hard to
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

imagine an audience able to catch small movements due to the distance between the performer and the audience itself; distance that, for the viewer of the video, is shortened with the help of different camera angles and zoom effects. In the provided analysis, the probable mapping strategies adopted can be summarized as follow:

- microphone $\rightarrow$ loop overlap of live audio sources (voice, audience applause etc.)
- tilting and distance $\rightarrow$ effects parameters (flanger, pitch shifting, granularization, distortion etc.)
- keyboards $\rightarrow$ selection of pre-recorded soundfiles / sampler
- pressure sensor $\rightarrow$ effect parameters (reverb)

However, given the complexity and the density of the sonic material the performer is able to create, it is plausible to think that the mapping strategies in place are far more complicated than what is presented here. At the same time, though, no information is known about the prerecorded sounds which could also be the cause of the density perceived in some parts. More details on the working mechanism of the loop machine can be extracted from the analysis of the performance held at NIME 2003 [Waisvisz2003]. We can see that in the beginning the little finger of left hand is down which could be thought as the command ‘start recording’ that will then fill the initial buffer of a predetermined length. At 0’37’ both middle and ring fingers are down while little and index fingers are up, a gesture that enables the overdub function. At 1’39” the middle, ring and little fingers are down which can be thought as the ‘restart buffer’ command. Also, it appears that the volume of previous loops is controlled by the thumb of the right hand. However, even assuming that the description of the mapping strategies in place is correct, it is not possible to state with certainty that the same mechanism is applied in the other performances. Indeed, it is obvious that identical gestures are mapped differently in each performance. Thus, the mapping here is performance dependent. These elements enhance, on one side, the versatility of the instrument while, on the other, can lead to confusion for the audience who sees similar movement as triggers for different sound source and/or effects. The
3.3 Data gloves - Analysis of Live Performances

![Figure 3.29: The Hands - NIME’03 Performance - (a) Overdub mode (middle and ring finger down while index and little finger up; (b) Tilting hands gesture. [Video screenshots source: Waisvisz 2003)](image)

confusion increases when multiple gestures are performed at the same time (for example tilting hand and pressing keys on the keyboard), a case in which it is difficult to discern and link individual gesture to audible output. Despite of all this, the performance is engaging and the sonic material produced of value. The performer’s stage presence and ability to interact with the audience, even only with eye contact, is a strong element in the performance and favours participation. The sonic material develops with clear trajectories, enabling the listeners to visualize and contextualise sounds within the frame of the composition. The impossibility to discern whether or not the performance makes use of automated systems, delimits the frame of the performance to the local function in which a mixture of real and imaginary elements seems to be present. Ancillary gestures are minimal, and, when used, they express effort in the musical sense of the gestures rather than the effort in controlling the instrument.

3.3.2 Lady’s Glove

The very first version of the Lady’s Glove was developed in 1991 by Letitia Sonami and Paul DeMarinis for the Ars Electronica Festival in Linz. It was made of a
pair of rubber kitchen gloves hosting five Hall effect transducers and a magnet on the right hand (Figure 3.30).

![Figure 3.30: The Lady’s Glove - First version of the Lady’s Glove in 1991.](source)

Since then, many other versions of the glove were developed. The latest version was built between 1994 and 2001 with the help of Bert Bongers from STEIM Lab in Amsterdam. It consists of a Hall effect sensor on the thumb with magnets on the other four fingers, five microswitches placed on the tip of each finger, an ultrasonic emitter on the palm of the hand and receivers on the foot and right hand. Also, it includes a flex sensor on each finger of the left hand, a mercury switch on the top of the hand, an accelerometer on the left hand and two accelerometers on the right one (Figure 3.31). All sensors, sewn ‘on the top of a thin black mesh, arm-length lycra glove tailored in Paris’ (Sonami 2012b, para 4), are connected to a STEIM SensorLab which sends MIDI signals to the host application. Letitia Sonami’s performances are documented on her website, where a few excerpts from her own compositions can be found (Sonami 2012a). For the purposes of this research, a full composition found at (Sonami 2010) and titled ’Why Dreams like a Loose Engine and Autoportrait’ is analysed here.

This performance sees an alternation of four instrumental and four speech sections followed by a coda section at the end. In the first instrumental section, the performer creates a sonic landscape that prepares the entrance for the first
3.3 Data gloves - Analysis of Live Performances

Figure 3.31: The Lady’s Glove - The Lady’s Glove and the STEIM’s SensorLab.
[Video screenshot source: [Sonami 2010]]

speech section. The gestures that clearly seem to be linked to a sound output seem to lie above a pre-recorded drone loop with a high content of low frequencies. The gestures determined are responsible for the bell-like sounds. In particular, the closeness of the two wrists, detected by the ultrasound proximity sensors, triggers a fixed low pitched bell sound. Bending of the fingers, recorded by the bending sensors, trigger instead very short (few milliseconds long) voice samples (Figure 3.32(a)). The second ultrasound receiver, placed on the performer’s foot, triggers a low rumbling sound when the data glove moves into its proximity, thus forcing the performer to bend down. Towards the end of this section, the performer starts moving all fingers and wrist almost erratically, which results in click-pop noisy sounds. Subsequently, the performer stretches the arm, a gesture that seems to trigger the loop that will be used as the sonic support for the coming speech. During the speech, it is possible to see that a drum pattern is introduced at some point, but it is not clear what gesture is the cause for this output.

The sonic material supporting the first speech is abruptly interrupted by a fast and quick movement of the wrist, thus marking the end of the speech section and the beginning of the second instrumental section. This section is, however, very
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

Figure 3.32: Letitia Sonami and the Lady’s Glove - (a) Fingers moving close to her mouth triggers grainy vocal sounds; (b) waving movement of both hands triggering low sound loops; (c) outward stretching of both arm in the Coda section. [Video screenshots source: (Sonami 2010)]

short and the previous loop, used in the first speech section, is soon re-introduced, thus moving on the second speech section. An abrupt movement again marks the end of the speech section and the beginning of the third instrumental section. Here, pushing both hands forward triggers a low pitched sound while bending the fingers produces again several grainy vocal sounds. Then, the performance moves onto the third speech section. This section stops then for few seconds where, on top of the usual loop, a few glitchy sounds are introduced presumably by the movement of the fingers. Then, the closing sentence is spoken followed by the coda section. Some of the gestures previously exposed are re-introduced here. A waving movement of both hands triggers a bell sound (Figure3.32(b)), and the bending of fingers are responsible for the grainy vocal samples. A few more gestures are introduced. The hands reaching the floor triggers an even lower sound loop with some rhythmic patterns while bending the wrist backward seems to stop all sounds. Lastly, a fast outward stretching movement of both arms seems to fade out all loops while leaving only a drone sound (Figure3.32(c)). This gesture is repeated few times and eventually is the gesture that marks the end of the performance by fading all sounds out, leaving the space for few scattered sounds triggered by the bending of the fingers.
3.3 Data gloves - Analysis of Live Performances

3.3.2.1 Discussion

The above discussion on the mapping strategies in place is only speculative as no document is available in that regards. However, beside the mapping strategies in place, three interesting elements can be observed in this performance. The first is that the performer repeats the same gestures over time, offering the audience a chance to repeatedly re-evaluate and reinforce the mental model with regards to the device properties. For example, the performer repeats the movement in which moving both wrists close to each other make a bell sound. While this technique may not help to understand precisely what data from the sensors is involved in the process (mapping), it helps however to create a sort of gestural vocabulary that the audience can study and learn.

The second observable element is the theatrical connotations of the movements performed. A clear example of this would be the way fingers are moved to produce short vocal samples. Indeed, the performer performs the movement with her hand close to her mouth as to, in a sort of metaphorical gesture, emphasize the creation of sounds from her mouth. This also helps to identify beginning and end points of a gesture, and, thus, it represents another element that facilitates the construction of the gestural vocabulary.

A third element is the way the performer uses the stage space. From her movements, it seems as if she is ‘communicating’ with her surroundings. This element brings to life the space in which the performer acts, enhancing the performatative act beyond the sole use of the data glove.

3.3.3 Collide

Joseph Butch Rovan uses a custom-built glove controller in his performances Continuities (1997) and COLLIDE (2002) [Rovan 2010]. The glove worn on the right hand uses force-sensitive-resistors (FSRs) on the fingertips, bend sensors, and an accelerometer. The left hand wears a glove with reflective material used to make an infrared sensor responsive to an infrared light. The analogue to digital conversion of the data from the sensors is done by two custom-built analog-to-digital interfaces. The software is developed in Max/MSP. On the working mechanism of the two gloves, Rovan states that ‘the left hand glove is used to
control macro-behavior of the glove-controller mappings; together the two gloves create a multi-modal controller where large-scale structure is controlled by the left hand, and fine detail is controlled by the right' (ibid., para 3). The author also states that in his two composition, he uses several ‘scenes’. He can control their duration but not the succession in time which is instead predetermined.

*Collide* (Rovan 2009), is a live performance made out of three scenes. The way the sonic material is presented in each scene has a strong connection to the ABA sonata form. The material presented in scene one is repeated with small variation in scene three, while scene two could be thought of as the ‘modulating’ section (the composition is, however, not tonal). According to Rovan, the typical gestures employed in his composition are ‘fingertip pressure, rotation of the hand/arm, and various full-arm gestures ranging in speed and energy’ (ibid., para 4). Collide starts with slow movement and quiet sounds. This introduction soon moves into a more agitated section (section A) matched by agitated movements of the performer. Similar movements are introduced in the second scene. However, the sonic material and movements here are more relaxed and slow. The third scene represents a similar gestural vocabulary to scene one. A Coda section with slow and low sounds concludes the performance.
3.3 Data gloves - Analysis of Live Performances

3.3.3.1 Discussion

What emerges from the observation of the performer’s movements is how both hand position and orientation in the space occupy a relevant portion of the overall gestural dictionary presented in the performance. The performer’s control over the audio material here is also extended to the live visual elements that in Collide are an homage to surrealist film makers Léger, Buñuel and others. However, the gestural vocabulary in place is cryptic. The same or similar gestures are presented in each section where different sonic material is presented. Thus, from an audience perspective, it is difficult to discern what movement is the cause for the audible sound or effect when the same movement is presented in two different sonic environments in such short space of time. The complexity of the sound material in the composition does not facilitate this process, as it is difficult to know what effect parameter is controlled live. In addition, the balance between the performer’s ancillary movements and sound producing ones is in greatly in favor of the first. In other words, it seems that the performer exaggerates the movements that are not strictly related to the control of sound as the audible changes in the sonic material do not match the performative effort enacted by the performer.

3.3.4 Imogen Heap’s data glove

Imogen Heap’s the Gloves is an ongoing collaborative project that has involved several engineers and artists (Music Gloves Ltd. 2012). The first data glove prototype was developed with Thomas Mitchell, and it served as a controller for an ad hoc application named SoniGrasp (Mitchell and Heap 2011). By wearing a 5DT 14 Ultra dataglove from Fifth Dimension fitted with a lavaliere microphone, the user is able to control a loop station connected to the mic. Eight different hand postures are identified via a neural network application that reads the data coming from the 14 flex sensors mounted on the 5DT 14 Ultra and mapped to the control parameters of the loop station (Figure 3.34). The gestural vocabulary enables the control of the application on two levels: an audio control mode which reads the continuous data from the sensors and a switch mode control that enables the selection of the function needed and then controls it through the continuous
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

Figure 3.34: The 5DT glove - The eight hand postures used to control the SoundGrasp Application. [Source: (Mitchell and Heap 2011, p.467)]

data in audio control mode (ibid.). For example, the user is able to switch to ‘Record’ mode by performing two gestures in sequence. After the selection has been made the performer is able to control the Record mode by performing one of the given gestures which are now interpreted as a function of the currently selected mode. Since one out of the eight gestures is used to let the application know that a new switch mode is about to be selected, the application allows for seven control parameters (record, play, filter, reverb, delay, lock).

Figure 3.35: Imogen Heap’s Gloves - The 5DT 14 Ultra data glove fitted with an Inertial Measurement Unit pack and LED. [Source: (Mitchell et al. 2012, p. 22)]
3.3 Data gloves - Analysis of Live Performances

The newer version of the Gloves is made of a data glove for each hand. An inertial measurement unit (IMU), fitted on the top of each hand, enables the retrieval of the hand’s orientation. A colour LED on each hand is also mounted to give visual feedback to the performer (and to the audience, too) on the current selected mode in the host application (Mitchell et al., 2012). With the addition of the IMU pack, the developed mapping strategies are more complex. The continuous controls now include the data from the fourteen optical flex sensors, the hands orientation data in Euler angle format and the relative angular rotation of the hands, retrieved by numerical integration and constantly updated by the gravitational value retrieved by the accelerometers. The incoming data is also used as discrete values to calculate movements’ peaks, segmenting the orientation values in regions and to identify different postures via neural networks. The combination of continuous and discrete data values enables for a great variety of mapping strategies. These can be observed in Imogen Heap’s performance for TEDxBristol (TEDxBRISTOL, 2011).

Here gestures and sound output are devised in order to enhance the clarity of the sound producing or sound control mechanism. The performance starts with the performer singing on the microphone fitted on the right hand glove. The opening and closing of the hand marks the start and end points of the loop (Figure 3.37). The performance then develops by adding to and processing this initial loop. Here several gestural mapping techniques are employed. The performer is able to add effects to the ongoing loop such as filters, reverb, distortion and panning (Figure 3.36). In order to achieve this, a gestural vocabulary is developed. In addition, it seems that the process of recalling the different modes necessitates the use of both hands and that a combination of flex sensor and IMU data is employed. The filter effect, for example, is triggered by closing the right hand near the abdomen and reaching the left hand in a sort of prayer gesture while the center frequency of the filter is controlled by the pointing direction of the hand. The gesture controlling the distortion effect is instead triggered by bending both right and middle fingers of both hands with the hands raised in the air. Panning is controlled by moving the left arm left to right (or vice versa). Reverb is instead triggered and controlled by a sort of ‘hugging’ movement with hands up. Interesting is the visual feedback offered by the LEDs to the performer,
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

Figure 3.36: Imogen Heap’s Gesture - Gestures used for the control of effect parameters such as filters (a), distortion (b), reverb (c) and panning (d). [Video screenshots source: [TEDxBRISTOL 2011]]

Figure 3.37: Imogen Heap’s Gestures - The hand closing (a) and opening (b) for voice loop recording. [Video screenshots source: [TEDxBRISTOL 2011]]
which sees them coloured as blue when gestures are linked to effect parameters. For all effects, the pair of gloves display a blue LED light, presumably indicating the performer that she is now able to control the effects.

In the other sections of the performance, the performer is able to control a synthesizer and a sampler with drum samples. In Figure 3.38 (a) and (b), the height of both hands is mapped to notes of a synthesizer which will play a note in tune with the live voice of the performer. In Figure 3.38 (c) the abrupt change in direction of both hands (peak) combined with the orientation data triggers a crash cymbal sample.

Figure 3.38: Imogen Heap’s Gestures - Imogen playing a synthesizer with both hand gestures (a) (b); Imogen playing a crash cymbal samples (c). [Video screenshots source: TEDxBRISTOL2011]

In Figure 3.39 (a), the performer is able to interact with a physical instrument...
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

and record the output live. This is done by switching the Gloves to Lock mode, so that no gesture is read and by activating the microphone mounted in one of the gloves. In Figure 3.39 (b), the performer again is playing a physical instrument. However, this time the gesture from the hands is read and processes the outgoing loop previously recorded by changing its speed. This is also confirmed by the blue LED light (effect mode) visible on both gloves. The final gesture concluding the performance is a silencing movement that sees the left hand with the index finger stretched moving towards the performer’s mouth (Figure 3.40).

![Imogen Heap’s Gestures](a) LED on Gloves are yellow indicating that recording of live instruments is on; (b) the LED are blue indicating the right glove is controlling the speed of the ongoing loop and microphone is active. [Video screenshots source: TEDxBristol 2011]

3.3.4.1 Discussion

The mapping strategies in place in this performance are complex. Despite this, the cause/effect link between gesture and sound is clear in most occasions. The author of this dissertation saw the video first and then read the available documentation. It was noticed that most of the mapping strategies adopted by the composer where clear to the viewer from the first view of the video and that the written documentation, while adding important details on the mapping strategies, served as a way of confirming the correctness of the mental model created. The reason behind the clarity of the cause/effect mechanism is determined by
3.3 Data gloves - Analysis of Live Performances

![Figure 3.40: Imogen Heap’s Gesture](image)

The silencing gesture concluding the performance. [Video screenshot source: TEDxBRISTOL 2011]

the use of metaphors. Closing the hands near the abdomen in order to activate a band pass filter, using pointing directions to pan the sound as well as using a silencing gesture to stop the performance are clear metaphors of the audible results. In addition, these gestures are enhanced by the fact that they belong to the gestural vocabulary we commonly use in support of verbal communication. Indeed, the weaker, so to speak, gestures are the ones that enable the control of the audio effects as finding the right gestural metaphor for distortion is more abstract and thus more arbitrary if compared, for example, to hitting the hand in space to hit a crash cymbal. It is important to notice that the musical style of the performance is also an element that facilitates the understanding of the correct cause/effect links. The use of the live voice and some traditional instruments helps understanding what is controlled by the gloves and what is instead caused by the use of the voice or by interaction with the traditional instruments. A similar discussion applies to the way the performer interacts with the software synth to play notes and the sampler to play drum samples. While the gestures to interact with the former do not have any specific reference to real world elements (other than the fact that the height of the arm is linked to the high or low pitch of the synth), the latter is controlled by gestures with strong connections to the way real drummers hit their drums.
3. SURVEY OF DATA GLOVES AND THEIR USE IN LIVE PERFORMANCES

3.4 Summary

This chapter presented first an historical review on the several data gloves developed by the industry from the early 1960’s to present days. This was followed by a review of data gloves developed by the NIME community for artistic purposes. Lastly, it presented a review of some performances in which a data glove has been used. Given the minimal information found regarding the performances, the analysis conducted was almost exclusively based on impressions from the perspective of an audience. Despite that, it served to discuss a few examples of interesting mapping metaphors and use of the glove in different scenarios.
4

Development of Pointing-at Data Glove

4.1 History and Partners involved

The research project, which forms the basis for the later development of the Pointing-at data glove, was known under the name of Celeritas Tyndall-NAP60 2012. The initial aim of the Celeritas project was to develop a wearable motion-tracking system for dance performances (Torre et al., 2007) (O’Flynn et al., 2007). The goal was to represent an alternative to conventional camera tracking systems such as Eyesweb InfoMus-Lab 2012 or PhaseSpace PhaseSpace 2012 and a competitor of the now fully developed Moven system XSENS 2012. The partners involved were the Tyndall National Research Institute of Cork who was responsible for the hardware development and the Interaction Design Centre at the University of Limerick who was responsible for the software development. The design of the performances using the wearable sensor unit was given to media artist Todd Winkler and dancer Cindy Cummings.

The wearable sensor unit was made of up to eight sensor nodes (also known as ‘motes’) displaced around the body of the dancer as depicted in Figure 4.1 and

\footnote{Brendan O’Flynn: Principal Investigator; Philip Angove, Javier Torres, Cian O’Mathuna, John Barton, Andrew Lynch, Mohammed Tarik Koubaa: Mote programmers and developers;}

\footnote{Mikael Fernstrom: Principal Investigator; Stephen Shirley and Marc McLoughlin: Driver programmers; Giuseppe Torre: Max/MSP Interface programmer;
4. DEVELOPMENT OF POINTING-AT DATA GLOVE

communicating wirelessly with a basestation connected to the host computer. The given displacement was designed to retrieve the human figure posture and use the data to control the audio-visual elements of the performance through the application chosen by the artist (O’Flynn et al. 2007; Torre et al. 2007). This early stage of the research did not see the development of a dance performance. In a private correspondence (August 2008), Todd Winkler let the team know about his inability to further contribute to the project due to his involvement with several other projects at that time. The author of this dissertation, initially involved in the project as Max/MSP interface programmer, decided to conduct a study about the development of live performances in which the use of the sensor technology developed could be utilised.

![Figure 4.1: Celeritas Body Suite](image)

Figure 4.1: Celeritas Body Suite - Sensor displacement of the sensor nodes (motes) on human body.

In this early stage of the research project, many technological issues still needed to be addressed before the sensor unit could be successfully implemented in a live performance scenario. The original system was quite complicated to handle as it required the simultaneous use of six motes. A study on the interpretation and numerical evaluation of the incoming data needed to be carried out. In addition, it was felt that the motivations for the devised mapping strategies needed to be coherently justified in order to enhance the cause-effect mechanism on the audience perspective and to enhance ‘meaning’ from compositional point
4.2 Pointing-at Glove Design

of view. Using six motes did not seem a convenient starting point for the research. Section 5.2 of this dissertation describes the issues encountered in the first attempt to use multiple motes in order to manipulate audio-visual elements with the available data. In light of these considerations, the idea of developing a data glove slowly emerged. By narrowing the focus to the hand only (as opposed to the entire human figure), the use of one single mote was required and a more detailed study on the sensor technology in use could be conducted.

4.2 Pointing-at Glove Design

The first prototype of the data glove (2008) was made out of a golf glove. A velcro strap on the hand’s dorsum allowed for the holding of the sensor pack. An ordinary piece of cloth was sewn along the index finger in order to host the bending sensor. Although used in many demos and performances, this version of the glove lacked in robustness, wearability and look (Figure 4.2).

![Figure 4.2: The earliest version of the data glove - The golf club and sensor pack.]

In late 2008, the author of this dissertation decided to contact Niall DeLoughry and Dermot McInerney from the School of Product Design at the University of Limerick to ask whether it was possible to receive help from someone to design a better glove for the mote. It was then suggested to offer the glove design project to students from the undergraduate course in Product Design & Technology as
4. DEVELOPMENT OF POINTING-AT DATA GLOVE

part of their Cooperative Education Program. In January 2009, two students, Cillian O’Sullivan and John McCall, decided to take up the project.

The work was conducted in order to emulate a real industry scenario in which the author of this dissertation was the client and the students the product designers who were in charge for designing and delivering the data glove on time according to the client’s specifications.

Three main characteristics were required by the new design of the glove: robustness, wearability and an appealing look. Also, it was important to find out a way of keeping the sensor firmly in one position with a protective cover while keeping the weight to a minimum. The case developed for the sensor complied with these specifications. The separation between the sensors (hosted in the protective case) and the battery (fitted on to a pocket next to the protective case) also allowed for easy recharging operations. Lastly, the glove structure was reduced to a minimum in order to improve the hand’s movements and reduce sweating. Figure 4.3 and 4.4 show a few sketches and prototypes. Figure 4.5 and 4.6 show the final results. A full list of drawings and photos of the work in progress have been included in the attached CD.

4.3 Hardware Specifications

The Pointing-at glove is designed around the Tyndall’s 25mm Wireless Inertial Measurement Unit (WIMU), which is an array of sensors connected to a high resolution Analog to Digital converter (ADC). The array has three single-axis gyroscopes, ADXRS150, two dual axis accelerometers, ADXL202, both from Analog Devices, two dual axis magnetometers, HMC1052L, from Honeywell and one 3-inch Flexpoint Bend Sensor from Flexpoint Sensor Systems, Inc. The ADC is a 12-bit AD7490 from Analog Device. The ADC converter has

---

1The Cooperative Education Program is part of any course at the University of Limerick where students have the opportunity to work for a company or an institution for a period of up to six months.

2Fitted only on the latest version of the glove.
4.3 Hardware Specifications

Figure 4.3: Pointing-at glove design - Sketches

Figure 4.4: Pointing-at glove design - Prototypes
4. DEVELOPMENT OF POINTING-AT DATA GLOVE

**Figure 4.5: Pointing-at glove** - The glove suit (left) and the protective case, bending sensor and battery (right).

**Figure 4.6: Pointing-at glove** - Final design and look of the Pointing-at glove.
4.4 UART settings and MaxMSP Data Input

A custom driver, developed at the Interaction Design Centre of the University of Limerick, enables the communication between the basestation and a host computer through the specified serial port (USB). The initial version of the driver

1 Designed and manufactured by the Tyndall National Institute of Cork (NAP60).
2 Please note that this configuration will create a copy for one axis only for both the two dual-axis accelerometers and magnetometers (two dual axis sensors read potentially 4 axes). This copy is discarded at code level. There are now three axis accelerometers and magnetometers available making this coding step unnecessary.
3 A detailed description of the driver’s code, written as unpublished internal report by Shirley (2006), is in Appendix C; full code is in Appendix D.

Table 4.1: Sensors’ Resolution

<table>
<thead>
<tr>
<th></th>
<th>Resolution</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gyroscope</td>
<td>4.5 mV / °/s</td>
<td>0.27 °/s</td>
<td>406 °/s</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>600 mV / g</td>
<td>0.002 g</td>
<td>2 g</td>
</tr>
<tr>
<td>Magnetometer</td>
<td>385 mV / gauss</td>
<td>0.317 gauss</td>
<td>6 gauss</td>
</tr>
</tbody>
</table>

The cluster of sensors constitutes a typical set adopted to retrieve the three-dimensional orientation of an object with respect to a fixed frame. To allow retrieval of the object’s attitude, it is important that all sensors are placed in such way they can read the exerted forces (acceleration, angular speed and polar magnitude) on each of the three orthogonal axes. In addition to this, a series of slots were built onto the motherboard to receive the daughterboard at exactly 90° facilitating the right displacement of the sensors (see Figure 4.7).

The WIMU is powered by a 3.7V lithium-ion battery pack and it communicates wirelessly with a basestation using an nRF2401 single-chip 2.4GHz transceiver from Nordic (Nordic Semiconductor 2012) (see Figure 4.8).
4. DEVELOPMENT OF POINTING-AT DATA GLOVE

Figure 4.7: 25mm Wireless Inertial Measurement Unit (WIMU) - Sensor displacement and motherboard-daughterboard configuration. Importantly, the sensors are placed to read each given axis.

Figure 4.8: 25mm WIMU - Battery pack (left) and basestation (right).
was coded by Steve Shirley and it enabled the reading of the accelerometers’ and gyroscopes’ data only. Subsequently the code was modified by Marc McLoughlin, also a researcher at the Interaction Design Centre of the University of Limerick, in order to also read magnetometers and flex sensor data. The data arrives at the basestation as a packet of bytes. The packet length is 23 bytes. The 23 bytes are made up of 21 bytes of data and 2 synchronisation / delimiting bytes. The delimiting bytes are Carriage Return (0x0A) and Line Feed (0x0D). The 21 bytes of data are divided as shown in Figure 4.9.

![Figure 4.9: Packet Length](image)

The Universal Asynchronous Receiver Transmitter (UART) speed is set to 38400 kbps, 8 bits data, 1 start-bit, 1 stop-bit and no parity. The overall transmission speed is circa 208 arrays/sec. The data received at the serial port is then imported into Max/Msp ([Cycling 74][2010]). A custom Max/Msp object named mote reads from the serial port at a constant rate of 11ms displaying the raw ADC data from the sensors into the Max 5 environment. Figure 4.10 shows the Max patch enabling the reading from the sensor node. The mote object receives a ‘bang’ on its inlet from a metronome working at 11ms. The leftmost outlet from the ‘mote’ object outputs the data received by the serial port as a list. The

---

1 See Appendix D or refer to attached CD for full mote object code.
4. DEVELOPMENT OF POINTING-AT DATA GLOVE

‘unpack’ object enables then the decomposition of the list in single integer values. Lastly, the middle outlet of the ‘mote object outputs a ‘bang’ every time it successfully receives a data packet from the serial port while the rightmost outlet output the time interval occurred between two consecutive packet of data.

![Figure 4.10: Max 5 mote patch - Reading and displaying of the sensor data.](image)

4.5 Numerical Evaluation

The digital data coming from the sensors, with values between 0 and 4096, have little meaning in their raw form. A set of mathematical operations is therefore needed. This represents a propaedeutic step to the interpretation and mapping development discussed in Chapter Five. The following section shows these operations.

4.5.1 Sensors’ Offset and Resolution

The Analogue to Digital converter has a resolution of 12 bits. Nominally the offset should be set at 2048. In the case of the gyroscope, this should be interpreted as an angular velocity of 0°/s. This is not the case for several reasons which include manufacture mistakes, battery charge and outside temperature. Therefore, it becomes useful to calculate the sensor’s offset mathematically with a proper set of equations.
To calculate the gyroscopes’ offset will be sufficient leaving the 25mm in a steady position for few seconds passing the data through a first order Infinite Impulse Response (IIR) lowpass filter. The filter has been implemented in MaxMSP using the \textit{alphafilter} object from the FTM Library \cite{IRCAM2012} (Figure 4.11).

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{fig4.11.png}
\caption{\textbf{Max 5 patch} - Calculating offset for the gyroscopes’ data.}
\end{figure}

To calculate the gyroscopes’ resolution the 25mm WIMU was placed on a turning table for each of its three sides, depending on which sensor we wanted to calculate the resolution. The constant speed of the turning table allowed the following equation:

\[
\text{ResGyro} = \frac{\text{BatteryVolts}}{\text{ADC Resolution}} \cdot \frac{\text{GyroReading}}{\text{Turning Table Speed (33rpm)}} \tag{4.1}
\]

which can be solved numerically as:

\[
\text{ResGyro} = \frac{3.7}{4096} \cdot \frac{\text{GyroReading}}{198} \tag{4.2}
\]

Calculating Offset and Resolution for Accelerometers and Magnetometers is instead much easier. Indeed, it is sufficient to fully rotate each axis by adding
or subtracting the maximum and minimum values retrieved and divide them by two (Equation 4.3 and 4.4).

\[\text{Offset} = \frac{\text{Max} + \text{Min}}{2}\] (4.3)

\[\text{Resolution} = \frac{\text{Max} - \text{Min}}{2}\] (4.4)

A Max patch has been developed for this task (Figure 4.12).

**Figure 4.12: Max 5 patch** - The patch used to calculate accelerometer’s and magnetometer’s offset and resolution. The process is repeated for each individual axis.

### 4.5.2 Bending Sensor

The bending sensor has been devised to work as a simple on/off switch. This has been made possible by programming the ADC input to only output three values (266, 522 and 778). Since the intermediate value (522) was too sensitive to minimal
and intermediate movements (from stretching to closing the index finger), a simple
Max patch has been implemented in order to accept only minimum and maximum
values (266 and 778), which are then scaled to 0 and 1 respectively in order to
act as toggle switch (Figure 4.13).

4.5.3 Attitude Estimation - TRIAD ALGORITHM

The presented cluster of sensors is a typical set to retrieve the attitude of an object
with respect to a fixed-reference system (usually the Earth). There are three
orthogonal axes for the object \((x_O, y_O, z_O)\) and three the orthogonal axes in the
reference system (the Earth, \(x_R, y_R, z_R\)). Therefore, the array enables 6 Degrees of
Freedom (DOF). Both Object and Reference system axes are orientated according
to the convention depicted in Figure 4.14.

In the initial stage of the research, Tyndall’s researchers have suggested using
an algorithm that enabled the calculation of the attitude using a combination
of values retrieved by the gyroscopes, accelerometers and magnetometer in the
form of rotation matrices. The algorithm used the rotation matrix generated by
the gyroscopes as the main attitude value and updated it against the inverted
rotation matrix retrieved by combining accelerometers and magnetometers data
(Torre et al. 2008). Combining these two sources of estimation was done in
order to compensate for the lack of long-term precision of the gyroscopes (values
were integrated over time) and the lack of short-term precision of accelerometers.
Figure 4.14: Orthogonal Axis System - The figure shows the convention adopted by the author for the numerical evaluation and discussion of object and reference system axes.

Figure 4.15: Pitch, Yaw and Roll movements. - Airplane vs hand orientation.
4.5 Numerical Evaluation

due to instantaneous acceleration forces caused by the hand movements. While
the algorithm presented is in theory a good method to estimate the attitude,
numerous problems were encountered in its practical implementation. Despite all
efforts, these issue were not resolved and the implementation of this algorithm
was eventually abandoned.

After further research, it was decided to adopt a different algorithm. The
algorithm chosen is known as TRIAD, and it is widely used by aeronautical en-
gineers to calculate the attitude of aero-vehicles and satellites [Bar-Itzhack and
Harman 1996; Chua 2010; Weisstein 2012]. The TRIAD requires two known and
two measured, non-perpendicular vectors. It is an elegant and efficient way of
retrieving attitude. In our case, we use the magnetometers’ and accelerometers’
readings as the measured vectors and the gravity and magnetic pole vectors as
the reference vectors. The magnetic pole vectors are not perpendicular to the
gravitational ones except when we are exactly at the two Magnetic Earth poles,²
therefore the TRIAD algorithm requirements are satisfied. Moreover, since the
order of the mathematical operations is important to get good results, the al-
gorithm assumes that one system vector is more precise than the other. In our
case, we will assume that the accelerometers readings are more reliable since
they operate mainly on two rotational axes: pitch (rotation around the X axis)
and roll (rotation around the Y axis). The magnetometer will compensate for
the Yaw (rotation around the Z axis) (please refer to Figure 4.15). The algo-
rithm is computationally efficient and does not require mathematical integration
of accelerometer values.

The TRIAD algorithm uses the following set of equations.³ Suppose we have
the following set of body vectors retrieved from the accelerometers and magne-
tometers readings:

1 A special thanks to the work of Martin Baker http://www.euclideanspace.com/ and Chris
Hall’s lectures’ notes without whom my comprehension of this very difficult math would have
been unlikely.

² Geographical North and South poles do not coincide with the true magnetic earth field.
Declination and inclination degrees are measures that describe this discrepancy which changes
with latitude and longitude.

³ A numerical example is given in Appendix B; the code for the Max/MSP external developed
is in Appendix E
4. DEVELOPMENT OF POINTING-AT DATA GLOVE

- Accelerometers reading vector $\hat{A}_{ba}\{x_{ba}, y_{ba}, z_{ba}\}$

- Magnetometers reading vector $\hat{M}_{bm}\{x_{bm}, y_{bm}, z_{bm}\}$

and the gravity and magnetic pole vectors of the known reference system:

- Gravity vector $\hat{A}_{ia}\{x_{ia}, y_{ia}, z_{ia}\}$

- Magnetic pole vector $\hat{M}_{im}\{x_{im}, y_{im}, z_{im}\}$

we need to define a new body frame $R^b\{\hat{t}_{1b}, \hat{t}_{2b}, \hat{t}_{3b}\}$ as:

$$\hat{t}_{1b} = \hat{A}_{ba}$$

(4.5)

$$\hat{t}_{2b} = \frac{\hat{A}_{ba} \times \hat{M}_{bm}}{|\hat{A}_{ba} \times \hat{M}_{bm}|}$$

(4.6)

$$\hat{t}_{3b} = \hat{t}_{1b} \times \hat{t}_{2b}$$

(4.7)

and a new reference frame $R^i\{\hat{t}_{1i}, \hat{t}_{2i}, \hat{t}_{3i}\}$ as:

$$\hat{t}_{1i} = \hat{A}_{ia}$$

(4.8)

$$\hat{t}_{2i} = \frac{\hat{A}_{ia} \times \hat{M}_{im}}{|\hat{A}_{ia} \times \hat{M}_{im}|}$$

(4.9)

$$\hat{t}_{3i} = \hat{t}_{1i} \times \hat{t}_{2i}$$

(4.10)

Finally, we can calculate the object’s attitude $R^a$ as:

$$R^a = R^b(R^i)^T = [\hat{t}_{1b}, \hat{t}_{2b}, \hat{t}_{3b}][\hat{t}_{1i}, \hat{t}_{2i}, \hat{t}_{3i}]^T$$

(4.11)
4.5 Numerical Evaluation

where the symbol $^T$ denotes transposition.

The resulting matrix can be then written in quaternion form and then normalised. These equations are included in the `ahrs_triad` and `ahrs_triad_mote` custom Max externals as part of the AHRS Library developed as part of this research project.

4.5.3.1 Accuracy Considerations

The main advantage of the TRIAD algorithm is that it does not require any mathematical integration to calculate the attitude. When algorithms employ integration, estimation errors that propagate and increase over time are introduced. These need to take into account in order to retrieve usable results (Flenniken IV et al. 2005; Saini et al. 2010). This involves at least filtering, error estimation calculation and updating real attitude values from Global Positioning System (GPS) at a constant interval of time. The attitude estimation errors that the TRIAD algorithm introduces are instead mainly due to sensor reading errors caused by flicker noise, calibration error, thermo-mechanical white noise and/or external agent such as temperature (Woodman 2007). However, since it has been decided to limit the data evaluation to attitude rather than positioning, the results obtained are satisfactory for the purposes of this dissertation.

One downside of the implemented algorithm is represented by the nearby electromagnetic fields that are created by devices around the performer, such as a computer. This could interfere with the reading of the earth magnetic field and cause the magnetometers to retrieve an erroneous sample and thus obtaining a wrong glove attitude. However, it has been been found that these errors are rare and can be avoided by correctly calibrating the sensor when starting up the software interface.

---

1Please refer to Appendix F for the complete list of externals developed.
2Kalman filter is one of the most common.
3The algorithm implemented in this paper should not be used alone in any critical inertial navigation system/unit.
4. DEVELOPMENT OF POINTING-AT DATA GLOVE

4.6 Max Sensor Interface

The Sensor Interface has been designed in Max 5. The interface is divided in five main sections (Figure 4.16):

- **Sensor Input** displays the incoming sensor data.
- **Calibration** calculates offset and resolution for the accelerometers, magnetometers and gyroscopes (only offset for gyros).
- **On Tour** allows the user to calibrate the system according to user geolocation.
- **Orientation Visualization** displays the orientation/attitude of the object/-glove (hand).
- **Calculated Values** displays the retrieved data after the numerical evaluation.

Running sequentially through these five sections will enable the user to correctly configure the data glove.

4.6.1 Sensor Input Section

This section activates the reading from the glove. A toggle button is in place that enables this function. In addition, the user can set the reading speed from the sensor by changing the speed of the ‘metro’ object which routinely calls for the function read() in the *mote* object. Data is displayed numerically and graphically.

4.6.2 Calibration

The subpatch **calibration** enables the user to retrieve offset and resolution values for the accelerometers’ and magnetometers’ data and offset values for the gyroscopes (Figure 4.17). In order to calculate offset and resolution for the accelerometer and magnetometers, the user must roll, yaw and pitch the sensor box 360°. These movements will enable the patch to read maximum and minimum accelerometers and gyroscopes values along each axis. Once this has been done, hitting the **bang** button will enable the patch to calculate the desired value. By
opening the gate next to \textit{calibration} object in the main patch, we let the values go the data to the inlets of the \texttt{ahrs_triad_mote} object, which computes the data and retrieves the desired attitude information. However, as depicted in Figure 4.18, hand and cube orientation and pointing direction do not match yet. This is because the \texttt{ahrs_triad_mote} object has, by default, a generic magnetic field vector for the Reference system (Earth), which sees the North pole as almost parallel to the horizon. This condition is only imaginary and approximates reality only when at the Earth’s Equator. The ‘On Tour’ section addresses this issue.

\textbf{4.6.3 \textit{On Tour} Section}

The strength of the earth’s magnetic field varies with location. Therefore the True North, or South if we were in the Southern Hemisphere, does not coincide with the Magnetic North registered in a given location. This discrepancy is evaluated in terms of Declination and Inclination.
4. DEVELOPMENT OF POINTING-AT DATA GLOVE

Figure 4.17: *calibration* subpatch. - A screenshot of the *calibration* subpatch in Max.

Figure 4.18: Calibration procedure - Cube’s orientation and green axis’ pointing direction do not match the orientation and pointing direction of the hand.
4.6 Max Sensor Interface

‘Declination is the angle of difference between true North and magnetic North. Inclination is at a given location, the Inclination is the angle between the magnetic field vector and the horizontal plane (the plane is tangent to the surface of the Earth at that point). The inclination is positive when the magnetic field points downward into the earth and negative when it points upward’ (U.S. Department of Commerce 2012).

Figure 4.19: NOAA webpage - Computed geolocation data for Cork (Ireland).

This section enables the user to calibrate the system according to their geolocation. The NOAA logo is a web link to the National and Oceanic Atmospheric Administration webpage (National Geophysical Data Center 2012) (Figure 4.19). Through this page the user is able to retrieve the exact coordinates of the Earth’s magnetic field for location by simply inputting Nation and City (or exact latitude and longitude). Once the webpage retrieves the Declination and Inclination values, the user will simply need to input them into the NOAA section of the Max interface.
4. DEVELOPMENT OF POINTING-AT DATA GLOVE

In addition, the user can further change the declination and inclination values to match the perspective of the cube direction in the Orientation Visualization section. For example, if the computer screen is pointing in the opposite direction of the magnetic north the user could decline $+180^\circ$ to match the views as in Figure 4.20.

![Figure 4.20: Calibration procedure - Result of a correct calibration and geolocation procedure. Hand pointing direction matches cube’s one (see cube’s green axis).](image)

4.6.4 Orientation Visualisation Section

This section displays the orientation of the user’s hand by mean of a cube. The hand’s pointing direction is the cube’s Y axis (green axis).

4.6.5 Calculated Values Section

This section displays the retrieved data after numerical evaluation. The data is:

- Hand’s Attitude - Displayed in Angle/Axis, Quaternion and Rotation Matrix format.
4.7 Summary

- Azimuth and Elevation for X, Y and Z axis.
- Sphere’s Slice - the slice number the given axis is pointing at.
- a copy of the raw incoming sensor data.

Each of these values can be sent via the ‘send’ Max object to any other Max patches where the data is mapped to the sound generating mechanisms designed for the performance.

4.7 Summary

This chapter presented first the research partners involved in the Celeritas project and their respective roles. It then offered a brief history on how the idea of a data glove slowly emerged from the original aim of the project. The data glove developed was named Pointing-at. A description of the hardware and the method to interface the data glove with the host computer was offered. Lastly, the chapter described the Pointing-at software interface developed in Max 5. The interface enables the reading of the raw data from the glove and offers a numerical method, know as TRIAD, for the retrieval of the sensor pack (hand) orientation. In addition, a numerical evaluation of the data allowed for the retrieval of more useful data such as the azimuth and elevation coordinates and the slice number from a virtual sphere. These characteristics of the software were made possible through the development of a custom library of Max externals named AHRS.
4. DEVELOPMENT OF POINTING-AT DATA GLOVE
5

Performances

5.1 Live Performances

In this chapter, a series of performances created as part of this dissertation and employing the Pointing-at data glove will be presented and discussed. Below is a list of the performances and venues where these were presented:

- **Preliminary Works 2007-2008** which include early studies on how to map data to sound and/or visuals.

- **3D-Dj Application** presented at NIME 2008 (5th-7th June 2008 - Genoa - Italy); Discovery Exhibition 2008 (16th-19th November 2008 - Cork City Hall - Cork - Ireland); Discovery Exhibition 2009 (14th-15 November - Cork City Hall - Cork - Ireland); Festa della Creativita’ 2008 (24 - 26 October 2008 - Florence - Italy); (14th-15th November 2008 - University of Limerick); Open Days UL (14th-15th November 2008 - University of Limerick) - (13th-14th October 2009 - University of Limerick) - (12th-13th October 2010 - University of Limerick);

- **Rex Levitatis Dance** presented at Fashion Moves (25th of July 2008 - Science Gallery Dublin - Dublin - Ireland);

- **Molitva** performed at Soundings 2009 (10th of September 2009 - Limerick - Ireland); NIME 2009 (5th of June 2009 Pittsburgh - USA); RE-New 09 (20th of May - Copenhagen - Denmark).
5. PERFORMANCES

- *Mani* performed at NIME 2010 (17 - 18 June 2010 - Sydney - Australia);

- *Agorá* performed at Contemporary Music Centre’s New Music Marathon (9th of March 2012 - DIT Conservatory of Music and Drama - Dublin - Ireland); Lifelong Learning Festival (30th of March 2012 - Ormston House - Limerick - Ireland); Borderline Club (5th of April 2012 - Palermo - Italy).

Video recordings of these performances have been included in the DVD attached to this dissertation and publicly available at [Torre 2012](#).

5.2 Preliminary Works

While the research conducted between 2007-2008 was no longer concerned with the idea of developing a dance suit, it preceded the development of the Pointing-at data glove. Particularly important at this early stage of the research was to ‘understand’ the sensors’ data retrieved. The early WIMU prototype enabled the tracking of angular velocities and accelerations along the three orthogonal axes. These values, dependent on the sensitivity of their internal mechanical components and the resolution of the internal clock, represent our initial ‘technical constraints’. Indeed, the internal clock speed was set to 8 MHz, and it, thus, restrained our range of trackable movements from slow to medium-fast movements. The 12-bit resolution of the AD converter guaranteed instead good precision for further scaling during the mapping process.

A second question concerned the choice of the object to which to apply the sensors for the retrieval of the forces exerted on it. In that regard, the discussion was driven by the analysis of the affordances of the objects examined and whether it was possible to numerically evaluate these characteristics with the sensors available. Thus, the discussion moved from the object’s affordances to the previously examined ‘technical constraints’. However, the reasoning could have followed the opposite direction (from constraints to affordances); the object would have been chosen according to the forces that could been acted upon. To further ease the data manipulation process and the user interaction with the object, it was decided to have one WIMU working exclusively on the angular velocity data and one WIMU working on the acceleration data.
The final instrument included two distinct objects: a DJ turntable and a pair of toy-basketballs, one of which hangs from the roof and the second is hand-held. The WIMU mounted on top of the spinning turntable enabled the calculation of its rotational speed. The WIMU placed inside the balls (one for each ball) enabled the calculation of the acceleration forces exerted on the balls when hit or thrown by the performer.

![Figure 5.1: Preliminary Works - A demonstration of the first developed instrument.](image)

The software used for the retrieval and mapping of the data was Pure Data [2012]. The mapping techniques employed were in the simple form of ‘one to one’. After proper scaling, the angular velocity recorded along the x-axis of the mote on the turntable was mapped to the base frequency of a ring modulator. The three dimensional acceleration forces exerted on the balls were mapped to a random harmonic generator of an additive synthesizer. The data was also mapped to visual elements of the patch which were programmed in GEM, the Pure Data’s
5. PERFORMANCES

visual library ([GEM 2012]). More particularly, the x-axis gyroscope data was mapped to the rotational speed of a sphere and the accelerometers’ data to the x, y and z location coordinates of the same sphere (Figure 5.2 and 5.3).

![Figure 5.2: Preliminary Works - Mapping of gyroscopes data.](image)

![Figure 5.3: Preliminary Works - Mapping of accelerometers data.](image)

A first performance was also developed. Its free improvisational character aimed at exploring the WIMU instrument functionalities in a live performance context. The performance concerned two performers only: a live vocal performer and a performer using the motes as previously described with the addition of a second turntable and, therefore, employing a total of four motes. In order to more distinctly define roles during the construction of the soundscape, both performers agreed at the beginning to work on different sound material. More precisely, the mote-system worked exclusively around additive synthesizer and bell-like sounds, while the vocal performer focused on the live manipulation of her voice while always maintaining a voice-like sound.

With regard to the Motes-instrument, the addition of the second turntable allowed the performer to experiment with beating frequencies by slightly varying the speed of one turntable with respect to the other. Both balls were instead hung from the ceiling waiting to be hit. The performance did not make use of
any of the visual elements previously described (GEM’s sphere) in order to lower the cpu usage.

5.2.1 Discussion and Lessons Learned

The system devised was limited in its functionality and it did not allow for much freedom of experimentation by the performer. However, this preliminary work enabled important considerations that informed much of the later developed work on the Pointing-at data glove, which represents the main tool used for the development of the performances discussed in this chapter.

With regard to the performance, a series of considerations can be made. The cultural constraints here are limited to the use of the novel technology, for which no ecological knowledge could be expected from the audience, and the use of a free improvisational forms. The technical constraints are, however, predominant. The range of valuable data is very limited as only restricted to accelerometers’ and gyroscopes’ data in their raw form. Also, the mapping strategies of the simple one-to-one kind, enabled the control of basic synthesizer parameters such as fundamental frequencies and/or partials. While this creates a robust interface for the performer to interact with, it presents itself as extremely limiting for the degree of control allowed to the performer over the music material. These technical constraints, however, play a major role in clarifying the cause-effect links that exist between gestures and sounds. The simplicity of the mapping strategies in place guarantee a simple connection between the performer’s gestures and the sound output, something that we can refer to, using Emmerson’s terminology, as a strong real local function \[^{Emmerson2007}\]. In addition, the affordances of the system are enhanced by the characteristics of the object hosting the sensors, the balls waiting to be hit or thrown and the speed of the rotary wheel on the turntable.

It appeared immediately clear, however, that the use of multiple motes did not necessarily guarantee more freedom of experimentation for the performer. Of course, the use of the ‘one to one’ mapping could have been replaced with ‘one to many’ or ‘many to one’, thus widening the range of possibilities. However, it was felt that ‘more’ did not indicate ‘better’. The main questions that rose
from this consideration were: what could one single mote be useful for? Is there a better way of exploiting its readings in a meaningful manner? It should be remembered here that one mote was made up of two dual-axes accelerometers, three gyroscopes, two dual-axes magnetometers. This would already guarantee enough data to read and interpret. Indeed, this particular cluster of sensors is mainly devised to retrieve the orientation data with respect to the fixed earth frame of the object that holds it. The calculation of the relative orientation of several motes displaced around the main joints of the human body would have enabled the virtual reconstruction of the human figure and its movement via software. However, as mentioned before, it was felt that working on a single mote would initially narrow the range of possibilities but would later help to more closely evaluate how much could be done with less. The presented demo offers an important clue to the further development of the discussion. A primary concern here is the way the performer handled the ball. Certainly the retrieval of the ball’s orientation could have been useful to the further development of the performance tool. A quick analysis of the balls’ properties would highlight that the ball affords, for example, throwing, squeezing and rolling. Unfortunately, with the sensors available it is only possible to calculate the rotational movements and orientation while leaving out any possibility to retrieve data from its squeezing (which would have required a pressure sensor) and throwing (which would have required position data). It was then thought to extract the mote from the ball and place it directly onto the performer’s hand, thus allowing for the calculation of its orientation in the physical space. The choice of building a data glove seemed, therefore, the most effective because it required the use of one single mote while, at the same time, enabling the programmer/composer to work on a large variety of hand gestures.

The next performances presented here worked around this new devised data glove named Pointing-at. Hand gestures were used to both control the software and enhance communication and readability of a live performance exploiting the controlling, communicative and metaphoric properties of hand gestures.
5.3 3D-Dj Application

3D-Dj represents the first developed application that exploits the orientation data from the glove. It was developed to investigate audio panning over a 3D array of speakers by employing ‘pointing gestures’. Loop spatialisation and overlay enable the performer to create short improvised compositions. Rather than a proper performance, 3D-Dj is an application developed for demo purposes where short musical excerpts could be performed. The work is based around the ideas that have informed studies investigating methods of spatialising sounds over a 3D array of speakers by means of hand gestures such as (Naef and Collicott 2006; Schacher 2007).

5.3.1 Application Features - v.1

The early version of the 3D-Dj (v.1 - 2008) application has two main modes of operation: a browsing mode and performing mode. In the browsing mode, the user is located at the centre of a virtual sphere (Figure 5.4), whose surface has been divided into thirty-two slices, sixteen for the Northern hemisphere and sixteen for the Southern hemisphere. The idea behind the creation and subdivision of this virtual space is that by reading the hand’s orientation in term of azimuth and elevation values for a given axis, we are able to know which slice of the sphere the hand is pointing towards. Each slice is then assigned to a sound loop so that when the user is pointing in the direction of one slice a different sound loop will be played (Figure 5.5).

In performing mode the user places the currently selected loop onto the available speakers’ array. In this mode, the user is also able to move (spatialise) the loop through the available speakers by simply pointing the hand in different directions. This is done by mapping the direction of the Y axis from the glove, given in the format of azimuth-elevation, to the azimuth-elevation coordinates of the

---

1 The application was developed in 2007. Since then many other studies have investigated methods of spatialising sounds by means of hand gestures such as (Marshall et al. 2009).

2 It is important to note that the glove enables calculation of the orientation of the hand and not position. Thus, we can assume that the sphere is always around the performer no matter where they are located in the physical space.
5. PERFORMANCES

Figure 5.4: Performer’s Virtual Sphere - The sphere system is implemented in both v.1 and v.2 of the 3D-Dj App

Figure 5.5: 3D-Dj App v.1 - Selecting loops.
Ambisonic tool developed at the Institute for Computer Music and Sound Technology of Zurich (2012). To switch to performing mode the user crooks the index finger. When the index finger is stretched again the software returns to browsing mode (Figure 5.6). This early version of the software has been presented at the NIME Conference 2007 (please refer to the attached DVD for video excerpt).

![Diagram of mode selection](image)

Figure 5.6: 3D-Dj App v.1 - Method for switching between browsing and performing mode.

### 5.3.2 Application Features - v.2

In the newer version of the software (2009) a function that enables the overlap of multiple loops onto the array of speakers has been implemented. In order to do so, it was necessary to separate the audio output of both browsing and performing mode which are now sent to the headphone mix (user) and main output (audience) respectively. The sequence of gestures that enables the switch between browsing and performing mode needed to be modified as well. The sequence starts with the index finger stretched and the user able to freely browse the sound library by pointing their hand in the direction of any slice of the sphere surrounding them. When selecting a sound the user crooks their index finger. Stretching the index finger will place the sound loop onto a specific point in the array of speakers. This point is determined by the direction in which the index finger is pointing (Y axis). With the index finger stretched, the user is now able to spatialise the sound
5. PERFORMANCES

loop by means of pointing gestures (Figure 5.7). In order to return to browsing mode the user will crook and then stretch their index finger (Figure 5.8). Thus, the user is now able to browse the library once again and, by following the same gestures sequence, ‘throw-in’ multiple sound loops onto the audience space. To delete a loop from the array of speakers, the user needs to select the slice number that matches the loop in question a second time when in browsing mode.

Figure 5.7: 3D-Dj App v.2 - Loop spatialisation mapping schematics.

For some of the developed demos, a mechanism for the control of audio effects in order to manipulate the generated soundscape was implemented. When that was the case, the effects were applied to all the loops that were currently playing in the 3D array of speakers. The reasons that motivated this choice were multiple. Firstly, this strategy allows the user to use the interface with much more ease. Although this may have been seen as a limitation to the degree of user freedom and control intimacy, at the same time it guaranteed a more versatile and immediate use. Secondly, if we manipulated all the loops together, rather than individually, we had a better chance of the audience understanding the link between gesture and applied musical effect. However, the implementation of the audio effect section was never properly developed due to difficulty encountered by
5.3 3D-Dj Application

Figure 5.8: 3D-Dj App v.2 - Gesture sequence in 3D-Dj app v.2.

the hardware in use at that time in managing the heavy computational resources demanded for such a task.

5.3.3 Discussion

The main purpose of the 3D-Dj application was to demonstrate, by means of hand gestures, a method for browsing a large sound-loop collection and to pan/overlay them in a given array of speakers. However, it has been the data retrieved from the glove that guided the development of the application, rather than the opposite. This is what we have previously described as a technically-driven approach. The two main sets of data made available by the glove were the bending sensor data and the accelerometers’ and magnetometers’ data. A numerical evaluation of the latter gave the hand’s orientation in the physical space. These initial technical constraints offered a solid starting point for the further development of the application. With regard to the hand’s orientation, it is important to note that the hand’s movements were not restricted to 3D-rotations around a fixed point in the space. In other words, the hand could be rotated around the wrist as well as following the arm in any direction desired by the performer. Because the
orientation is not dependent on the position, the mote’s orientation would have been almost identical, both in the case of the arm close to the body and with the arm stretching outwards. We could think of the hand and the body of the performer as one single unit orientated in the physical space.

A second consideration was what hand gestures could have been afforded in light of the data available (i.e. orientation data). As previously mentioned, the hand can be used for a myriad of tasks, such as manipulating objects and communicating. The orientation data seemed to best fit the purpose of enabling the computer to recognise human gestures. A properly trained gesture recognition software would have recognised gestures patterns, which then could have been mapped to several audio functions. At an early stage of the application development, it was decided to experiment with the FTM Gesture Recognition Library from Ircam [IRCAM 2012]. Although this research thread seemed to be promising, it was soon preferred to follow a different path for several reasons. Firstly, the development of a gesture library would have served the purpose of demo ap-
plication but it would have been less suited to the scenario of a live performance. It was thought that the higher degree of abstraction between hand gestures and their functions would have led audience being confused in their efforts to understand the device’s functionalities and thus compromising the overall judgement of both the performance and performer’s skill. Overall, such implementation would have forced the discussion too much towards HCI issues and, thus, the way the performer would interact with the device. In addition, a gesture library could have been more easily used by the performer to recall effects ‘on-the-fly’. This is because they are usually present in a limited number during a live performance. However, it would have then been more difficult to recall sounds. A library of twenty sounds, for example, would have required twenty different gestures, unless a sequential browser with traditional ‘next’ and ‘previous’ functions was envisaged.

The alternative was then represented by those common gestures that are normally used in daily communicative practice to enhance or sometimes to replace speech. In this manner, the communicative characteristics of the hand gestures could have been exploited in a metaphoric manner that would have been enacted by the performer in their use of the device. Grabbing and pointing were the two affordances investigated in this application. The strength of these gestures relies on the fact that they both belong to the previous experiences, or ecological knowledge, of both performer and the audience. Their metaphorical use in a performance scenario would then represent a better starting point for the interpretational work by the audience with regard to the mapping employed and the device’s functionalities.

The use of the virtual sphere enabled the performer to freely browse their sound library by using pointing gestures. A reconfiguration of the software could easily increase the number of the sphere’s slices, if desired. With regard to version 2 of the 3D-Dj app, the device’s browsing functionalities are, however, completely hidden to the audience. While the performer browses their sonic space, the audience does not receive any audible feedback because it is all happening in the performer’s ear monitors. The pointing and grabbing gestures will instead produce an audible output when the performer has selected the sound, ‘thrown’ it at one of the available speakers and panned it, if desired, across the available set of
5. PERFORMANCES

speakers. In this case, the browsing function certainly represents a negative aspect of the system because of its lack of communicative properties. This is probably the main characteristic that made the tool more suitable for demo purposes rather than performances. However, if we consider the demo as a very short performance a series of consideration can be made. As previously mentioned, the technical constraint of orientation data played a fundamental part in the development of the software. However, some cultural constraints were also taken into account as the main operational mechanism was developed, although interpreted in a simple manner, around a known approach to live performances such as the use of loops in some Dj performances. These characteristics of the software highlight the reasons behind the decision of not creating clear sound design strategies and methods of interacting with them. The user’s interaction is limited to the selection and overlap of thirty-two loops and their spatialisation onto the available array of speakers.

Compared to the mapping strategies implemented in the ‘early experiment’, the numerical evaluation conducted on the available data has widened the range of mapping possibilities. This, however, did not translate to an improvement of the control intimacy as the software enabled the control of loop selection and spatialisation only. However, the retrieval of the orientation added a strong meaning to the data available and this greatly help in the process of developing meaningful mapping strategies. Mapping pointing gestures to the location of a sound source is a clear cause-effect mechanism. From an audience perspective we can indeed state the device affords pointing and grabbing.

In the ‘early experiment’ section, the direct manipulation of the sound generating mechanism can be described, using Emmerson’s terminology, as a local real function [Emmerson 2007]. In the 3D-Dj app, while the connection between gestures and sound is strong, the user relies exclusively on pre-recorded material. This function is instead referred to as local imaginary. A further peculiarity of the system is given by the way the sonic material is displaced across the performer and the audience spaces (in Emmerson’s terminology we refer to them as frames). If we consider the local frame as the space occupied by the performer and the field frame the one dedicated to the audience, we see that their relationship differs from the usual displacement (Figure 5.10 (a)). Indeed, the use of the
surround speaker configuration, with the user at centre and the audience around
the user and within the area delimited by the speakers, collapses the field frame
onto the local one (Figure 5.10 (b)). It is also correct to state that local frame
function extends and reaches the field one.

Figure 5.10: 3D-Dj Frames - (a) Conventional frames’ displacement for perfor-
former and audience areas; (b) 3D-Dj displacement of frames.

5.3.4 Lessons Learned

In summary, the development of this first application allows for the following
considerations:

- The numerical evaluation of the data allowed for the development of more
  sophisticated mapping strategies.

- The use of meaningful data, such as the hand’s attitude, allowed for the
  establishment of more coherent gesture-sound metaphors.

- The use of communicative gestures (pointing and grabbing) helped to clarify
  the cause/effect link between gesture and sound from an audience perspec-
tive.
5. PERFORMANCES

- The use of centered stage performances helped to enhance the significance of the mapping strategies in use.

5.4 *Rex Levitates* and Pointing-at

*Rex Levitates* is a solo dance performance choreographed by Liz Roche and performed by Justin Doswell. It was performed at the *Fashion Moves* Festival hosted by the Science Gallery Dublin in 2008. This performance employs the use of the Pointing-at glove by the dancer and the software in use presents similar functionalities to version 1 of the 3D-Dj app previously described.

![Rex Levitates](image)

**Figure 5.11:** *Rex Levitates* - Science Gallery 2008 (Dublin - Ireland) - Coreographer: Liz Roche; Performer: Justine Doswell) [Video screenshots source: [Higghins 2008]]

5.4.1 Application Features

In particular, the new application does not have the two ‘browsing’ and ‘performing’ modes that characterised the previous application. The ‘browsing’ function
has been put in place of the ‘performing’ mode. Thus, both performer and audience will now receive an audible feedback from the interaction of the dancer with the Pointing-at glove during the exploration of the sonic space that exist within the virtual sphere surrounding the performer. While the virtual sphere and its subdivision in thirty-two slices mapped to thirty-two audio loops has been retained, the data retrieved by the bending sensor, and therefore from the movement of the index finger, have been mapped in the following manner:

• By keeping the index finger stretched, the dancer can explore the virtual sphere.

• By crooking the index finger, the dancer can manipulate the chosen soundfile by reversing it.

• By closing the index finger, the dancer can hold the soundfile, hence perform to it.

This was achieved by incorporating the middle value (522) coming from the bending sensor into the mapping process (see Section 4.5.2). Thus, with respect to the values coming from the bending sensor we now have the value ‘778’ enabling the ‘sphere browsing’ mode, the value ‘522’ enabling ‘reverse’ function and the value ‘266’ enabling the ‘hold’ function.

Finally, the removal of the ‘performance’ mode also removes the possibility to spatialise and overlay loops over time. The spatialisation feature were removed because of the limited amount of speakers and technology (soundcard, mixer etc.) made available at the place of the performance. The output was limited to a mono channel (Figure [5.12]) (Please refer to attached DVD for video documentation of the performance).

5.4.2 Discussion

The dance performance presented here may be categorized under the label of postmodern. However, as noted by Carlson (1996), it is also true that the distinction between modern and postmodern in dance performances starting from the 1960’s to date have been hotly debated by both theorists and practitioners.
5. PERFORMANCES

As Banes notes, what it is now called postmodern dance has striking similarities to what, throughout the 1930’s 1940’s and 1950’s, was defined as modernism in other arts. The search for the essential qualities of the art form, the rejection of narration and mimesis, the formal abstraction are all elements that belong to modernism and influenced by the ideas of Artaud and his vision of the theater (ibid.). This phenomenological approach was opposed to the expressionism that dominated traditional art. Thus, the older idea of an art object intended as finished and unchanging has been replaced by a ‘performance’ within an ever-changing, playful, disjunctive process of indeterminate form which is wide open to experimentation. In dance, the movement of the body and its relation to the space will likely represent the essence of the art.

In the case of *Rex Levitates*, the absence of narration and mimesis, the set of a museum gallery, a stage delimited only by the audience gathered around the dancer, the primary focus on body and presence, the experimental and improvisational character are all elements that can be found in a ‘modern’ performance. The postmodern characteristics of the performance are instead reflected in the use of reference to classicism and other distant dance cultures and its increased accessibility (against the hermeneutic modernist approach) to both an audience of experts and a general public, in line to what has been defined by Jencks, in architecture, and Banes, in dance, as ‘double-coding’ ([Carlson 2004](#) p. 145). A ‘classic’ element is here, for instance, represented by the use of music. However, in line with a modernist approach, which denies the primacy of the music over the dance, the main difference is that the dancer does not dance to music created...
by a composer or a co-performer, but dances instead to music generated live by
the dancer with the use of the Pointing-at glove. Thus, the music is an ancillary
element that helps the dancer to construct a pseudo-narrative that is generated
*ex-tempore*.

Since the performance makes use of the 3D-Dj app version 1, many of the
observations made with regard to constraints, affordances, sound design, frames
and field function are applicable to this context as well (see Section 5.3.3). How-
ever, the use of the application in the context of a dance performance offers the
possibility for a few more reflections. The virtual sphere surrounding the dancer
offers a more ‘tangible’ and dynamic way of exploring and investigating the body’s
presence within the space. In Laban’s terminology, we can state that the dancer
explores the edge of her kinesphere, a term in the theory of body movement that
describes the space surrounding the body of the performer and delimited by the
reachable edges of its limbs ([Bartenieff and Lewis 1980] pag. 25). Furthermore,
if the dancer’s presence has been conventionally explored in relation to an empty
and unchanging environment, the device allows the dancer to enter into a dia-
logue with a now responsive spatial environment. The explorative processes, the
sense of discovery and the improvisational nature of the performance are given
by the path that the performer decides to follow inside the sphere.

The interaction, exerted through pointing, grabbing and crooking gestures,
affords both an active and passive behavior as regards the dancer in the virtual
environment. Indeed, while browsing (pointing gesture), the sound output is
hardly predictable by the dancer unless a meticulous sequence of movement has
been planned. This unpredictability, also enhanced by the insufficient reliability
of the earliest version of the orientation algorithm[^1] makes the dancer passive and
always responding to the current sound output. Full control is instead exerted
when the current output is put on ‘hold’ (closing the index finger), hence allowing
the performer to dance to it. This active role is also exhibited by crooking the
index finger, which then allows for some simple manipulation of audio such as
reversing it.

A further consideration regards the difference between gestures enacted by a
musician and those by a dancer in order to control the device. If, from an audience

[^1]: The TRIAD algorithm was not yet implemented at the time of this performance.
5. PERFORMANCES

perspective, musician’s gestures are strictly relatable to the end of controlling the musical output, for a dancer the same gestures assume a double meaning being, at the same time, functional to the creation of the musical output and the development of the dance performance. In that regard, the distinction between ‘controlling’ and ‘metaphoric’ gestures suggested by Zhao and McNeill has been weakened (see Section 2.2). In other words, while a dancer’s goal is to use movements as a creative force, a musician uses movements in order to attain a musical goal. The movement is not explicitly considered but it is the visible result of the musician’s strategy for the successful attainment of the musical goal. In the Rex Levitates performance the difference in the genesis of the movement from a dancerly and musical perspective can be considered as a continuum navigated by the performer in her live act.

In that regard, the Rex Levitates performance has strong similarities with the Embodied Generative Music project from the Universität für Musik und Darstellende Kunst Graz [Kunst Uni Graz 2012a]. At the core of the Embodied Generative Music project is the Aesthetic Lab which consists of ‘15 Vicon M2 tracking cameras, a V624 data station, a hemispherical loudspeaker array with 24 or 29 speakers and a planar loudspeaker array with 48 or 64 speakers, the Vicon iQ 2.5 tracking application and IEM’s CUBEmixer sound projection software’ [Kunst Uni Graz 2012a, para 1]. The aim of the project is to allow the dancer to be responsible for the sound generating mechanism through their movements. Thus, the body itself becomes the instrument. It is argued that:

The intuitivity of a digital instrument’s playability depends on how closely its mapping heeds bodily knowledge of sound making. This bodily knowledge is not primarily expert knowledge of an acoustical instrument’s idiomatic playing gestures, but rather knowledge from everyday experience in the qualitative relation between a gesture as making sound via intentional or non-intentional touch, and its corresponding (tactile and proprioceptive) feel [Kunst Uni Graz 2012a, para 4].

A ‘scenario’, presenting a similar idea to the virtual sphere used in the Rex Levitates performance, is the one of Spheres [Kunst Uni Graz 2012a]. In Spheres, a
5.4 *Rex Levitates* and Pointing-at

A series of circular or elliptic virtual spheres are distributed around the stage area. The sounds are produced by the dancer’s movements and interaction with the spheres, each of which represents an FM synthesiser. In particular, the Euclidean coordinates, representing the position of the dancer in the space, are mapped to the brightness, roughness, and fundamental pitch of the ‘spherical’ FM synthesiser.

![Image](image_url)

**Figure 5.13: Embodied Generative Music** - A dancer in the Aesthetic Lab. [Source: (Kunst Uni Graz 2012)]

To a certain extent, we can state that both *Rex Levitates* and the Embodied Generative Music project allow the performer to play with the space. However, there are a few differences worth noting. Firstly, the *Rex Levitates* performances make use of a data glove while *Spheres* employs high-end camera tracking technology. The data glove does not allow the tracking of the performer’s hand position in the space while *Spheres* does. This difference in technological constraints is at the origin of the two different mapping strategies devised in each performance, which sees a single sphere for the *Rex Levitates* performance and an arbitrary number of spheres for *Spheres*. Lastly, in *Rex Levitates*, the performer controls, for the most part, the in and out points of a selected sound loop located at the edges of the virtual sphere. This is achieved by pointing gestures using the data glove. The dancer, although surrounded by the sphere, does not interact with
5. PERFORMANCES

the space within. This mechanism of interaction can be described as a local imaginary function [Emmerson 2007]. On the other hand, in Spheres, the dancer controls a FM synthesiser by interacting with the space within the spheres (not the edges). This can be described as a local real function (ibid.).

In light of these considerations and combining the experience form the previous performances and demos with the experience of the Rex Levitates performance a further consideration can be made. In particular, the idea that the clarity of mapping are related to the communicative attributes of the gestures in use and their use within the context of the performance. The higher the degree of abstraction between gesture and sonic output the less possibilities seem to be to validate gestures in a performative contest while, at the same time, decreasing the chances for understanding the affordative attributes of the instrument and the evaluation of the performer’s skill from the audience perspective. In particular, we are referring here to the crooking gesture that enabled audio manipulation by reversing the current audio loop. While the pointing and grabbing gestures have strong communicative attributes that are functional to both the exploration of the virtual space and the dance, the crooking gesture does not have any clear communicative properties. Also, it cannot be considered functional to the dance discourse. On the other hand, in Spheres, the movements are functional to both the dance discourse and the sound producing mechanism but the relation between the gesture and sounds producing mechanism (FM synthesiser) approximates the abstractness of the crooking gesture in Rex Levitates.

5.4.3 Lessons Learned

In summary, the Rex Levitates performance highlighted the following issues:

- The physical presence of the dancer and her way of playing the space within her kinesphere represented an embodiment of the software virtual sphere thus enhancing its existence.

- Pointing gestures were used outside the spatialisation paradigm and yet retained their metaphorical and communicative properties.
• The relationship between gesture and audio effects parameters is problematic due to the difficulties in finding appropriate metaphors.

• The level of control intimacy is minimal.

• The boundaries between musician and dancer roles have here been reduced.

5.5 Molitva

From the program note for NIME 2009 - Pittsburgh - USA:

Molitva is Macedonian for prayer. Prayer also intended as desire, plea. The work incorporates manipulations of Macedonian sacred chants and songs. Musically, the piece builds on melodic, harmonic and rhythmic elements found in the Macedonian sacred and secular music. This performance has been written for Voice and live-electronics, laptop, 3-D Setup of Speakers and the Pointing-at glove.

5.5.1 Sound Design and Score Macro-Structure

The Molitva performance has been written for three performers: a vocalist, a laptop performer and a Pointing-at performer. Since the early development stage, the main concern for the performers was to clearly differentiate the sonic material across the three instrumentalists. This choice was informed by the need to offer the audience the possibility to more easily identify and link a performer to specific sounds in the soundscape presented. The use of a laptop, potentially able to reproduce any kind of sound, and a newly developed data glove of unknown properties can easily mislead the audience from the creation of the correct mental model and correct interpretation of the live elements of the performance. The final division of the sonic material was as follows:

Laptop performer

Source material is based on live recordings of Macedonian sacred chants. The audio manipulation of these recordings was done previous the performance and
resulted in the creation of several loops. The live interaction is limited to the ‘in’ and ‘out’ insert points of the created loops and the volume and spatialisation values of these.

**Vocalist**

The source material for the vocalist was represented by her own voice and few kitchen tools. Live sampling of these sound sources is made via a loop station. Live manipulation of the sound sources is limited to reverse playback as this is the only feature available by the loop station used by the performer.

**Pointing-at performer**

The performer wearing the Pointing-at data glove works around a grainy-glitchy bank of sound loops and their live manipulation. The bank of sounds was downloaded from the freesound.org database. Keyword search entry for the selection of sound was: ‘glitch’[^1](See next section for more details) (Freesound.org)

[^1]: Unfortunately the author of this dissertation is unable to provide more details on the list

---

**Figure 5.14: Molitva - RE:New 2009 (Copenhagen - Denmark) - Performers from left: Giuseppe Torre (Pointing-at Glove); Dorota Konczewska (Voice and Live Elecronics); Robert Sazdov (Live Electronics). [Courtesy of Eoin Brazil]**
The score of *Molitva* was written so that it highlighted the sonic differences between the performers and more in particular, given the known characteristics of the human voice, between the laptop and data glove performers. In order to do that individual interventions in the timeline were structured so that each instrumentalist had at least one ‘solo’ moment throughout the performance. The performance has three main section: an Intro, a Corpus and a Coda section. Figure 5.15 schematically shows how performers’ interventions and rests were divided across these three sections.

The ending point for a performer intervention is given by the starting point of another performer, therefore serving as a cue. Within these limits, the three performers have room for free improvisation whose boundaries are set by the degree of freedom of expression that their instrument allows for. Lastly, the performers’ interventions are also ordered so that the maximum tension and density of sonic material (climax) is achieved toward the end of the performance (before the Coda section) at which point all three performers are playing at the same time. Since

\[ \text{of sound selected as this data was lost permanently.} \]
5. PERFORMANCES

at that moment in time, both the Pointing-at and the laptop performer had their ‘solo’ intervention, the audience had already the chance to learn what performer is responsible for what sound and hopefully retains this ability during the climax.

5.5.2 Mapping Strategies

Figure 5.16: Molitva - Gesture 1 - Mapping of gestures in the Intro section.

With regards to the Pointing-at glove part, three main gestures were investigated and these are presented in the Intro, Corpus and Coda section respectively. In the ‘Intro’, the spatialisation gestures previously investigated in the 3D-Dj app are exposed. However, the 3D virtual sphere included in the 3D-Dj application that enables the user to browse a bank of sound files has not been implemented.

\[\text{\footnote{The third gesture was performed at the NIME and RE:New performance only. Technical problems occurred at the Sounding performance, the one for which the recorded performance is made available on the DVD attached, did not allow for the third gesture to operate correctly. More in detail, the data glove stopped working towards the end of the performance due to a broken cable inside the sensor pack.}}\]
here. Loops are instead introduced sequentially by stretching and then crooking the index finger. The sequence of stretching and crooking is counted by a ‘counter’ which resets itself and blocks the passage of any data from the data glove once it reaches the number four, having output the fourth loop. When the fourth loop has been triggered all sounds currently playing are faded out. Also in the Intro section, the roll values retrieved by the rolling movements of the hand are mapped to the feedback parameter of a delay effect to which Loop 1 is connected. While Loop 1 is distributed evenly across the whole array of speakers, the remaining three loops are spatialised by mapping the y-axis azimuth value of the hand to the azimuth coordinate in the array of speakers via the VBAP external for Max/MSP (Pulkki 2006 (5.16)).

Figure 5.17: Molitva - Gesture 2 - Mapping of gestures in the Corpus section.

In the ‘Corpus’ section of the score, the mapping strategies in place are similar to the one introduced in the Intro section but with a few variations. Delays and granular synthesis are applied to a series of sounds through the rotational movement of the hand. The counter here is not limited to ‘4’ and thus the performer
can introduce, always sequentially, a pre-ordered list made of thirty-two sound loops (the time constraint, however, rarely allows the performer to introduce the entire sequence of loops). While ‘roll’ movements control the delay feedback parameter as in the Intro section, ‘pitch’ movements affect the grain length/size parameter of the Spectral Granulation plug-in from Michael Norris’s Soundmagic Spectral plug-in library (Norris 2012). Both roll and pitch movements here are affecting all the currently playing loops. The y-xis azimuth coordinate (yaw movement) is mapped to the azimuth coordinate of the spatialisation engine as in the Intro section.

![Figure 5.18: Molitva - Gesture 3 - Mapping of gestures in the Coda section.](image)

The last gesture is introduced between the last part of the Corpus and the beginning of the Coda section. In this section the Pointing-at performer interacts with the vocal performer by recording short samples in real time of the live voice and spatialising these samples. This is achieved by alternatively crooking and stretching the index finger. Crooking the index finger will stop the recording of the live voice filling a buffer in a Max/MSP custom built software loop station.
Stretching the index finger will place the loop onto the available array of speakers and at a location that matches the pointing direction of the hand. With the index finger stretched, the performer is able to spatialise the loop by freely moving his hand. By crooking the index finger again, the loop is permanently placed at the location matched by the latest y-axis’ azimuth and elevation data (pointing direction). Stretching the finger will enable the recording of a second loop. From this point the sequence of events can be repeated. The ‘grabbing’ part of the sequence, the one that enables the recording of the live voice, is performed by pointing the hand in the direction of the vocal performer. While this is not strictly necessary, as the engine is not triggered by the direction in which the hand is pointing, it adds a theatrical element to the performance with the intent of highlighting the mapping strategies in place.

5.5.3 Discussion

Performances that employ more than one digital instrument are arguably confusing. This confusion may arise from the fact that it is difficult, if not impossible at times, to discern who is responsible for each given sound in the ‘sound cloud’ that characterises most electronic or electroacoustic performances. The differentiation of the sound material across the members of the group seemed to be a logical solution to this constraint. In order to further highlight this, the adoption of a structured performance in which at least one ‘solo’ part was given to each performer seemed also to be a valid choice.

To that end, it is relevant to explore how the three performers and their own digital instruments are somewhat different in the ways that they are able to unveil their affordances, constraints and more general properties. The performer, using her voice, was also using a loop station and a few ordinary daily objects such as knives, wooden spoons and the like. From an audience perspective, it is quite easy to understand, recognise and link sounds to the performer in such scenario; performer’s intentions and errors are also somewhat easy to read. The ‘Laptop’ performer is at the other end of the scale. His position, behind the laptop screen, unveils little about the degree of controls and intentions the performer has over his own part. Here, the audience is probably more inclined
to focus on the performer’s ancillary gestures to get some clues on his intentions, but that does not probably tell enough on whether the intentions were successful or not. Conversely, it is plausible to state that one of the affordance properties of the laptop is, at the very least, the ability to playback multiple sounds, control their volumes and displacement. The audience can easily understand that, as this way of using the laptop in a live scenario can be regarded today as ecological knowledge. Indeed, in the Molitva performance, the laptop performer does just that and any speculation from the audience on the possible existence of more complex algorithms and ways of creating or manipulating sounds is only leading to a wrong interpretational path.

In that regard, the performer that uses the data-glove lies in an area somewhat half way between the two situations described above. The level of complexity of the mapping strategies adopted and the degree of gestural abstraction versus communicativeness play a fundamental role here in whether the instrument approximates the ‘secrecy’ of a laptop performer or the clarity of a voice and electronic one. The gestures presented in each section are a clear example of this. The pointing gestures enabling spatialisation, presented in all three sections, could be described as having a simple mapping design (one to one) and a high level of communicativeness (pointing gesture). The grabbing and releasing gestures enabling the playback start control of a given sound loop also have similar attributes. Thus, they greatly clarify the performer’s intentions and their outcomes form an audience perspective.

Conversely, a more obscure control movement is given by the gestures mapped to several parameter of the audio plug-ins in use. Although, as previously described, the link between gesture and effect parameter is direct (one to one mapping), the use of multiple data coming from the hand orientation suggests that a more complex mapping design (one to many) and a higher degree of abstraction (a wavy hand movement that controls both delay and granular synthesis at the same time) have been employed.

In light of these considerations, it is interesting to see the relationship between the control intimacy allowed by the instrument and the clarity of the cause/effect links enabled by the mapping strategies in place. It would appear that an increase of clarity of the mapping strategies in place would correspond to a decrease in
the level of control intimacy perceived by the performer and vice versa. A trigger control such as the one enabling the playback of sound loops and their spatialisation via grabbing and pointing gestures has clear communicative attributes, but it greatly limits the degree of control for the performer. Conversely, a continuous controller, such as the many control parameters of sound effects, controlled by hand tilting movements increases the level of intimacy perceived by the performer. However, given the abstract relationship between the tilting hand gestures and their effect parameters counterpart, it is difficult to interpret, and thus creating the correct cause/effect model, by an audience.

In addition, the scenario in which these gestures are enacted is different from the one seen in the *Rex Levitates* performance. A first distinction is the lack of theatrical elements which enable differentiation of the audience’s attention on aspects of the performance that were aside from ‘controlling’ and mastering sounds. The metaphoric and controlling gestures’ attributes identified were blended into a unified continuum which sees them enacted by the dancer’s movement. The dancer’s movements also explore the space and control and interpret the sounds. In *Molitva*, these movements are presented in a different context in which the controlling and communicative properties of gestures are highlighted. Moreover, music, a secondary element in the dance performance, is the dominant element here. The gestural interaction with the device is thus seen to exclusively control sounds over time and is enacted through gestures featuring strong communicative properties such as pointing and grabbing.

An exception to this lack of theatrical elements is maybe given by the performer’s actions in the ‘Intro’ section as performed in Pittsburgh and Copenhagen, but not in Limerick, and the gesture in which an interaction between the data glove performer and the vocalist take place. The initial sequence of ‘throw-ins’ and consequent spatialisation is acted in the audience space. By leaving the stage and breaking the ‘fourth wall’, the performer has placed himself at the centre of the hemispherical set of speakers. This action morphs and dynamically rearranges the *frames* from a traditional configuration (Figure 5.10(a)) to a wholly centered one (Figure 5.10(b)). The implications of this performing action are many. Firstly, the pointing gesture that enables the spatialisation of the sound loop results in more precision; it is from the center of the speaker array rather
5. PERFORMANCES

than from the edge of the circle, where the stage is usually located. Secondly, since the spatialisation of sounds is at the core of the Pointing-at’s functionalities, the instrument is virtually extended to the speakers’ array of which it gains control. The performer is at the centre of the array, and thus attention to this feature is enhanced. The relationship between glove and speakers has been enacted in a more theatrical manner. Lastly, the more usual implementation of complex surround systems in theaters and concert venues has destabilised the primacy of the traditional stage area. It now expands to the entire perimeter of the building, with the audience fully immersed and thus is virtually located at the centre of the artistic scene. The performer’s movements towards the audience area in the centre of the theater, can therefore be interpreted as a way of addressing the inadequacy of the traditional separation between stage and audience areas found in the vast majority of theater and venues\footnote{Few examples of Central-stage theater exist such as the Cockpit Theater in London and other theaters across the US.} but also as a way to let the audience feel and live with the performer the performing space.

5.5.4 Lesson Learned

In summary, the Molitva performance highlighted the following issues:

- Live electronic performances employing more than one instrument can be problematic to the goal of achieving clarity of the cause/effect mechanisms from an audience perspective. A clear differentiation of the sonic material in use by each performer can at least help relate sounds to a specific performer.

- Finding adequate gestural metaphors to link gesture to audio effect parameters is again complicated due to the high level of abstraction that characterises an audio effect.

- The implementation of simple mapping strategies decreases the perceived control intimacy by the performer, but, if supported by strong gestural metaphors, it can enhance the audience’s understanding of the device’s properties.
• The level of control intimacy perceived by the performer seems to be inversely proportional to the possibility of creating clear cause/effect links from an audience perspective. For example, the pointing gestures, characterised by a low level of control intimacy, are clear to the audience. Conversely, the control of audio effect parameters through continuous controller enhances the perceived control intimacy by the performer, but it is highly enigmatic for an audience.

• The traditional displacement of audience and performance in public venues does not help highlight the pointing and grabbing affordances of the Pointing-at data glove. A Central-stage theater would suit better the scope.

5.6  Mani

From the program note for NIME 2010 - Sydney - Australia:

Mani means hands in Italian but it also sounds like the word ‘money’ in English. Our future is in few people’s hands. Do you feel safe?

Figure 5.19: Mani - NIME 2010 (Sydney - Australia). - The performers from left: Robert Sazdov (Laptop), Giuseppe Torre (Pointing-at glove).
5. PERFORMANCES

5.6.1 Sound Design and Score Macro-Structure

Mani is a performance for two performers using a laptop and the Pointing-at glove respectively. As in Molitva, the performance is designed so that each performer has at least one ‘solo’ part throughout the live act in order to facilitate the audience in understanding which of the sonic material presented comes from each performer. This time, the design and sequence of interventions follow the sequence of events outlined in a storyboard. Thus, important theatrical elements are added to the performance. Indeed, the performers are standing behind two lecture podiums and wearing formal suits. These imitate the scenario of two institutional figures discussing political and economical issues in front of the press or an audience. The theatrical elements, the theme chosen and the instruments selected enable the composer to follow a storyboard while striving for gestural communicativeness.

As depicted in Fig. 5.20 the first two sections of the score (‘speech A’ and ‘speech B’) represent two solos and metaphorically represent two people expounding on their ideas: the laptop performer stressing the importance for, at any cost, a constant economic growth and the data glove performer outlining the poverty spreading around the world and arguing for social equity.

![Diagram of Mani - Macro Structure](image)

**Figure 5.20: Mani - Macro Structure** - Coloured boxes denote performers interventions.

The ‘overlap’ section (Fig. 5.20) sees the two speeches overlapping which results in an increase of the musical tension, as in a moment of strong disagreement or hard debate. The ‘Finale’ sees the two parts coming to a compromise, or to
something that could be described as a mutual agreement. Despite that, the intensity of the sonic material surpasses the one of the previous section, creating a contrast between what is acted on stage and what is perceived sonically by the audience.

The opposite point of views of the two characters are expressed sonically as well. The sound sources of the laptop performer are based upon recordings of short audio excerpts from several economic news reports (BBC, SKY, ABC etc.). The short excerpts were then modified and processed previous to the performance in order to create a library of drones and confusing report speeches. The performer’s live interaction is limited, similarly to the *Molitva* performance, to volume and spatialisation controls. The data glove performer uses short audio recording of news reports as well but with news related to poverty and other social issues. These short excerpts are modified and distorted live using several audio effects units in Ableton Live (section ‘overlap’) (Ableton 2012). Other sounds are generated by controlling an additive synthesiser (section ‘speech A’). In the ‘Finale’, the sounds are heavily processed drone sounds created by Robert Sazdov (laptop performer). All these elements are controlled by a mixture of automated systems and live interactions by the hand movement. The next section will outline the methods employed.

### 5.6.2 Mapping Strategies and Pointing-at Score Part

The Pointing-at performer controls several aspects of the performance. This is achieved via a combination of Max/MSP and Ableton Live patches. The system’s signal flow sees the Max patch serving as the application for the retrieval and numerical evaluation of the data coming from the glove. The Max patch also sends MIDI messages to the Ableton Live patch in order to control its interface. At the same time, Ableton internally routes the audio signal back to the main Max patch in order to spatialise the sound onto the available array of speakers (Figure 3.21).

The main Max patch interface resemble a graphical score. The three sections in which the Pointing-at performer intervenes are presented in an ordered manner so that they can be read left to right. Since, as explained below, the patch
5. PERFORMANCES

Figure 5.21: Mani - Setup - Signal flow in Mani.

includes several automated systems, the patch presents several LED objects whose purpose is exclusively to give visual feedback to let the performer know the correct functioning of the underlining automated systems and the correct activation of each mechanism at the right time. A ‘panic button’ resets all controls to their initial values in both Max and Ableton (Figure 5.22).

Speech A

The ‘speech A’ section presents a combination of real-time elements and automated systems. The latter works as a triggering device that enables and disables control functionalities of the live interaction at precise moments in time. At the start of the live act the performer hits the ‘0’ key on the laptop keyboard which activates the playback of a sound loop 3’33” long, approximately corresponding to the duration of the section. In the given space of time the performer has the possibility to introduce live elements on top of the ongoing background loop in two different manners which are triggered sequentially (i.e. it is not possible to invert their order of appearance). The duration between these two is not set but the performer needs to presents both within the time constriction set by length of the background loop.

The red and orange boxes in Figure 5.22 refer to the sections in the patch
Figure 5.22: Mani - Max patch. - Outlined in this picture the three section of the score. The black arrow indicates the way the patch should be read.

Figure 5.23: Mani - Section Speech A score - Automated score events and performer’s live interaction distribution.
5. PERFORMANCES

that enable these two ways of interaction. In the area delimited by the red box, the designed system allows for the manipulation of an additive synthesiser. The performer is able to control the strength of twenty odd partials of a 60Hz sine tone. This is achieved by mapping the y-axis azimuth and elevation coordinates to the list of partials’ amplitudes. The coding involved the use of the `mnm.list2row` object from the FTM library ([IRCAM] [2012b]) which enables the scaling of a list of ‘n’ elements to a single row. In order to make this system work, a calibration step is required previous the performance. In order to interpolate between the data, the software needs to compare input data (y-axis azimuth and elevation) against a stored row of numbers which reflects the strength of each partial. Three hand positions were recorded and mapped to three graphical representation of the partials’ strength as depicted in Figure 5.24. Once this calibration step has been done, the software is able to interpolate values between the three positions.

![Mani - Gesture 1](image)

**Figure 5.24:** Mani - Gesture 1 - Hand’s pointing directions and their corresponding partial strength’s graphs.

The area delimited by the orange box includes the patch enabling the second mode of interaction by the performer. Here a mix of automated and live elements are presented. By hitting the ‘0’ key once again the performer deactivates the
additive synthesiser previously described. By pressing the same key again the new mode of interaction is activated. Now a timer controls the playback start point of four loops (loaded in Ableton but spatialised via Max; see Figure 5.21) which are introduced sequentially (loop1 start time 0”; loop2 start time after 20”; loop3 start time after 40”; loop4 time after 60”;) (Figure 5.25).

Figure 5.25: Mani - Section Speech A - part 2 score - Automated score events and performer’s live interaction distribution.

For each new loop introduced, the performer is able to control its volume and its location in the available array of speakers (Figure 5.26). When a new loop is introduced the previous loop is automatically placed at one of the corners in the array of speakers. The entire sequence of events lasts approximately eighty-five seconds, after which all sound loops are faded out in three seconds. At this point in time, the performer waits for the end of the background loop (if not already ended), which marks the end of the speech A section and the beginning of the speech B section in which the laptop performer intervenes.

The estimated length is only approximated because the counter object in Max follows the internal Max scheduler and not the cpu clock.
5. PERFORMANCES

Overlap

In the section ‘overlap’ the performer controls the playback starting time of the other four loops in a different manner to the previously described one. Each loop is activated via laptop key controls. Loops one to four have been assigned to the ‘n’, ‘m’, ‘,’ and ‘.’ keys respectively. Thus, the performer is able to decide at which moment in time each loop should be played. Spatialisation controls are retained and, as usual, assigned to the y-axis azimuth and elevation values of the data glove. In addition, the x-axis elevation values are here mapped to the ‘freeze probability’ parameter in the Grain Strainer plug-in (Norris 2012). Lastly, the performer can fade out all loop at any time by hitting the ‘/’ key. This event marks the end of the section.

Finale

In the ‘Finale’, the gestural mapping control is activated by a handshake between the two performers. The hand-shaking gesture triggers an algorithm, which then generates non-repeatable random numbers between 0 and 6 and sends them, via MIDI, to the Ableton patch on the computer of the second performer which hosts seven pre-recorded loops\(^1\). At this machine each of the seven pre-recorded loops has been set to receive a specific MIDI message to enable ‘play’ each time a given number of x-axis ADC acceleration values above a fixed threshold were

---

\(^1\)This was done in order to lower the CPU usage on the Pointing-at performer’s laptop.
Figure 5.27: Mani - Section overlap score - Performer and user triggered events.

Figure 5.28: Mani - Gesture 3 - Gesture mappings in section 'overlap'.
5. PERFORMANCES

found (Figure 5.30). In other words, there is a constant increase in the density of sonic material (Figures 5.29). Once all the loops have been triggered, the performers prepare to suddenly interrupt the hand-shaking gesture (arm pointing down which also means y-axis elevation less than -60°) that is recognised by the computer as ‘stop all loops’. This gesture ends the performance.

Figure 5.29: Mani - Section Finale score - Graphical displacement of the loops over time as caused by the handshaking gesture.

5.6.3 Discussion

As a performance, Mani investigates the communicative properties of hand gestures in order to enhance the readability of the mapping strategies and/or device properties for the audience’s perspective. In order to achieve this, and also to add to the palette of interactive gestures described in the previous performances, it has been envisaged to introduce theatrical elements to the performance as it was thought that these can help find appropriate metaphors for the mapping strategies to be devised.
Figure 5.30: Mani - Gesture 4 - Mapping strategies for the handshaking gesture.

The creation of a theme for the performance is what will then allow for the gestural metaphors in place. In that regard, in Mani, as in the Rex Levitates performance, the influences of postmodern theater are strong. In particular, the similarities with the political performances of the 1990’s and first decade of the current century are evident. Early examples of politically aware theatre can be found during the 1960’s and 1970’s when major political events, such as the Vietnam War and the Cold War, were heavily affecting security and provoked a demand for peace. With the end of the Vietnam War, the politically aware theatre production seemed to decline slightly, only to flourish again in the 1990’s to the present day. The main themes addressed were gender, sexuality and ethnicity. In particular, there was a new tendency to ask who was responsible for the political and social crisis of identity rather than opposing the crises of the 1960s -70s. Feminist and gay performers were probably ahead of other segments of society because performances that investigated the subject from this perspective can be found as early as the 1960’s with artists such as Yoko Ono and Spalding Gray.
5. PERFORMANCES

In *Mani*, the socio-political issue investigated regards the inequalities that the monetary system produces. Performers’ costumes and the presence of two lecture podia are the visual elements that place the performance in an unspecified contemporary time and a well-defined space: the press conference. The two performers, acting as opposing characters, enact the social battle between financial interests and social justice interpreted by the laptop and Pointing-at performers respectively.

The subdivision of roles enables the gestural vocabulary in place to work both as mechanism to control sounds and meaning carrier because carrying communicative properties that would fit the theatrical performance and its theme.

The beginning of the section ‘speech A’ offers a first example of this kind of gesture. Theatrically, it is accusatory; it is movement directed to the audience (the hand moving right to left and vice versa). It is an invitation to silence and reflection enacted by the hand pointing up and placed close to the mouth of the performer. These movements were mapped to the intensity of a given number of partials which changed the drone-sound waveform (Figure 5.24). Given that the path followed by the hand was improvised, a series of points in the path were used as ‘sonic’ anchor points which were giving clear and distinct audible results. These were: hand pointing far left, far right and straight up (the gesture that invited silence). Thus, in order to clarify the mapping of gestures, it was now up to the performer to improvise, while also constantly reiterating these anchor points, engaging in a sort of learning session with and for the audience.

The ‘hand-shaking’ gesture enacted in the ‘Finale’ also presented strong communicative and theatrical properties. In this simple and common gesture we could read mutual respect, agreement or simple salutation. In the context of the performance, it represents the formal handshake politician would enact at the end of their public speech to signify agreement or perhaps a more vague ‘we will work together in mutual interest’ or ‘Ok for now. We will discuss the matter in the near future’. The gesture is linked to the musical output so that the length of time of the handshake is directly linked to the input cue of seven numerically labeled loops which were then introduced progressively in a random order.

It is hard to believe that this mapping was fully understood by the audience: the random order of the entry points of the loops cannot be heard without hav-
ing seen the performance at least twice. However, the audience can see that the length of time of the handshake was linked to the quantity of sound material outputted. As time passed, the sonic landscape becomes fuller. The link between the handshake and the sound is further highlighted by the final gestures, which saw the two performers abruptly ending the handshake which resulted in an abrupt arrest of all the sounds. However, the combination of controlling gestures that also have strong communicative properties in a metaphoric context (the performance) seemed to be a valid approach. As in the *Molitva* performance, mapping gestures with strong communicative properties seems to enhance the clarity of the cause/effect link between gestures and sound. Conversely, the weakest mapping strategies are the ones in which the level of abstraction is higher, such as the mapping of the hand’s movement to various effect parameters (e.g. the x-axis elevation value mapped to the freeze probability parameter of the Grain Strainer plug-in). The same relationship was found between the control intimacy and the trigger/continuous modes of interaction. Increasing the control intimacy by means of continuous controller did not seem to enhance the clarity of the cause/effect mechanism and vice versa. In *Mani*, however, this is not the case in the last part of ‘speech A’, where the performer is able to control spatialisation and volume values at the same time. This gesture retains its clarity not only because of the spatialisation feature but also because of the simplicity of the mapping strategies in place with regard to volume (hand up will result in a louder volume and vice versa). The simplicity of the mapping can be thought as the mapping of control parameters that forms part of the ecological knowledge of an audience such as volume and pitch.

Many elements in the performance are controlled via automated systems. In Emmerson’ terminology these are referred to as *field functions*. Although the performer still retains some sort of control over few of these functions, such as the volume and spatialisation, their entrance points in time are automated. These elements of the performance need to be interpreted as events happening outside the *local frame* of action. Thus, clarity of the working mechanisms behind these is not required nor is it the aim. The attribute of these field functions (sound loops) are in many both real (voices) and imaginary (synthesised drones and landscapes).
5. PERFORMANCES

With regard to the frame’s displacement in the performance, the sonic material extends to both the performers and the audience space: events on stage created by the performers and the arena and landscape extending to the audience space. It is visible how the combination of the sonic material in the performance, whether produced live or by means of automated algorithms, produces a constant re-interpretation of the quantity of activity located in each frame. In ‘speech A’ the activity of the event/stage frame sits on top of an ever changing landscape frame of imaginary field function (low synthetic sound) which transforms itself onto a real frame function located in the arena field (baby crying). Seemingly, the sonic material presented in the Finale seems to belong entirely to the arena frame to then move abruptly to the event frame as soon as the the handshake is interrupted causing the stopping of all sounds.

Informally gathered feedback from both expert and non-expert members of the audience suggested that the gestural performative actions presented here was successful in generating an enjoyable experience based on a sufficient understanding of the performance and its elements.

5.6.4 Lessons Learned

In summary, the Mani performance highlighted the following issues:

- Combining the gestures’ communicative properties in relation to the theatrical performance and its theme, while retaining their controlling functions, helped in designing effective gestures in which both properties were highlighted.

- The gesture presented in the initial part of Speech A gives an example of three-layer mapping in which the controlling of the instrument goes hand-in-hand with the communicative properties of the gesture performed. In addition, the instrument allows for a good degree of control intimacy from the performer perspective while, at the same time, offering strong metaphors to highlight the cause-effect mechanisms behind its control and affordances. Once again, the introduction of the theatrical theme is the key element that allowed for this.
The ‘handshaking’ gesture offers less control intimacy (almost none) as the system is controlled by a random number generator algorithm. This translates to the secrecy of the gesture from an audience perspective. However, the communicative effectiveness relies on the final movement (hands releasing their grasp) which, by stopping all sounds, offers an insight (or an opportunity for speculation) on some of the underlying mechanisms.

The combination of both automated systems and live interactions by the performer creates a constant re-interpretation and movements of the ‘frames’. This, while at times helping to discern what is live and what is not, facilitates the comprehension of the theme of the performance and, thus, offers clues for the correct interpretation of the gestural vocabulary in place from an audience perspective.

5.7 ‘AGORÁ’

There are in our existence spots of time,
That with distinct pre-eminence retain
A renovating virtue, whence-depressed
By false opinion and contentious thought,
Or aught of heavier or more deadly weight,
In trivial occupations, and the round
Of ordinary intercourse-our minds
Are nourished and invisibly repaired;
A virtue, by which pleasure is enhanced,
That penetrates, enables us to mount,
When high, more high, and lifts us up when fallen.

(Wordsworth 1850, verses 208-218)

5.7.1 Concept

Agorá is a voyage inside the performer’s musical and acoustic memories. The title, derived from the ancient greek ἀγορά, refers to the public square, the heart of old greek towns, in which business, political debates and many other social
5. PERFORMANCES

Figure 5.31: *Agorá - CataRT* - Screenshot of the CataRT application and its settings for Agorá.

activities took place. In that regard, Agorá is a metaphor which sees the performer placed at the center of the Agorá while observing and interacting with its surroundings (i.e. his own musical memories). The theme of memory is inspired by the Wordsworth’s poem excerpt (above), which deals with the ability we posses to recollect past experiences in our life and how these memories can help to ‘lift us up’ in dark moments of our lives. In the performance, the theme of memories is also considered in terms of our inability to fully re-live all the sensations experienced in the original instance of the memory, as these are intrinsically connected to the present temporal dimension (the here and now) of the real experience lived. In that regard, the performance enacts the effort in trying to bring up these memories with all the sensory cues associated with them when initially lived. Proust explains this well in the following excerpt:

‘And I begin to ask myself what it could have been, this unremembered state which brought with it no logical proof, but the indisputable evidence, of its felicity, its reality, and in whose presence other states of consciousness melted and vanished. I decide to attempt to
make it reappear. I retrace my thoughts to the moment at which I drank the first spoonful of tea. I rediscover the same state, illuminated by no fresh light. I ask my mind to make one further effort, to bring back once more the fleeting sensation. And so that nothing may interrupt it in its course I shut out every obstacle, every extraneous idea, I stop my ears and inhibit all attention against the sound from the next room. And then, feeling that my mind is tiring itself without having any success to report, I compel it for a change to enjoy the distraction which I have just denied it, to think of other things, to rest refresh itself before making a final effort. And then for the second time I clear an empty space in front of it; I place in position before my mind’s eye the still recent taste of that first mouthful, and I feel something start within me, something that leaves its resting-place and attempts to rise, something that has been embedded like an anchor at a great depth; I do not know yet what it is, but I can feel it mounting slowly; I can measure the resistance, I can hear the echo of great spaces traversed.’ (Proust 1982, p. 48)

The performers memories are then in the form of music and sounds. Within the process of discovery is the desire of re-appropriation of something that is, in its essence, lost forever.

The performance develops with the agorá metaphor in mind. The first section (Part A) sees the performer unveiling his musical memories, which slowly come to focus. The metaphor would here see the performer standing at the center of the agorá in the act of slowly refocusing the scene around him after a long period of self-imposed isolation, despite all the noise that has been surrounding him. In the second section (Part B), the performer interacts with his musical memories with the intent of studying them. Thus, the aim is to re-appropriate the memories and reflect on them in order to ‘understand’ the lived experience. This action is a sort of personality development process in which the performer is involved in first person. The third section (Part C) is a revelation. Sounds from the subconscious emerge to the mind consciousness revealing the ‘real’ underlining
reasons for the actions made. That is, the memories were only paths to the most precious memory of all, something that is, however, lost forever in its essence but for which the mind, by means of interlocked pathways, used its defensive mechanisms to safely preserve the related memory.

5.7.2 Development

Each of the three sections presents a different performer’s paradigm of interaction with his own musical memories. Their development is described in the sections below.

5.7.2.1 Technical Specifications

Agorá is written for a solo performer using the Pointing-at data glove and it requires a quadraphonic audio system. The performance is written using a combination of Max and Ableton Live patches with the former controlling the latter via MIDI and OSC messages.

5.7.2.2 Part A

The sonic material at the base of the performance is represented by forty songs selected from the composer’s personal digital library.\footnote{The complete list of forty songs is given in Appendix G.} Each song represents then a musical memory which is also, at times, associated to a composer’s personal experience. The unveiling process that characterises this section is enacted by the performer through a series of gestural mapping techniques and automated systems. In that respect, the gestural vocabulary in place is built in order to highlight the unveiling action. The visual reference to this gesture is found in multidisciplinary projects which allow the manipulation of particles systems through physic emulators libraries such as the MSAFluid \cite{Akten12} developed for the open-Frameworks C++ toolkit \cite{Lieberman12}. A physical modeling library, initially developed for Pure Data but also available for Max/MSP, is the pmpd library \cite{Henry12}. This library was the ideal tool, as it did not necessitate
the use of extra software (such as the one eventually developed in openFrameworks). Indeed, it could have been used within the Max environment, already in use for the glove data retrieval and manipulation, and avoided the parsing of data between different applications, which could have ultimately effected cpu performances\footnote{At this stage of the development, the use of the Ableton Live software was already decided. Thus, the simultaneous use of the two applications (Max and Ableton) was thought to be already enough from a cpu load perspective.}

Figure 5.32: MSA Visual - Video screenshot - A screenshot from the MSA Fluid test video showing the MSAFluid library at work (Ivanov 2009).

In line with the visual projects mentioned above, the aim here is to find a metaphor to rearrange the perceived visual movements in acoustic terms. The created metaphor sees the tail of the visual particle matching the volume of the sound (a sort of fade-out). The gestural mapping generated follows this metaphor, too, based on the sweeping-like movement of the hand-arm in the air as if to remove dust from a furniture without physically touching it (i.e. the dust is removed by the wind (here volume) created by the hand’s movement near the furniture).

This hand gesture controls a series of elements in the software. The software itself contains a modified version of the 3D-Dj application described earlier in
this chapter. For the purpose of clarity, let us assume for now that the user is able to control the volume of a sound loop found in a slice of the virtual sphere surrounding the performer. The volume parameter is then controlled by a combination of user input gestures interfacing with a custom-built abstraction made with the `pmpd` library, which ultimately controls the output volume for the loop. More in detail, each time a new `slice_y` value is found, a `bang` triggers a physical modeling patch that emulates the motion of a spring of given rigidity, length, thickness, damping of link deformation and damping of mass speed values. The output values of the spring emulator module are then scaled and sent as MIDI control change messages to Ableton Live to control the volume of the loop. However, it is important to note that the spring, when receiving the ‘bang’, will oscillate only once as to recreate a sort spring-like motion in which loop will fade-in fast and fade-out slower (i.e. the wind power exerted on the dust at the passage of the hand).

The aforementioned assumption that the user could control a single loop can now be disregarded. In fact, the patch allows the user to control, or at least interact with, a library of forty songs. These songs are hosted in the Ableton Live patch and distributed randomly across eight Ableton audio tracks each of which hosting five songs. The selection of the loop is controlled via Max. The same bang that triggers the spring emulator also triggers a random generator unit that controls which song will be played and its starting point within the song file via MIDI note and MIDI control change messages respectively. There are eight of this module in the Max patch, each of which controls one of the eight Ableton tracks.

Lastly, the spatialisation engine is implemented here in Ableton via the Am-bipanner 1.0 Max for Live audio device (Moore 2011), which is controlled by the y-axis azimuth values scaled in the range between 0 and 127 and sent from Max to Ableton via MIDI control change messages.

\[1\]The value indicates the slice number the y-axis is pointing towards. Only Northern hemisphere of the virtual sphere is considered for this section of the performance.
5.7 ‘AGORÁ’

Figure 5.33: Agorá - Part A - Gesture mapping and automated events in Part A.

Automated systems - Part A

There are several elements in Part A that have been pre-programmed and for which the performer does not have any real time input. At the beginning of Part A, the performer controls the volume of a ‘whoosh’ sound loop created by combining in line the following sound sources: [FxProSound 2012a b c]. This sound could be thought of as the ‘dust’ in our metaphor. The way the volume of this loop is controlled is identical to the way described earlier in relation to the memory songs (spring emulator module). The way this whoosh sound is slowly faded out, leaving space for the emerging memory songs, is, however, automated by dynamically changing over time the rigidity, length, thickness, damping of link deformation and damping of mass speed values of the spring emulator modules of both the whoosh and memory songs. In other words, the stiffness of the spring for the whoosh loop will increase over time while decreasing for the memory songs modules. This action, combined with the performers’ gestural input previously described, is the second element that helps enacting the metaphor of removing
the dust to unveil the musical memories. A further element is given by the automated dry parameter value of a reverb unit in Ableton that moves from a high to low value. A high amount of reverb slowly fading out over time emulates the sound memory moving from a distant and out-of-focus place to a closer and in focus location. The length of this automated event is set according to the algorithm discussed in Section 5.7.2.5 (Figure 5.37). Part A ends abruptly with the entrance of a breaking glass sound triggered at the end of automated sequence of pre-programmed events and evenly spatialised across the array of speakers.

Figure 5.34: Agorá - CataRT - Screenshot of the CataRT application and its settings for Agorá.

5.7.2.3 Part B

Part B is the section in which the performer manipulates and aims to gain control over his musical memories. Thus, the source sound material is once again made
of the forty selected memory songs. The section presents also a mixture of real-time and automated elements. Three modes of real-time interaction are devised in this section and are enabled sequentially via an automated chain of events. These events are triggered over time and their active mode is signaled via specific automatically triggered sounds or events appearing throughout the section (please refer to Section 5.7.2.5 - Part B for more details). We will call these modes with the general names of \textit{Freeze, Hey} and \textit{Resonator}.

In \textit{Freeze} mode, the user is able to freeze and loop short excerpts (few millisecond long) taken from the forty memory songs. This signal flow enabling this mode sees a combination of the main Agorá patch, the Ircam’s Cata-RT Max application (IRCAM 2012) and Ableton Live. The Cata-RT application is a software enabling corpus-based concatenative sound synthesis, which is based around the idea of storing a large data set of short sounds snippets retrieved from the analysis of the selected sound library (i.e. the forty songs) in order to reassemble them according to the selected descriptors extracted from the analysis (Schwarz 2004). Thus, the forty songs are imported, analysed and reorganised in a Cartesian graph according to a selected descriptor in the CataRT application. The software has been set to repeatedly select random snippets from the loaded audio library. The audio output of the CataRT app is then routed into four dedicated Ableton audio tracks, each of which is connected to a single audio output channel (no spatialisation allowed here) and faded in automatically. In each Ableton audio track, an instance of the Freeze 1.0 plug-in has been loaded (Egor 2012). The user interaction is limited here to the deactivation and activation of the freeze function in the Freeze 1.0 plug-in by crooking and stretching the index finger (bending sensor data) respectively. The user can decide with which instance of the plugin (audio track or speaker) to interact by pointing the hand in the direction of the speaker he chooses. Thus, having divided the circumference around the performer in four segments, the y-axis azimuth coordinate works as a splitter that routes the bending sensor data to the desired channel (speaker or Ableton audio track). Each audio track also has an instance of the native Able-

\footnote{Diemo Schwarz has recently published a paper for NIME 2012 in which suggest different ways of interacting live with his software in a live performance scenario (Schwarz 2012).}
5. PERFORMANCES

The automatic fade out of all the ongoing sounds signals the enabling of the Hey mode. In this mode the performer is able to control the in-points of four vocal samples screaming ‘Hey!’ The sound selection is achieved by using the Y elevation and azimuth coordinates retrieved by the pointing direction of the hand. If the elevation value is lower than a certain threshold the performer is able to send a ‘start playback’ command (via MIDI message) for one of the four loops by crooking and stretching the index finger (bending sensor data). In order to select one of the four loops, the azimuth value needs to be within one of the four segments of the circumference around the body of the performer.

The automatic introduction of sounds resembling distant voices signals the enabling of the Resonator mode. In this mode the performer can control the dry/wet parameter of the Ableton’s Resonator native plug-in. The parameter is controlled by the x-axis elevation values (tilting movement of the hand) scaled and sent via MIDI to Ableton.

Figure 5.35: Agorá - Part B - Gesture mapping in Part B.
The automated elements in this section are used to signal to the performer the enabling of a new mode of interaction with the software. The appearance of these elements is timed (please refer to Section 5.7.2.5 - Part B). Another automated event is used to lower the computational power required by the simultaneous use of several software. In fact, the CataRT software’s random selection function requires a great deal of cpu resources. Thus, the function is enabled only at the start of Part B and disabled during section A. The start and stop commands for the CataRT patch are sent by the main Agorá patch via OSC messages (CNMAT 2012).

5.7.2.4 Part C

Part C is the short Coda section of the performance, and it includes automated events only. All gates in all patches are closed, preventing any kind of performer’s input to the software and its parameters. This is signaled to the performer by the automatic fading out of all ongoing sounds while new sounds are introduced. The new sounds are an ‘unfrozen’ version of the ongoing random snippets selection mechanism from the CataRT application that is now output to all the channels in the array of speakers via four additional Ableton audio tracks. An automated function pitch-shifts this output down by controlling the pitch shift parameter within the Cata-RT application. The dialogue between three voices, extracted from one of the author’s family footage, marks the end of the performance (see Section 5.7.2.5 - Part C and Figure 5.39).

5.7.2.5 Score Macro Structure - Length and Structure of the Whole Performance

The several automated elements of the performance are timed from a macro to a micro structure level. The macro level defines the length of the performance. Each section of the performance is using up a precise percentage of the overall length. While the length of the entire performance can be changed, the length of each section cannot because they are bounded to a precise percentage calculated with respect to the overall performance length value. The same proportional system is applied down to the single automated event within a section. This
5. PERFORMANCES

system was devised for two main reasons. Firstly, it offers a way of setting a precise length for the performance according to the needs of the venue hosting it and/or performer’s ‘feel’ for the occasion. Secondly, it allows for a fast method to test the automated events included in each section without the need for waiting a long time. In other words, if the performance had a fixed length of twenty minutes and we wanted to test an automated event at the eighteenth minute, we would have had to wait eighteen minutes. With the method devised, instead, we could set the length of the performance to be two minutes and wait one-hundred-eight seconds only to test that automated event.

Assuming the overall performance’s length to be one-hundred, each section takes up the following percentage:

- Part A: 40%
- Part B + Part C: 60% of which → 97% (Part B) and 3% (Part C)

Within each section, all the macro events are further divided with reference to the duration of the section they belong to. In summary we have:

- Part A = Duration Act (99%) + Pause after the ‘breaking glass’ soundfile is played (1%)
- Part B = Freeze Gesture (35%) + Hey Gesture (20%) + Resonator Gesture (45%)
- Part C = Automated events (3% of the Part B + Part C duration)

With regards to the three gesture in Part B, the system is designed so that when the automated line of events within each ‘gesture section’ reach its end it will enable the next ‘gesture section’. Thus, the percentage in this case refers to the length of the overall automation line event before triggering the next gesture. Figure 5.36 summarises the macro event structure here discussed.

The percentage for each section has been decided according to what the composer felt to be a balanced length. Within each section, each event is then timed using the same percentage system or through a series of delay functions whose time values are scaled with proportion to the section length.
Figure 5.36: *Agorá* - Macro Structure - Percentage allocated to each section and subsection (see ‘p performance-duration’ patch in Agorá.maxpat).

Figure 5.37: *Agorá* - Structure of Part A - Percentage allocated to each subsection in section Part A and automated events.
5. PERFORMANCES

Part A

At the beginning of Part A the force applied to both the ‘whoosh’ sound and the memory song list (mems) is automated\(^1\). The automated values are set as to create a sort of crossfade effect. This crossfade last 60% of the ‘Duration Act’ subsection. At the end of the ‘whoosh’ section, an automated line for the wet parameter of the reverb effect on the mems channel is triggered. The direction of value is from high to low (wet to dry) and it last for the entire duration of the mems section (Figure 5.37).

Part B

In Part B, the volume for each Ableton audio track receiving audio from the CataRT software (and going through the Freeze 1.0 plugin) is automated. The four tracks are introduced sequentially (channel 1 - 3 - 2 - 4). Shortly after all channels are active, all sounds are faded out. The length of this entire sequence of events is set to be 35% of the overall length of Part B and it has been named Freeze. At this point in time the performer can decide how long to wait before enabling the Hey section. This section is enabled as soon as the performer decides to trigger the first ‘hey’ sample. When this action is performed, all four Ableton tracks are faded in at once. The performer is now able to use gestures from subsections Freeze and Hey simultaneously. A delay line is set to make this section last for 20% of the overall Part B length point at which the Resonator subsection is enabled. The auditory cue given to the performer in order to let him know that the Resonator subsection has been enabled is given by the automatic entrance of sound loops containing confused voices extracted from a footage of a family party of the performer. This soundfile is faded in and out twice over the the length of the subsection set to be 45% of the overall Part B length. When the soundfile is faded out for the second time the performer knows that the section Part B is about to end and Part C about to start.

Part C

\(^1\)Please note that a change in force value does not result in sound output. The value refers only to the force that the performer will eventually exert on the spring module via the use of the data glove.
Figure 5.38: *Agorá* - Structure of Part B - Percentage allocated to each subsection in section Part B and automated events.

Figure 5.39: *Agorá* - Structure of Part C - Automated events in section Part C.
5. PERFORMANCES

Part C is completely automated. Three are the main automated functions. The first is a crossfade between the four Ableton tracks receiving the audio from CataRT and processed by the Freeze 1.0 plugin and four Ableton track receiving the CataRT signal and processed through a reverb unit in Ableton. The second function regards instead the pitch shift parameter in the CataRT unit which moves from an high value (50) to a low one (-20). This affect only the tracks connected to the reverb unit because the other tracks are in freeze mode. At the same time and for the same tracks, the wet reverb parameter moves from 127 to 20 (wet to dry). When the volume for the CataRT + reverb moves below a certain threshold (last part of its automation line), the final sound file, reproducing a short dialogue between three people at the table, is introduced (Figure 5.39). This ends the performance.

5.7.3 Discussion

The Agorá performance draws on the experiments from the performances earlier discussed. The theatrical element is one of these. The set is the inner mental space of the performer while the theme is represented by the memories that this space holds. More importantly, the reasoning and analysis conducted for the development of the previous performances now allow the composer to have a literature of performances, although small, to draw from. The properties of the glove have been explored, and, while this is without a doubt a potentially endless process, the composer knows the technical constraints offered by the device. In contrast, the data glove affordances have been found to be context dependent as new properties are crafted, associated and discovered for the device in use. In this way, it can be stated that the development of Agorá followed artistic needs first. The artistic needs here are to allow the audience to ‘enter’ the memory space of the performer. From this point of view, Agorá presents similarities with the Rex Levitates performance in which the dancer’s kinesphere was explored. However, in Agorá the performer’s kinesphere metaphorically represents his mental space bounded within the limit of the quadraphonic array of speakers. These boundaries are also highlighted by the choice of a dark stage scene, designed for the ‘UL
5.7 ‘AGORÁ’

Studio’ version of the performance that reflects the personal, somewhat sacred, if you will, space in which the audience is about to enter.

The entire gestural vocabulary presented in Agorá is built upon the metaphor and theme it represents. In Part A, the ‘dust’ metaphor is fundamental for the development of the mapping strategies devised. The visual reference to existing multimedia projects is also an element which facilitated the artistic imagination and the creation of appropriate mapping metaphors. The use of the *pmpd* allowed for the technical development of this metaphor, while also providing an example of three-layer mapping technique (see Section 2.1.5 ([Hunt and Wanderley 2003]).

The data from the glove is not directly mapped to the sound producing mechanism, but it is instead mediated by the *pmpd* physics emulator engine which allows for the implementation of the thought metaphor. More specifically, the volume here is not controlled by the direct input of some data from the glove but mediated by the physic emulator algorithm used which interprets the ‘wind’ forces acted upon the object (memory), freeing this from its dust. The volumes, controlled by the y-axis elevation values, controls the intensity of the force applied to the spring emulator. The resulting sound output, on the other hand, seems to follow the rising and falling of the hand giving coherency to the gesture, the mapping strategies and the sound output, while at the same time offering a high level of control intimacy from the performer perspective.

The spatialisation gestures enable the definition of the *meta-kinesphere’s* boundaries and collapse, similarly to the Rex Levitates performance, the *field* function into the *local* one. The function’s attributes of *real* and *imaginary* also seem to move from the former to the latter. Indeed, what at first seems to be directly controlled by the performer becomes, as the time passes by, an act in which the performer clearly controls sounds loops. The movement from the *real* to the *imaginary* is here enabled by the automated functions in place which change the way the performer interacts with the design algorithm (i.e. the automated changing

---

1. Please refer to Agorá UL Studio version in attached DVD.
2. The spaces provided for the performances held in Dublin and Limerick, did not have the required darkness. The sensation of intimacy was in this case stressed via the performer’s gesture of holding the right hand in his pocket signifying the stress perceived in allowing people to enter this personal space but at the same time ‘granting’ them the permission to do so.
of the force values for both the whoosh sound and the mems). This combination of automated systems and live interactions does not seem, however, to preclude to the understanding of the affordances of the data glove. The combination of spatialisation gestures and the ‘dust removal’ gesture informs and helps speculate about the possible mapping strategies in place. In particular, the pointing gestures reinforce the live act of delimiting the boundaries of the meta-kinesphere while the ‘dust removal’ gesture helps unveil the ‘objects’ (memories) located within the created space.

Part B presents similar characteristics. The meta-kinesphere is now delimited by the fixed-point sound sources (no real-time spatialisation is involved) given by the quadraphonic displacement of the array of speakers. This again collapses the field function into the local one. The boundaries between imaginary and real are instead less clear. The first section, ‘Freeze’, sees a random generator algorithm at the core of the mapping strategies, making it difficult to judge the allowed control of the performer over the sound-producing mechanism. The sounds are prepared in advance (analysed and imported as metadata into the CataRT software), and this would be a characteristic of an imaginary function. Furthermore, these sounds are initially introduced by an automated event that progressively fades them in. However, the real sources are somewhat hidden and heavily processed behind a mechanism that, to a certain extent, is controlled directly by the performer and thus belonging to the real function. As stated by Emmerson, though, ‘the distinction of truly real and imaginary lies with the composer and performer, not the listener’ (Emmerson 2007, p. 93). If so, the author of this dissertation is more inclined to talk of real functions due to the real-time processing, although admittedly not fully controlled by the performer, of the sound producing mechanism. In addition, the random generator unit is also seen as a metaphor for the memory trying to fool the performer, and over which he is constantly trying to gain control. Thus, the random generator unit is an inclusive rather than separate element of the sound producing mechanism. Conversely, in section ‘Hey’ the imaginary attribute of the function as the performer interacts with inserts points of pre-recorded sound files is clear. In section ‘Resonator’, instead, we have a direct manipulation of the sound producing mechanism and,
although limited to the control of the wet/dry effect’s parameter, we can still speak of *real* function.

The variety of mapping strategies experimented with in Part B are, however, in contrast with the complexity of the single one presented in Part A. This is reflected also in the level of control intimacy perceived by the performer which has decreased in Part B. This, however, can be counter balanced by the clarity of the cause/effect mechanisms presented in this section from an audience perspective. To a certain degree it can be stated that it was tried to contrast the quality (Part A) with the quantity (Part B) of the mapping strategies devised. In that regard, another observation concerns the way each new gesture is introduced in Part B. Indeed, all gestures (‘Freeze’, ‘Hey’ and ‘Resonator’) are introduced and enabled progressively throughout the section. This was done for two reasons. Firstly, it was thought that this method would have enabled the audience to engage in a sort of learning session in which the affordances of the devices could have been investigated. Each gesture has its own space of time to reveal its properties before the new gesture is introduced. This way a sort of gestural vocabulary is built dynamically and over time. Secondly, the progressive addition of new gestures to the vocabulary aimed at the creation of a climax for the performance which thus developed in complexity.

This complexity is also interpreted by the increasing number of automated score events added starting from the second half of Part B. We are referring to the pre-recorded voices extracted from one of the author’s family tape. The audio extracted from the tape is introduced repeatedly at precise intervals of time and contains confused and loud voices of both adults and children. The complexity here is given in two ways. On the one hand, the voices add to the soundscape that the performer is creating. On the other hand, these voices begin a process of re-evaluation of the performance’s *frames*. They belong to the Field function. Furthermore, since strong are the references to the real world (human voices), they are a *real field* function. Thus, if in Part A the *field* and *local* functions were a *continuum* confined in the Local territory, now they begin to separate. This *frames*’ movement continues in Part C where no performer interaction is allowed and thus now sees the Local function vanishing to leave space for the Field one only.
5. PERFORMANCES

In the metaphoric context of the performance, the action in Part B represents the moment in which subconscious memories start to emerge and mix with the conscious ones the performer is trying to gain control over. But, this is only the start of a process that culminates with Part C, in which the performer loses any control, and his conscious memories are wiped out by forces he cannot control. Forces that, however, allow for a personal re-interpretation of the same memories the performer has manipulated up to a moment before.

5.7.4 Lessons Learned

In summary, the Agorá performance highlighted the following issues:

- The use of theatrical elements helped define a frame in which to build metaphors that can then be used for the development of coherent mapping strategies.

- The creation of a theme offers a better frame over which building coherent mapping metaphors.

- The link between visual and audio movement is a strong one. The use of a physics emulator library helped transpose visual paradigms of interaction-reaction into sonic movements via coherent metaphors.

- The implementation of a three-layer mapping technique enhanced the control intimacy perceived by the performer while offering coherent metaphors that can enhance the understanding of the underlying cause/effect mechanisms from an audience perspective.

- The use of more direct mapping techniques presented the same issues of good clarity of the cause/effect mechanisms (when not applied to effects’ parameters) from an audience perspective and poor control intimacy from a performer perspective. However, the progressive introduction of several gestures offered the audience the possibility to create a gestural vocabulary on-the-fly.
5.8 Summary

- The integration of automated events which dynamically change the mapping strategies in place was found useful. However, its use needs to be validated within the context of the performance and the metaphors presented in it.

- The use of pre-recorded material does not preclude the perceived feel of controlling the sound producing mechanism if the mapping strategies in place include a combination of complex automated systems and live performer’s interactions.

- The simultaneous use of automated algorithms and live elements in the performance generated a great deal of movement and relocation of the frames. In turns, these frames helped define the overall structure of the performance and gave the needed coherency to both the mapping and the artistic metaphors in place.

5.8 Summary

This chapter presented the development of a series of live performances that employed the use of the Pointing-at dataglove. A detailed explanation of the mapping strategies in place was offered. The idea behind each performance and the way it has been developed was also presented. Each performance was then analysed and evaluated.
6

Conclusions and Future Directions

6.1 Summary

The main question addressed in this dissertation was how to develop, compose for and perform with a newly developed data glove, named Pointing-at, focusing on gestural performative actions which are both visually and sonically legible to audience members and that retain a high level of controllability (‘control intimacy’) from the performer perspective. The discussion was framed around some of the key issues surrounding the NIME community. These issues were introduced in Chapter Two. Chapter Three presented an historical survey of data gloves commercialised by the industry and data gloves developed by the NIME community. Four performances in which the use of different kind of data gloves were also analysed and discussed. A detailed description of the design process and the preliminary data analysis methods employed for the Pointing-at glove were offered in Chapter Four. Chapter Five discussed and analysed the development of a series of performances in which the Pointing-at glove was employed. The development of these performances helped to iteratively develop an understanding of how the main question of this dissertation can be addressed in the author’s performance practice. Each performance presented a series of issues that were discussed. The following section summarises the outcomes of the practice-led research.
6. CONCLUSIONS AND FUTURE DIRECTIONS

6.2 Conclusions

Each of the performances developed presented a series of issues. The main aim was the development of mapping strategies that were transparent in the cause/ef-
fect mechanisms between gesture and sound from an audience perspective while, at the same time, reinforcing the perceived control intimacy of the instrument from the performer perspective. The chronological order in which these perfor-
mances were developed played an important role; the failures and successes of earlier performances have, at times, informed the development of the next one. In other words, rather than beginning with an exclusively theoretical model, whose validity needed to be ‘tested’ in practice, here we have a mixture of theoretical concepts and practical elements that inform the development of each performance. This modus operandi enabled a series of discussions around the way the design of the performances was conducted. By drawing an overall analysis on the work done, it is possible to draw the lines for what can be described as the author’s approach to the design of nuanced gestural performative actions. The video docu-
mentary, titled ‘Documentary’, attached to this dissertation presents, in addition to an explanation of the mapping strategies in place, a brief analysis for each performance around the key elements discussed hereafter.

The common features that informed the performances’ post-analysis can be re-conducted within a few parameters. These are: a device’s affordances, techno-
logical and cultural constraints, control intimacy, mapping, sound design, local and field function and frames.

Affordances and Constraints

The initial, and probably the most important, step that allowed for a greater sophistication of the mapping strategies was the numerical evaluation of the data retrieved from the sensor cluster. This allowed the author to overcome the tech-
nical constraints found in the ‘Preliminary Works and Demo’, where the data was treated in its raw form (i.e. ADC values). Indeed, it was found that the combination of the data from the cluster of sensors enabled the evaluation of the orientation of the object with respect to the Earth fixed frame of reference. This knowledge immediately widened the range of possibilities offered to the composer.
The data had indeed a precise meaning which, in our case, was the attitude of the object hosting the sensor (i.e., the hand). This allowed us to translate the attitude of the hand to its pointing direction. Furthermore, the numerical evaluation and the consequent mapping strategies enabled us to state that the data glove affords ‘pointing’. In addition, the bending sensor for the index finger was used as a switch for which the binary states were given by the extending and flexing of the finger. When this gesture purposely, or theatrically, involved the whole hand\footnote{Although no sensors were mounted on the other fingers.} we can state that the device afforded ‘grabbing’ (hand-closing) and ‘throwing’ (hand-opening). The affordance attributes of the device were exploited in a different manner across the five performances presented. The list below summarises the investigated affordances for each performance:

- **3D-Dj**: pointing, grabbing, throwing.
- **Rex Levitates**: pointing, grabbing, dancing.
- **Molitva**: pointing, grabbing, throwing.
- **Mani**: pointing, grabbing, throwing, silencing, handshaking.
- **Agorá**: pointing, grabbing, throwing, dust removing.

The affordances of ‘dancing’, ‘handshaking’ and ‘silencing’ presented in the *Rex Levitates* and *Mani* performances, and in relation to the glove device, are expanding the interpretation of ‘affordance’, as presented by Gibson, Chemero and Norman. Indeed, stating that the glove ‘affords’ dancing or handshaking could sound unusual. However, it is also true that in the composer/performer’s intentions and movements (and likely in the eyes of the audience) that is exactly what the Poining-at glove is and what it does. The difference is that the performance scenario presents peculiar characteristics that are only partially considered in the Gibson/Chemero and Norman explanations of affordance. Of particular relevance here is the metaphoric nature of the live performance, for which the interaction between object and subject is dealt through metaphor and reference rather than directly and pragmatically. These affordances became explicit only in
6. CONCLUSIONS AND FUTURE DIRECTIONS

a precise performance context as, for example, for the ‘silencing’ in *Mani*. It was by introducing the theatrical elements into the performance that we were able to extend the palette of gestures and affordances. In this way, the gestures, rich of both communicative attributes and metaphors, facilitate the description of the device’s affordances from an audience perspective. This would eventually facilitate the comprehension of the cause/effect (gesture/sound) mechanisms, too. If that is the case, we can now state that the glove affords ‘spatialising’ as a result of the pointing gesture (e.g. 3D-Dj, Agorá) or ‘triggering loops’ as a result of the grabbing and throwing ones. Thus, the difference resides in that the ‘pointing’ and ‘grabbing’ affordances relates to the theatrical context of the performance, while the ‘spatialising’ and ‘triggering’ loops are specific to the sound producing mechanism.

The technical constraints offered by the data available and its numerical evaluation was another element that helped the development of the performances. Knowing the limitations of the digital instrument allowed for a better exploitation of the data available. In other words, it helped thinking what more can be done with the attitude and switch mechanism (bending sensor) data. The performances developed are an example of how this approach can be fruitful.

On the other hand, the cultural constraints have been used here to connect the performers with their audience with elements that go beyond the use of the newly developed digital instrument. Indeed, the Pointing-at glove is a new instrument of unknown properties from an audience perspective (there is a lack of ecological knowledge). The introduction of a storyboard (as in *Mani*), a theme (*Agorá*), and a multidisciplinary art environment (*Rex Levitates*) served then the purpose of offering other elements for the audience to latch on to, at least initially.

*Mapping Metaphors*

Mapping gesture to sound is a key issue in the development of live performances that employ digital instruments. The main problem is two-fold. On the one hand, it is important to find mappings for which the cause/effect links between gesture and sound are evident to an audience. On the other hand, the mappings aim at increasing the perceived control intimacy from the performer perspective.
In the performances presented, a series of mapping techniques have been implemented. One of the outcomes from the discussion of the first few performances put an accent on the idea that the degree of transparency of the mapping was inversely proportional to the level of perceived control intimacy. This was due to the fact that the performer’s control over the music material was limited in many occasions to the triggering and spatialising of pre-recorded loops as, for example, in the first version of the 3D-Dj application. At the same time, these gestures were characterized by strong communicative attributes (pointing, throwing, etc.) and, thus, their cause/effect link were made more evident. Conversely, the control of audio effect parameters was offering a greater degree of control to the performer, but it was highly enigmatic to an audience, as no consistent relationship between the gesture and related parameter could be found. However, the establishment of appropriate metaphors in relation to the theme of the performance helped develop mappings in which both the transparency and control intimacy attributes could be increased. A first example of this was given in section Speech A of Mani. Here, the hand’s gestures controlled the partials’ amplitudes of an additive synthesizer. The interaction was made via pointing gestures mapped to continuous control data. This offered the performer a possibility to control nuances in the sound producing mechanism and, thus, it increased the perceived control intimacy. On the other hand, the pointing gestures were also used in the context of the performance and its theme. In particular, the pointing gestures were accusatory gestures. The communicative attribute of the gesture helps define an area of interaction. The audience may not fully understand the sound generating mechanism behind the gesture, but, at least, it offered an important element to hold on to. The repetition of the gesture helped then define the several regions and their corresponding sounds within the area of interaction. To a certain extent, it can be said that the pointing direction of the hand controlled a graphical synthesizer mapped around, and in the direction of, the audience space.

This approach to devising gesture to sound mappings is also presented in Part A of Agorá. The theme of the performance allowed for the establishment of a coherent metaphor between the gesture performed and the sound output created. The ‘pointing’ gesture became a ‘dust removing gesture’ defining a
space, metaphorically representing the memories of the performer and physically defined by the quadraphonic array of speakers. The perceived control intimacy from the performer’s perspective, is enabled, instead, by the underlying algorithm generating the sounds. In particular, the implementation of the physics emulator library helped to provide a more natural and, thus, rewarding connection between gesture performed and sound output. Also, the design of the mapping followed a precise idea of movement, which coherently related to the metaphor in place (the wind generated by the hand removing the dust).

Thus, mapping metaphors that are coherent to the sound producing mechanism and the theatrical context in which they operate seem to be the key aspect. Indeed, metaphors have been found to be helpful to both the performer/composer and an audience. With them, the performer has a tool to devise coherent mechanisms of interaction within the theme of the performance. On the other side, the audience will be able to interpret gestures within the context of the performance first. Then, if coherent mapping strategies are in place, the chances to understand the cause/effect mechanisms between the gesture seen and the sound heard will increase. As a result, the audience will be in the position to state that the device affords ‘something’.

Sound Design

The choice of sounds in each performance was, on most occasions, limited to the selection of pre-recorded soundfiles. However, this selection was not random. The selection was conducted according to the needs presented in each performance. A common thought was that the sounds needed to suit the theme of the performance. Also, all sounds needed to suit each other, especially if overlaying was envisaged. In Molitva, sounds were chosen in order to offer a contour to the soundscape created by the vocalist and the laptop performer. In addition, it was also thought to select sounds with timbral characteristics different from the sound controlled by the other two performers in order to clearly mark roles in the creation of the soundscape and, at the same time, enable the audience to recognise which performer was responsible for each sound generated.

In Mani and Agorá, the sound material was chosen according to the theme of the performance. In Mani the sounds are taken from excerpts of radio news
in which social issues are discussed. This was because they needed to reflect the theatrical role of the Pointing-at performer within the context of the performance. The low rumble sound in Speech A together with the additive synthesizer aim to stress the accusatory gesture enacted. The noise sound loops for the handshaking were selected in order to create tension. In *Agorá*, the sound sources are also selected according to the theme of the performance and, thus, are restricted to a selection of forty songs representing the musical memories of the performer. Thus, sounds are selected, generated and manipulated in order to, once more, highlight the metaphors of the performance.

**Functions and Frames**

The modalities with which these sounds are introduced in the performance allowed for the dynamic re-arrangement of the local and field functions and frames (Emmerson 2007). In the 3D-Dj demo and *Rex Levitates* performance, the space of action was limited to the local imaginary function, thus, collapsing the field frames onto the local one.

In *Molitva*, instead, the space of action for the Pointing-at performer, as well as for the vocalist, is, again, limited to the local imaginary function. However, the Laptop performer seemed to fill the field function. Frames were displaced here in a more traditional format which sees a stage and an audience in front of it. This dichotomy exists despite the use of a surround system extending to the audience area.

The static arrangement of functions and frames started to become dynamic in the *Mani* performance. In particular, the arrangement of the two performers’ interventions caused a change in the quantity of activity between the real and imaginary local function and the real and imaginary field function. Section ‘speech A’ has seen the activity of the Pointing-at performer moving from the local real (control of the additive synthesiser) to the local imaginary (spatialisation of pre-recorded loops). The absence of visible live interactions by the laptop performer in section ‘speech B’, locates the space of activity in the Field function and frames. The intervention of both performers in the ‘Overlap’ section creates activity in both the local and field function and frames. In the ‘Finale’, the initial secrecy of the gesture presented located the action in the field frame. However,
when the handshaking is interrupted, thus revealing the cause/effect mechanism behind the gesture and sound, there is an abrupt change which sees the Field frame collapsing into the local one.

Agorá, presents similar changes in quantity of activity between function and frames. Indeed, the space of action in Part A is limited to the local frame while moving from a local real to a local imaginary function. This was achieved by introducing automated functions in the mapping process, which changed dynamically the response to the performer’s gesture. Thus, the sounds, that in the beginning seemed to be generated by a synthesiser, slowly reveal their original sources (pre-recorded soundfiles). In Part B, the automated events are devised in order to create a Field function and frame. The performer’s local frame is now surrounded by sonic elements which are introduced independently from his interaction. In addition, these sonic elements represent more of the performer’s memories, and, thus, they have been selected according to the metaphors in place (voices from a family tape). Lastly, Part C, being completely automated, sees the local function and frames disappearing, leaving the entire space of action in the real field function and frame.

The inclusion of both live and automated events proved to be a valid approach for the development of the live performance. The use of Emmerson’s terminology of function and frames also helped to better described the mechanisms behind its design. Thus, they represent an important tool for both the development and analysis of the performances. It is important to note that function and frames were developed with in mind the metaphoric content of the performance. Their use, static or dynamic, can, therefore, represent a further tool for defining and highlighting the metaphors and the theme of the performance and validate the gestural interactions devised.

In light of these considerations, it is the reasoning around the concepts of affordances, constraints, mapping metaphors, sound design, functions and frames that has helped defining a strategy to the design of performative actions that enables a good level of controllability of the sound generating mechanism for
the performer and that are both visually and sonically legible from an audience perspective. In particular, it was found that the concept of metaphor is central to the establishment of coherent choices for the development of all the elements forming the performance. The establishment of coherent metaphors, within the theme of the performance, has helped devise sophisticated, while also readable, gesture to sound mappings. Also, it influenced the selection of the sound sources and whether these needed to be controlled live or via an automated chain of events.

The design of appropriate metaphors required the interpretation of the live music performance within the broader context of performance art. This allowed for the introduction of theatrical elements which, in turn, justified the use of the metaphors. In that regard, the performance becomes more than a music performance. Indeed, it can even become a multidisciplinary performance in which a combination of music, visuals, lights and robots could eventually be controlled by the performer. Of course, this approach would extend to the development of automated events, too. However, the creation of a cohesive literature, from which different composers and performers can draw, is fundamental. It is hoped that both the practical and theoretical outcomes of this research will inform and inspire the work of other performers who intend to make use of newly developed digital instruments and, thus, contribute to the making of a wider and coherent performance literature.

6.3 Contributions

In support of the critical reflections discussed above, an audio-visual documentary, attached to this dissertation, was developed. This documentary, alongside the authors critical reflections, is the primary research contributions.

In addition, the research required the development of a series of tool that enabled to conduct the investigative work: the Pointing-at data glove and the AHRS MaxMSP Library. Their development was documented in this dissertation and in the CD and DVD provided. While this documentation is not to be considered as a research contributions in its own right it helped however the author (and it is hoped to help others) in the research process.
6. CONCLUSIONS AND FUTURE DIRECTIONS

6.4 Future Work

The work discussed in this dissertation sets the basis for further developments in the field. Indeed, the performances presented here form a detailed literature on which other composers interested in developing more performances for the Pointing-at glove (or data gloves based on similar technology) can latch on to. The use of the proposed approach to the design of nuanced performative actions may not be considered, as it may be thought of as too subjective. However, the existence of a literature opens up the discussion and a reference for comparison. If the approach is followed, it is, then, interesting to see how it will be interpreted by other composers.

The successful implementation of mapping strategies, for which a high level of control intimacy on the performer’s perspective and a transparent cause/effect link from the audience one (such as the accusatory gesture in Mani and the dust removing gesture in Agorá), is found here to be strictly related to the finding of coherent metaphors within the theatrical context of the performance. More experimentation with the concept of metaphor is, therefore, sought. Only the building of a vast and detailed performance literature developed by many composers can eventually validate an approach and/or set the basis for an artistic trend. This can be achieved with the use of other digital instruments, too, and not necessarily with the use of the Pointing-at glove. An option that seems, in the opinion of the author, to better suit the nature of the NIME community.
Appendix A

WIMU Specifications

The gyroscopes have a default measurement range of 150 °/s but this can be increased up to 600 °/s if required. The accelerometers have a measurement range of +/- 2g with the magnetometers being specified with a measurement range of +/- 6 gauss. In terms of ADC steps:

- The AD7490 ADC ([Analog Devices](https://www.analog.com/en.html) 2012a) is powered by a 5V supply and has a resolution of 12 bits, which provides a voltage step of 1.22mV. The ADC inputs are offset around 2.5 V, which means a zero voltage will read as 2048. The maximum positive voltage it can read is 2.5V.

- The ADXRS150 gyroscopes ([Analog Devices](https://www.analog.com/en.html) 2012c) range has been modified to 406 °/s and have a resolution of 4.5mV / °/s. A single ADC step increment corresponds to 0.27 °/s rate of turn. A rate of turn of 406/s will produce a 1.84V output which is well within the 2.5V limit.

- The ADXL202 accelerometers ([Analog Devices](https://www.analog.com/en.html) 2012b) resolution has been recorded as 600mV / g. A single ADC step increment corresponds to a 2mg acceleration (39.85 mm/s). The maximum acceleration it can register corresponds to a 1.2 V signal, which is well within the 2.5V ADC limit.

- The HMC1052L magnetometer ([Honeywell](https://www.honeywell.com/en.html) 2012) resolution has been registered as 385mV/gauss. A single ADC step increment corresponds to 317mG (317nT). The maximum magnetic field that the sensor can register is 6gauss, which corresponds to a voltage of 2.31, which is well within the ADC maximum limit.
Figure 1: 25mm WIMU - FPGA Schematics.
Figure 2: 25mm WIMU - Sensor Interface Schematics.
Figure 3: 25mm - Printed Circuit Board.
Appendix B

TRIAD Algorithm: a numerical example

The following numerical example take into consideration a cluster of sensors made up of a three-axis accelerometer and a three-axis magnetometer.

- Accelerometers reading vector
  \( \hat{A}_{ba}\{−0.2350, 0.0030, 0.8389\} \)
- Magnetometers reading vector
  \( \hat{M}_{bm}\{0.1021, 0.1428, −0.5705\} \)
- Gravity vector
  \( \hat{A}_{ia}\{0, 0, 1\} \)
- Magnetic pole vector
  \( \hat{M}_{im}\{−0.0837, 0.5959, 0.7986\} \)

Our first step is to normalise each given vector by dividing each vector element by:

\[ \frac{1}{\sqrt{x^2 + y^2 + z^2}} \]

which gives:

\( \hat{A}_{ba}\{−0.2698, 0.0034, 0.9629\} \)
\( \hat{M}_{bm}\{0.1711, 0.2393, −0.9557\} \)
\( \hat{A}_{ia}\{0, 0, 1\} \)
\( \hat{M}_{im}\{−0.0837, 0.5960, 0.7987\} \)
Appendix

Define the new body frame $R^b\{\hat{t}_{1b}, \hat{t}_{2b}, \hat{t}_{3b}\}$ as:

\[
\hat{t}_{1b} = \hat{A}_{ba} = (-0.2698, 0.0034, 0.9629) \quad (1)
\]

\[
\hat{t}_{2b} = \frac{\hat{A}_{ba} \times \hat{M}_{bm}}{|\hat{A}_{ba} \times \hat{M}_{bm}|} \quad (2)
\]

\[
\begin{align*}
\hat{t}_{2b} = & \frac{(y_{ba}z_{bm} - y_{bm}z_{ba}, z_{ba}x_{bm} - z_{bm}x_{ba}, x_{ba}y_{bm} - x_{bm}y_{ba})}{|A_{ba}||M_{bm}|\sqrt{1 - (\hat{A}_{ba} \cdot \hat{M}_{bm})^2}} \\
& = (-0.2338, -0.0931, -0.0651) \\
& = (0.2601)
\end{align*}
\]

which if normalised gives:

\[
\hat{t}_{3b} = \hat{t}_{1b} \times \hat{t}_{2b} \quad (3)
\]

(normalised resulting vector)

\[
\hat{t}_{3b} = (0.3439, -0.9337, 0.0997)
\]

which defines our new body frame as:

\[
R^b = [\hat{t}_{1b}, \hat{t}_{2b}, \hat{t}_{3b}] = \begin{pmatrix}
-0.2698 & -0.8995 & 0.3439 \\
0.0035 & -0.3581 & -0.9337 \\
0.9629 & -0.2504 & 0.0997
\end{pmatrix}
\]

and, using equations (4), (5) and (6) \footnote{1The calculated vectors values that are given are normalized.} a new reference frame $R^i\{\hat{t}_{1i}, \hat{t}_{2i}, \hat{t}_{3i}\}$ as:

\[
\hat{t}_{1i} = \hat{A}_{ia} = (0, 0, 1) \quad (4)
\]

\[
\hat{t}_{2i} = (-0.9905, -0.1391, 0) \quad (5)
\]
\[
\hat{t}_{3i} = (0.1391, -0.9905, 0)
\]  
which defines our new body frame as:

\[
R^i = [\hat{t}_{1i}, \hat{t}_{2i}, \hat{t}_{3i}] = \begin{pmatrix}
0 & 0 & 1 \\
-0.9905 & -0.1391 & 0 \\
0.1391 & -0.9905 & 0
\end{pmatrix}
\]

Finally, we can calculate the object’s attitude \(R^a\) as:

\[
R^a = R^b(R^i)^T = [\hat{t}_{1b}, \hat{t}_{2b}, \hat{t}_{3b}][\hat{t}_{1i}, \hat{t}_{2i}, \hat{t}_{3i}]^T
\]

where \(T\) denotes transposition therefore:

\[
R^a = 
\begin{pmatrix}
0.9387 & -0.2155 & -0.2698 \\
0.2248 & 0.9746 & 0.0035 \\
0.2618 & -0.0639 & 0.9629
\end{pmatrix}
\]

The TRIAD algorithm ends here.

However, we need a further step if we want to have the virtual object in our Jitter patch moving in the same direction as in the real 3D space. This can be easily achieved by simply calculating the Inverse of the calculated matrix (normalization is advisable):

\[
R^{a(-1)} = \begin{pmatrix}
0.9384 & 0.2247 & 0.2621 \\
-0.2155 & 0.9742 & -0.0639 \\
-0.2694 & 0.0036 & 0.9630
\end{pmatrix}
\]

The resulting matrix can then be converted in any desired format such as Quaternion, Angle-Axis Euler or Cartesian.
Appendix C

Driver code explanation.

(Internal report by Shirley [2006]). The data from the master node consists of a stream of packets, one per node that the master is aware of. The processing of this stream is handled by the cel_ser library. The packet format starts with a node number in ascii and the ':' character, and finishes with '\n\r'. As the request for data from the applications using cel_ser is not synchronous with the data stream, the library tries to be intelligent about how it handles requests. Another consideration to be taken into account is that not all nodes may be present at any given time, and we want to avoid having to reconfigure the library every time the list of connected nodes changes. Bearing all these in mind, the rest of this section describes the algorithm used in cel_ser. cel_ser is multi-threaded; the serialio thread handles reading from the serial port and processing the data, the main thread handles the passing of data back to the application that calls cel_ser_read(). Whenever the application calls cel_ser_read(), it attempts to lock the new_data_mut mutex. This mutex is only unlocked by the serialio thread whenever a full cycle has been completed, ensuring that the application never gets old or partial data. The serialio thread loops constantly, calling cel_ser_read_real(). This function starts off by setting all the node data buffers to 0xFF, so that if a node doesn’t send data, the application will get all -1’s for that node’s sensors. This makes it easy to notice and deal with a node going offline from the application level. If the library is out of sync with the data-stream (happens on start-up, or if the stream became temporarily corrupted), it will begin searching the data-stream for a valid packet. This is done by reading an amount of data equal to twice the packet length. That data is then searched for a sequence of bytes the same length
Appendix

as a packet, that start with a byte between ‘1’ and ‘8’, followed by ‘:’, and has \n\r as the last two bytes. If this is found, then that sequence is a valid packet. Further checks could be done, as each of the sensor bytes have the sensor number encoded into the top four bits - but experience showed that this was ‘overkill’. Once a valid packet has been found, it is processed (i.e. the node number and various sensor readings are extracted) and the result stored in the appropriate node data buffer. cel_ser_read_real() then starts reading in one packet lengths worth of data and processing it, and continues doing so until it encounters an invalid packet (in which case the stream is corrupted and a re-sync needs to be attempted), data for all nodes has been collected, or a duplicate node number is detected. In the latter case, any nodes that havent shown up in the data stream so far are assumed to be missing/offline as the master node cycles through all the nodes sequentially. When cel_ser_read_real() has reached one of those terminating conditions, it unlocks the new_data_mut mutex allowing cel_ser_read() to read the node data buffers and return that data to the calling application.
Appendix D: Mote External and Driver

MOTE.C

/* "mote" Object for Max/Msp 4.6 and 5
   author: Giuseppe Torre - University of Limerick - Ireland
date: September 2012 */
#include "ext.h"
#include "ext_mess.h"
#include <string.h>
#include "celser.h"

void *this_class; // Required. Global pointing to this class
typedef struct mote // Data structure for this object
{
    t_object m_obj; // Must always be the first field: used by Max
    long m_value; // Last value in left inlet
    void *m_out; // listout
    void *m_out1; // bang middle outlet
    void *m_latency; // gap between new package leftmost outlet
} t_mote;

// Prototypes for methods: need a method for each incoming message
void *mote_new(long value); // object creation method
void mote_bang(t_mote *bang); // method for bang message
void mote_int(t_mote *mote, long value); // method for integer
void mote_assist(t_mote *mote, void *b, long msg, long arg, char *s);
void mote_free(t_mote *mote);

int main(void)
{
    // set up our class: create a class definition
    setup((t_messlist**) &this_class, (method)mote_new, (method)mote_free, (short)sizeof(t_mote), 0L,A_GIMME, 0);
    addbang((method)mote_bang); // bind method "bang_bang" to the "bang" message
    addint((method)mote_int);
    addmess((method)mote_assist, "assist", A_CANT, 0);
    finder_addclass("Devices", "mote");
    post("...I'mMOTEObject!...for 4.6...",0);
}
return 0;
}

/* ------------------------------ */
/* ------------------------------ */
void *mote_new(long value)
{
    t_mote *mote;
    mote = (t_mote *)newobject(this_class); // create the new instance and return a
    mote->m_latency = floatout(mote);
    mote->m_out1 = bangout(mote);    // create a bang outlet
    mote->m_out = intout(mote);

    if(!celser_setup()){
        post("Failed to connect to serial port", 0);
    } else {
        post("Connected to serial port", 0);
    }
    return(mote); // must return a pointer to the new instance
}
/* ------------------------------ */
void mote_bang(t_mote *mote)
{
    outlet_bang(mote->m_out1);    // send a bang to the outlet bang->m_out (leftmost
    outlet)
}
/* ------------------------------ */
void mote_int(t_mote *mote, long value)
{
    static int running = 0;
    static time_t last = 0, current = 0;
    static int count = 0;
    time(&current);
    if(current == last)
        count++;
    else {
        post("Called \$\{times\per second\", count, 1);
        count = 0;
    }
    last = current;
#endif
    running = value;

    if (running == 1) {
        data = celser_read(&diff);
        if(data == NULL) {
            post("Error reading from serial port", 0);
            return;
        }
        for(i=0; i<DATA_SIZE; i++) {
SETLONG(& mylist[i], data[i]);
}

outlet_list((mote->m_out), 0L, DATA_SIZE, mylist);
outlet_float((mote->m_latency), diff/1000000);
mote_bang(mote);
printf("%d", data[i]);
} else {
    post("I need '1' responding to a metro",0);
}

 Pertinent code

*/ −−−−−−−− mote assist −−−−−−−−∗

void mote_assist(t_mote *mote, void *b, long msg, long arg, char **s)
{
    if (msg == ASSIST_OUTLET)
    {
        switch (arg)
        {
        case 0: sprintf(s, "%s", "msgs coming from mote device are output as list");
            break;
        case 1: sprintf(s, "%s", "bang output");
            break;
        case 2: sprintf(s, "%s", "gap between data packages");
            break;
        }
    } else if (msg == ASSIST_INLET)
        sprintf(s, "%s", "I need '1' to start and '0' to stop");
}

 Pertinent code

*/ −−−−−−−− mote_free −−−−−−−−∗

void mote_free(t_mote *mote)
{
    // if (!cel->err_shutdown())
    //    post("serial port NOT Disconnected!", 0);
    //} else{
    //    post("Fine!...BYE!", 0);
    //}
Appendix

CEL_SER.C

#include "cel_ser.h" /* First for __unix__ and __windows__ definitions */
#include <stdio.h>
#include <errno.h>
#include <fcntl.h>
#include <string.h>
    // #include <sys/time.h>
#include <time.h>

#if defined(__unix__)
#include <termios.h>
#include <unistd.h>
#elif defined(__windows__)
#include <windows.h>
#include <process.h>
#endif

#include "cel_ser_util.h"

#if USE_SERIAL == 1
static int cel_ser_findstart(unsigned char buf[]);
#endif

#if defined(__unix__) && USE_SERIAL == 1
static struct termios term_before;
#endif

#if !defined(LOG_OUTPUT)
static THREAD_HANDLE(serial_io);
static THREAD_FUNCTION(cel_ser_loop);
#endif

short data_done[DATA_SIZE]; // Data that has been fully received.
long timediff = 0;
static int thread_loop;
static int started = 0;

FD_DECLARE(fd);

int cel_ser_setup()
{
    int ret;
    #if USE_SERIAL == 1
    #elif defined(__unix__)
    struct termios term;
    #elif defined(__windows__)
    DCB dcb;
    #endif
    #endif /* USE_SERIAL */
    if(started > 0) {
        print_output(1, "cel_ser_setup: Setup already done.");
        return 0;
    }

    #if USE_SERIAL == 1
    #if defined(__unix__)
    // Open serial port
    fd = open(SER_DEVICE, O_RDWR | O_NOCTTY);
    if (fd < 0) {
        print_output(1, "cel_ser_setup: Failed to open device " SER_DEVICE "");
        perror("cel_ser_setup");
        return 0;
    }
    #endif
    #endif
}
Appendix

// Read in current serial port settings
ret = tcgetattr(fd, &term);
if (ret < 0) {
    printf("cell_ser_setup: Failed to get status of %s, %s", SER_DEVICE, strerror(errno));
    cel_ser_shutdown();
    return 0;
}

// Backup current serial port settings
memcpy(&term_before, &term, sizeof(struct termios));

// Clear termios structure
cfmakeraw(&term);

// Set serial port speed
ret = cfsetspeed(&term, SER_SPEED);
if (ret < 0) {
    printf("cell_ser_setup: Failed to set speed of %s to %d: %s", SER_DEVICE, SER_SPEED, strerror(errno));
    cel_ser_shutdown();
    return 0;
}

// Set data size
term.c_cflag &= ~CSIZE;
term.c_cflag |= CS8;

// Set no parity
term.c_cflag &= ~PARENB;

// Set stop bits
term.c_cflag &= ~CSTOPB;

// Set no software flow control
term.c_iflag &= ~(IXON | IXOFF | IXANY);

// Set no hardware flow control
term.c_iflag &= ~(CRTSCTS);

// Set raw mode input mode
term.c_iflag |= ~(ICANON | ECHO | ECHOE | ISIG);

// Set raw output mode
term.c_oflag &= ~(OPOST);

// Set to read mode
term.c_cflag |= CLOCAL | CREAD;

// Write out new serial port settings
ret = tcsetattr(fd, TCSANOW, &term);
if (ret < 0) {
    printf("cell_ser_setup: Failed to write out new settings to %s", SER_DEVICE, strerror(errno));
    cel_ser_shutdown();
    return 0;
}

#endif defined(__windows__)
fd = CreateFile(SER_DEVICE,
    GENERIC_READ | GENERIC_WRITE,
    0,
    0,
    OPEN_EXISTING,
    FILE_ATTRIBUTE_NORMAL,
    0);
if (fd == INVALID_HANDLE_VALUE) {
    printf("cell_ser_setup: Failed to open %s: %d", SER_DEVICE, GetLastError());
}
return 0;
}

FillMemory(&dcb, sizeof(dcb), 0);
dcb.DCBlength = sizeof(dcb);
BuildCommDCB(SER_SPEED, &dcb);
ret = SetCommState(fd, &dcb);
if (!ret) {
    print(output(1, "cel_ser_setup:Failed to write new settings to \"%s\", SERDEVICE, GetLastErr());
    cel_ser.shutdown();
    return 0;
}
#endif
#endif /* USE_SERIAL */

thread_loop = 1;
#if defined(__unix__) && !defined(LOG_OUTPUT)
    ret = pthread_create(&serialio, NULL, cel_ser_loop, NULL); //FIXME: cleanup in shutdown()
    if (ret != 0) {
        print(output(1, "cel_ser_setup:Failed to create serial I/O thread: \"%s\", strerror(errno));
        cel_ser.shutdown();
        return 0;
    }
    ret = pthread_mutex_init(&data_mut, NULL); //FIXME: cleanup in shutdown()
    if (ret != 0) {
        print(output(1, "cel_ser_setup:Failed to create serial I/O mutex: \"%s\", strerror(errno));
        cel_ser.shutdown();
        return 0;
    }
    ret = pthread_mutex_init(&new_data_mut, NULL); //FIXME: cleanup in shutdown()
    if (ret != 0) {
        print(output(1, "cel_ser_setup:Failed to create new_data_mutex: \"%s\", strerror(errno));
        cel_ser.shutdown();
        return 0;
    }
#endif
    started++;
    return 1;
}

int cel_ser_shutdown()
{
    int ret;
    thread_loop = 0;
#if defined(__unix__) && !defined(LOG_OUTPUT)
    ret = pthread_join(serialio, NULL);
    if (ret < 0) {
        print(output(1, "cel_ser_shutdown:Failed to join serial I/O thread: \"%s\", strerror(errno));
    }
#endif
    return WaitForSingleObject(serialio, INFINITE);
}
#ifdef USE_SERIAL == 1
#else defined(__unix__)
    if(fd > 0) {
        // Write old serial port settings
        ret = tcsetattr(fd, TCSANOW, &term_before);
        if(ret < 0) {
            print_output(1, "cel_ser_shutdown : Failed_to_write_out_old_settings_to_%s : %s", SER_DEVICE, strerror(errno));
        }
        // Close serial port
        ret = close(fd);
        if(ret < 0) {
            print_output(1, "cel_ser_shutdown : Error_closing_device_%s : %s", SER_DEVICE, strerror(errno));
            return 0;
        }
        fd = -1;
    } else {
        CloseHandle(fd);
    }
#endif
#endif /* USE_SERIAL */
/
*/ USE_SERIAL */
#endif

#endif

static THREAD_FUNCTION(cel_ser_loop) {
    print_output(1, "Serial_i/o loop running");
    while(thread_loop) {
        cel_ser_read_real();
        //SLEEP(10); //DEBUGFIXME: this shouldn't be needed
    }
    print_output(1, "Serial_i/o loop exiting");
    THREAD_FUNCTION_RET;
}
#endif

short * cel_ser_read() //long * t
{
    static short data_out[DATA_SIZE]; // Data to be passed back to max
    MUTEXLOCK(new_data_mut);
    MUTEXLOCK(data_mut);
    //t = timediff;
    memcpy(data_out, data_done, sizeof(data_done));
    MUTEXUNLOCK(data_mut);
    //print_output(0, "%d"); //DEBUG
    return data_out;
}

void cel_ser_read_real()
{
    int i;
    static short data_in[DATA_SIZE]; // Data that is being read in
    #if USE_SERIAL == 1
        static int started = 0;
        int done = 0;
        int j, ret, read = READ_IN;
        int first_node = -1;
        unsigned char buf[PKT_LEN + 2];
        //static struct timeval fast, current;
        //struct timeval between;
    #endif /* USE_SERIAL */
    #if USE_SERIAL == 1
        #endif
    }
Appendix

```c
if (fd < 0)
    return; //FIXME: handle error better
#else
    SLEEP(10);
#endif /* USE_SERIAL */

// Reset the data
memset(data_in, 0xFF, sizeof(data_done));
for (i = 0; i < NUM_NODES; i++)
    data_in[i * NODE_DATA_SIZE] = i + 1;
//memcpy(blast, current, sizeof(struct timeval));

#if USE_SERIAL == 1
while (!done)
{
    if (!started)
    {
        ret = cel_ser_findstart(buf);
        if (!ret)
        {
            print_output(1, "findstart Returned in error");
            return; //FIXME: handle error better
        }
    }
    else
    {
        short∗ cur;
        int cur_node;
        if (i == 0)
            first_node = buf[0] - '1';
        cur_node = (first_node + i) % NUM_NODES;
        cur = &data_in[cur_node * NODE_DATA_SIZE];
        if (cur_node != (buf[0] - '1'))
            //print_output(1, "Packet missing (expected %d, got %c)", cur_node + 1, buf[0]);
            read = READ_GOT_ONE;
        else
        {
            for (j = 0; j < NUM_SENSORS; j++)
                // Store the values, masking off the top 4 bits
                cur[j + 1] = ((short)(buf[2 + (j * 2)] & 0xFF) << 8) | (short)buf[2 + (j * 2) + 1];
                //for (j = 0; j < NUM_MAGNOMETERS; j++)
                // Store the magnometer values
                //cur[j + NUM_SENSORS + 1] = buf[2 + (NUM_SENSORS + 2) + j];
            continue;
        }
    }
}
#endif
```

192
done = 1;
}  
#endif /* USE_SERIAL */
//#ifdef USE_SERIAL
//gettimeofday(&current, NULL);
//timersub(&current, &last, &between);
timediff = (between.tv_sec * 1000000) + between.tv_usec;
MUTEXLOCK(data_mut);
memcpy(data_done, data_in, sizeof(data_done));
MUTEXUNLOCK(data_mut);
MUTEXUNLOCK(new_data_mut);
}
#endif /* USE_SERIAL */

static int cel_ser_findstart(unsigned char buf[])
{
    int i, offset = -1;

    while(1) {
        // Read in 2 packets worth of data so we can find the delimiter
        if(!do_read(buf, PKT_LEN * 2))
            return 0;

        // Search buffer for start delimiter
        for(i=0; i < PKT_LEN + 1; i++) {
            if(check_offset(buf, i)) {
                offset = i;
                break;
            }
        }

        // Copy first found packet to buffer
        if(offset >= 0) {
            memmove(buf, &buf[offset], PKT_LEN * 2 - offset);
            if(!do_read(&buf[PKT_LEN * 2 - offset], offset))
                return 0;
        }
        else
            continue;
    }
}  
#endif /* USE_SERIAL */
// vim: nocpandtab
#include "cel_ser.h" /* First for __unix__ and __windows__ definitions */

#include <errno.h>
#include <stdarg.h>
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/uio.h>
#include <unistd.h>
#if defined(__unix__)
#include <sys/ioctl.h>
#else
#include <windows.h>
#endif

#include "cel_ser_util.h"
#if !defined(SOLO_RUN)
#include "ext.h"
#endif

// static int end_node = 10;

int check_offset(unsigned char buf[], int offset) {
    unsigned char *cur;
    // int i;
    cur = &buf[offset];
    if( (cur[0] < 's' || // cur[0] > '9' + 'f' ||
        // cur[1] != 'n' ||
        cur[2] != '\n' ||
        cur[22] != '\r')
        return 0;
    // for (i = 0; i < NUM_SENSORS; i++) {
    //     if((cur[2 + (i * 2)] >> 4) != i + 3)
    //         return 0;
    // }
    return 1;
}

int do_read(unsigned char buf[], int len) {
#if USE_SERIAL == 1
    int ret, size = 0;
    #if defined(__unix__)
    while(size < len) {
        ret = read(fd, buf + size, len - size);
        switch(ret) {
            case 0:
                print_output(1, "cel_ser_read:Read_EOF_from_device_{\%s},__shutting_down", SERDEVICE);
                cel_ser_shutdown();
                return 0;
            break;
            case -1:
                print_output(1, "cel_ser_read:Error_reading_from_device_{\%s},__shutting_down:{\%s}(fd:{\%d}),SERDEVICE, strerror(errno), fd)", SERDEVICE, strerror(errno), fd);
                fflush(stderr);
                cel_ser_shutdown();
                return 0;
            break;
        default:
            size += ret;
            break;
        }
    }
#endif
#endif
}
 Appendix

#else if defined(__windows__)
for(size=0; size < len; size++) {
    long int ignored;
    ret = ReadFile(fd, buf + size, 1, &ignored, NULL);
    if(!ret) {
        print_output(1, "Error num: %d", GetLastError());
        return 0;
    }
}
#endif
#endif
*/USE_SERIAL*/

return 1;
}

void print_packet(unsigned char buf[], int len) {
    int i;
    for(i=0; i < len; i++) {
        print_output(0, "%02x,", buf[i]);
    }
    print_output(1, "\n");
    flush(stdout);
}

void print_output(int newline, const char *format, ...) {
    char buf[1024];
    va_list args;
    va_start(args, format);
    vsprintf(buf, format, args);
#endif
    vsprintf(buf, format, args);
    printf("%s%s\n", buf, newline?"\n":"");
    fflush(stdout);
#ifdef SOLO_RUN
    va_end(args);
#endif
    post(buf, newline);
}

va_end(args);
#if defined(__APPLE__) || defined(__linux__)
#define __unix__
#endif

#elif defined(__WINDOWS__) || defined(WIN32)
#define windows
#define WIN32
#define CRT_SECURE_NO_DEPRECATE
#endif

#ifdef __unix__
//#define SER_DEVICE "/dev/ttyUSB0"
#define SER_DEVICE "/dev/pts/ptc0"
#define SER_SPEED B38400
// old mote baud 11500
#elif defined(__windows__)
#define SER_DEVICE "COM1"
#define SER_SPEED "38400,n,8,1"
// old mote 11500
#endif

#define NUM_NODES 8 // needs to be changed to 10
#define NUM_NODES 1

#define NUM_SENSORS 6 // needs to be changed to 9
#define NUM_SENSORS 10

#define NUM_MAGNOMETERS 3 // can be deleted

#define PKT_LEN ((NUM_SENSORS + 2 + NUM_MAGNOMETERS + 2) * 2 + 2) 
// can be changed to 2 + Num sensors + 2 + 2
#define PKT_LEN (1 + NUM_SENSORS + 2 + 2 + 2)

#define NODE_DATA_SIZE (NUM_SENSORS + NUM_MAGNOMETERS + 1) // can be changed to num sensors + 1
#define NODE_DATA_SIZE (NUM_SENSORS + 1) // can be changed to num sensors + 1

#define DATA_SIZE (NUM_NODES * NODE_DATA_SIZE) // data size will increase to 10 * 10

#define SOLO_RUN //DEBUG
#if defined(__unix__) && defined(SOLO_RUN)
#define CHOOSE_OUTPUT 1
#endif

#define FULL_OUTPUT 1
#define CHANNEL_OUTPUT 2
#define LATENCY_OUTPUT 3
#define LOG_OUTPUT 4
#define NO_OUTPUT 5

#define READ_IN 0
#define READ_ONE 1
#define READ_TWO 2
#define READ_THREE 3
#define USE_SERIAL 1

int cel_ser_setup();
int cel_ser_shutdown();
short *cel_ser_read();
void cel_ser_read_real();
#if defined(__unix__)

#define FD_DECLARE(a) int a = -1
#define EXTERN_FD_DECLARE(a) extern int a
#define SLEEP(a) usleep(a)
#define THREAD_FUNCTION(a) void *(void *arg)
#define THREAD_FUNCTION_RET return NULL
#define THREAD_HANDLE(a) pthread_t a
#define MUTEX_DECLARE(a) pthread_mutex_t a
#define MUTEX_LOCK(a) pthread_mutex_lock(&a)
#define MUTEX_UNLOCK(a) pthread_mutex_unlock(&a)

#if defined(_windows_)
#define FD_DECLARE(a) HANDLE a
#define EXTERN_FD_DECLARE(a) extern HANDLE a
#define SLEEP(a) Sleep(a)
#define THREAD_FUNCTION(a) void a(void *arg)
#define THREAD_FUNCTION_RET return
#define THREAD_HANDLE(a) HANDLE a
#define MUTEX_DECLARE(a) HANDLE a
#define MUTEX_LOCK(a) WaitForSingleObject(a, INFINITE)
#define MUTEX_UNLOCK(a) ReleaseSemaphore(a, 1, NULL)
#endif

#ifndef timersub
#define timersub(a, b, result)
    do { 
        (result)->tv_sec = (a)->tv_sec - (b)->tv_sec;
        (result)->tv_usec = (a)->tv_usec - (b)->tv_usec;
        if ((result)->tv_usec < 0) {
            --(result)->tv_sec;
            (result)->tv_usec += 1000000;
        }
    } while (0)
#endif

CEL_SER_UTIL.H

int check_offset(unsigned char buf[], int offset);
int do_read(unsigned char buf[], int len);
void print_packet(unsigned char buf[], int len);
void print_output(int newline, const char *format, ...);

#if USE_SERIAL == 1
EXTERN_FD_DECLARE(id);
#endif /* USE_SERIAL */
Appendix E

Code for estimating attitude

AHRS_TRIAD.C

#include "ext.h"
#include "ext_mess.h"
#include <string.h>
#include <stdio.h>
#include <math.h>
#define EPSILON 1e-6

//************************CALIBRATION STUFF***************************/
/* INTRODUCE THE OFFSET VALUES IN THE FOLLOWING VARIABLES IN ADC values */
#define ACCELX_OFFSET 2043 // initial offset in accelerometer X
#define ACCELY_OFFSET 2053 // initial offset in accelerometer Y
#define ACCELZ_OFFSET 2264 // initial offset in accelerometer Z
#define MagX_OFFSET 1917 // initial offset in Magnetometer X
#define MagY_OFFSET 1813 // initial offset in Magnetometer Y
#define MagZ_OFFSET 1667 // initial offset in Magnetometer Z

// WE SHOULD CALIBRATE THE FOLLOWING VALUES FOR EACH PARTICULAR IMU axis */
#define ACCELX Resolution 536
#define ACCELY Resolution 661
#define ACCELZ Resolution 559
#define MagX Resolution 186
#define MagY Resolution 189
#define MagZ Resolution 170

//************************ END CALIBRATION STUFF ***************************/

void *this_class; // Required. Global pointing to this class

typedef struct _triad // Data structure for this object
{
    t_object m_ob; // Must always be the first field; used by Max
    Atom m_args[9]; // we want our inlet to be receiving a list of 10 elements
    long m_value; // inlet
    void *m_R1; // these are all the outlets for the 3 X 3 Matrix
    void *m_R2;
    void *m_R3; // R1 --> first top left cell ... R2 middle cell of the first
    void *m_R4; // row ... and so on
    void *m_R5;
    void *m_R6;
    void *m_R7;
    void *m_R8;
    void *m_R9;
    void *m_R10;
    void *m_R11;
    void *m_R12;
    } t_triad;

void *triad_new(long value);
void triad_free(t_triad *triad);

void triad_list(t_triad *x, Symbol *s, short argc, atom *argv);

void MatrixByMatrix(double *Result, double *MatrixLeft, double *MatrixRight);

void Matrix2Quat(double *Quat, double *Matrix);

void Quat2Matrix(double *Matrix, double *Quat);

void inverseQuat(double *InvQuat, double *RegQuat);

void NormQuat(double *YesQuat, double *NotQuat);

void Slerp(double *NewQuat, double *OldQuat, double *CurrentQuat);

void NormVect(double *YesVect, double *NotVect);


// SLERP Variables

double trace, Suca;

double tol[4], omega, sinom, cosom, scale0, scale1, tez, orientationMatrixA[9];

int main(void)
{
    // set up our class: create a class definition
    setup((t_messlist**) &this_class, (method)triad_new, (method)triad_free, (short)
        sizeof(t_triad), 0L, 0GIMME, 0);

    addmess((method)triad_list, "list", 0GIMME, 0);
    addmess((method)triad_assist, "assist", 0CANT, 0);

    finder_addclass("Maths", "triad");

    post("... I'm TRIAD Object! ... from \AHRS\Library ...", 0);

    return 0;
}

/* ----------------- triad_new -----------------*/

void *triad_new(long value)
{
    t_triad *triad;

    triad = (t_triad*)newobject(this_class); // create the new instance and return
    a pointer to it

    triad->mR4 = floatoutput(triad);
    triad->mR3 = floatoutput(triad);
    triad->mR2 = floatoutput(triad);
    triad->mR1 = floatoutput(triad);

    return(triad);
void triad_list(t_triad *x, Symbol *s, short argc, t_atom *argv)
{
    accex_ADCnumber = argv[0].a_w.w_float;
    accey_ADCnumber = argv[1].a_w.w_float;
    accez_ADCnumber = argv[2].a_w.w_float;
    magnx_ADCnumber = argv[3].a_w.w_float;
    magny_ADCnumber = argv[4].a_w.w_float;
    magnz_ADCnumber = argv[5].a_w.w_float;
    ecs = accex_ADCnumber - ACCELX_OFFSET;
    y = accey_ADCnumber - ACCELY_OFFSET;
    z = accez_ADCnumber - ACCELZ_OFFSET;
    mx = (magnx_ADCnumber - MagX_OFFSET);
    my = (magny_ADCnumber - MagY_OFFSET);
    mz = (magnz_ADCnumber - MagZ_OFFSET);

    // Vector ONE is the stongest — Accelerometers
    temp[0] = ecs / ACCELX_Resolution;
    temp[1] = -y / ACCELY_Resolution;
    temp[2] = z / ACCELZ_Resolution;

    // Normalizing Vector Accelerometer
    temp[0] *= accnorm;
    temp[1] *= accnorm;
    temp[2] *= accnorm;

    // Vector TWO is the weakest — Magnetometers
    temp[3] = mx / MagX_Resolution;
    temp[4] = -my / MagY_Resolution;
    temp[5] = mz / MagZ_Resolution;

    // Normalizing Vector Magnetometer
    temp[3] *= magnorm;
    temp[4] *= magnorm;
    temp[5] *= magnorm;

    // —— Gravity
    ref[0] = 0.0;
    ref[1] = 0.0;
    ref[2] = 1.0;

    // —— Magnetic Earth Field
    // check this page http://www.ngdc.noaa.gov/geomagmodels/IGRFWMM.jsp
    //======== use scale object in max/msp to determine values (scale 90 - 90 1 -1) then y and z
    // same but -90 degrees with respect to z
    ref[3] = argv[6].a_w.w_float;
    ref[4] = argv[7].a_w.w_float;
    ref[5] = argv[8].a_w.w_float;

    ref[3] *= earthnorm;
    ref[4] *= earthnorm;
    ref[5] *= earthnorm;
Appendix

TRIAD ALGORITHM

Body Vectors

\[ \text{VectAx} \]
\[ \text{VectAx}[0] = \text{temp}[0]; \]
\[ \text{VectAx}[1] = \text{temp}[1]; \]
\[ \text{VectAx}[2] = \text{temp}[2]; \]

\[ \text{VectAy} \]
\[ \text{MagnCrosProd}_A = 1.0 / \sqrt{1.0 - ( (\text{temp}[0] \times \text{temp}[3]) + (\text{temp}[1] \times \text{temp}[4]) + (\text{temp}[2] \times \text{temp}[5]))}; \]
\[ \text{VectAy}[0] = ((\text{temp}[1] \times \text{temp}[5]) - (\text{temp}[2] \times \text{temp}[4])); \]
\[ \text{MagnCrosProd}_A; \]
\[ \text{VectAy}[1] = ((\text{temp}[2] \times \text{temp}[3]) - (\text{temp}[0] \times \text{temp}[5])); \]
\[ \text{MagnCrosProd}_A; \]
\[ \text{VectAy}[2] = ((\text{temp}[0] \times \text{temp}[4]) - (\text{temp}[1] \times \text{temp}[3])); \]
\[ \text{MagnCrosProd}_A; \]

\[ \text{NormVect}(&\text{VectAy}[0], &\text{VectAy}[0]); \]

\[ \text{VectAz} \]
\[ \text{VectAz}[0] = (\text{VectAx}[1] \times \text{VectAy}[2]) - (\text{VectAx}[2] \times \text{VectAy}[1]); \]
\[ \text{VectAz}[1] = (\text{VectAx}[2] \times \text{VectAy}[0]) - (\text{VectAx}[0] \times \text{VectAy}[2]); \]
\[ \text{VectAz}[2] = (\text{VectAx}[0] \times \text{VectAy}[1]) - (\text{VectAx}[1] \times \text{VectAy}[0]); \]

\[ \text{NormVect}(&\text{VectAz}[0], &\text{VectAz}[0]); \]

Reference System Vectors

\[ \text{VectBx} \]
\[ \text{VectBx}[0] = \text{ref}[0]; \]
\[ \text{VectBx}[1] = \text{ref}[1]; \]
\[ \text{VectBx}[2] = \text{ref}[2]; \]

\[ \text{VectBy} \]
\[ \text{MagnCrosProd}_B = 1.0 / \sqrt{1.0 - ( (\text{ref}[0] \times \text{ref}[3]) + (\text{ref}[1] \times \text{ref}[4]) + (\text{ref}[2] \times \text{ref}[5])); \]
\[ \text{VectBy}[0] = ((\text{ref}[1] \times \text{ref}[5]) - (\text{ref}[2] \times \text{ref}[4])); \]
\[ \text{MagnCrosProd}_B; \]
\[ \text{VectBy}[1] = ((\text{ref}[2] \times \text{ref}[3]) - (\text{ref}[0] \times \text{ref}[5])); \]
\[ \text{MagnCrosProd}_B; \]
\[ \text{VectBy}[2] = ((\text{ref}[0] \times \text{ref}[4]) - (\text{ref}[1] \times \text{ref}[3])); \]
\[ \text{MagnCrosProd}_B; \]

\[ \text{NormVect}(&\text{VectBy}[0], &\text{VectBy}[0]); \]

\[ \text{VectBz} \]
\[ \text{VectBz}[0] = (\text{VectBx}[1] \times \text{VectBy}[2]) - (\text{VectBx}[2] \times \text{VectBy}[1]); \]
\[ \text{VectBz}[1] = (\text{VectBx}[2] \times \text{VectBy}[0]) - (\text{VectBx}[0] \times \text{VectBy}[2]); \]
\[ \text{VectBz}[2] = (\text{VectBx}[0] \times \text{VectBy}[1]) - (\text{VectBx}[1] \times \text{VectBy}[0]); \]

\[ \text{NormVect}(&\text{VectBz}[0], &\text{VectBz}[0]); \]

Multiply vectors and Generate Rotation Matrix

\[ m[0] = \text{VectAx}[0]; \]
\[ m[1] = \text{VectAy}[0]; \]
\[ m[2] = \text{VectAz}[0]; \]
\[ \text{columns has become rows} \]
\[ m[3] = \text{VectAx}[1]; \]
\[ m[4] = \text{VectAy}[1]; \]
\[ m[5] = \text{VectAz}[1]; \]
\[ m[6] = \text{VectAx}[2]; \]
\[ m[7] = \text{VectAy}[2]; \]
\[ m[8] = \text{VectAz}[2]; \]

\[ n[0] = \text{VectBx}[0]; \]
\[ n[1] = \text{VectBy}[0]; \]
\[ n[2] = \text{VectBz}[0]; \]
\[ \text{as it was} \]
\[ n[3] = \text{VectBx}[1]; \]
\[ n[4] = \text{VectBy}[1]; \]
\[ n[5] = \text{VectBz}[1]; \]
\[ n[6] = \text{VectBx}[2]; \]
\[ n[7] = \text{VectBy}[2]; \]
\[ n[8] = \text{VectBz}[2]; \]
Appendix

// We generate our Matrix 3x3 from our 6 vectors the we apply the following transformation {
// in order:
// Conversion Matrix to Quaternion
// Normalize Quaternion
// Invert Quaternion

MatrixByMatrix(&orientationMatrix[0], &m[0], &n[0]);
Matrix2Quat(&quat_e[0], &orientationMatrix[0]);
NormQuat(&quat_e[0], &quat_e[0]);
inverseQuat(&quat_e[0], &quat_e[0]);
Slerp(&quat_new[0], &quat_old[0], &quat_e[0]);
NormQuat(&quat_new[0], &quat_new[0]);
Quat2Matrix(&orientationMatrix[0], &quat_new[0]);

/\ COOK THE OUTPUT \!!!

for (i = 0; i < 9; i++) orientationMatrix[i] = orientationMatrix[i];
for (i = 0; i < 4; i++) quat_old[i] = quat_new[i];

/\ END COOK THE OUTPUT \!!!

void MatrixByMatrix(double *Result, double *MatrixLeft, double *MatrixRight)
{
    *(Result) = (*(*MatrixLeft)) + (*(*MatrixRight)) + (*(*MatrixLeft+1)) + (*(*MatrixLeft+2)) + (*(*MatrixLeft+3)) + (*(*MatrixLeft+4)) + (*(*MatrixLeft+5)) + (*(*MatrixLeft+6));
    *(Result+1) = (*(*MatrixLeft)) + (*(*MatrixRight+1)) + (*(*MatrixLeft+1)) + (*(*MatrixLeft+2)) + (*(*MatrixRight+2)) + (*(*MatrixLeft+3)) + (*(*MatrixLeft+4)) + (*(*MatrixLeft+5));
    *(Result+2) = (*(*MatrixLeft)) + (*(*MatrixRight+2)) + (*(*MatrixLeft+1)) + (*(*MatrixLeft+2)) + (*(*MatrixRight+3)) + (*(*MatrixLeft+3)) + (*(*MatrixLeft+4)) + (*(*MatrixLeft+5));
    *(Result+3) = (*(*MatrixLeft+1)) + (*(*MatrixRight+3)) + (*(*MatrixLeft+2)) + (*(*MatrixRight)) + (*(*MatrixLeft+4)) + (*(*MatrixLeft+5)) + (*(*MatrixRight+4));
    *(Result+4) = (*(*MatrixLeft+1)) + (*(*MatrixRight+4)) + (*(*MatrixLeft+2)) + (*(*MatrixRight+5)) + (*(*MatrixLeft+3)) + (*(*MatrixRight+6));
    *(Result+5) = (*(*MatrixLeft+2)) + (*(*MatrixRight)) + (*(*MatrixLeft+3)) + (*(*MatrixRight+7));
    *(Result+6) = (*(*MatrixLeft+3)) + (*(*MatrixRight+5)) + (*(*MatrixLeft+4)) + (*(*MatrixRight+6));
    *(Result+7) = (*(*MatrixLeft+4)) + (*(*MatrixRight+7));
    *(Result+8) = (*(*MatrixLeft+5)) + (*(*MatrixRight+8));
}

void Matrix2Quat(double *Quat, double *Matrix)
{
    trace = (*(*Matrix)) + (*(*Matrix + 4)) + (*(*Matrix + 8)) + 1.0;
    if (trace > EPSILON)
    {
        Suca = 0.5 / sqrt(trace);
        *(Quat) = 0.25 / Suca;
        //w
        *(Quat + 1) = (*(*Matrix + 7)) - (*(*Matrix + 5));
        Suca; //x
        *(Quat + 2) = (*(*Matrix + 2)) - (*(*Matrix + 6));
        Suca; //y
        *(Quat + 3) = (*(*Matrix + 3)) - (*(*Matrix + 1));
        Suca; //z
    }
    else
    {
        if ((*(Matrix)) > (*(*Matrix + 4)) && ((*(Matrix)) > (*(*Matrix + 8)))
        {
            Suca = sqrt(1.0 + (*(Matrix)) - (*(Matrix + 4)) - (*(Matrix + 8))) * 2.0; // S=sqrt
        }
    }
}

203
Appendix

\*(\text{Quat}) = ( (\text{Matrix} + 7) ) - ( (\text{Matrix} + 5) ) / \text{Suca}; \quad \text{w}
\*(\text{Quat} + 1) = 0.25 * \text{Suca}; \quad // x
\*(\text{Quat} + 2) = ( (\text{Matrix} + 1) ) + ( (\text{Matrix} + 3) ) / \text{Suca}; \quad // y
\*(\text{Quat} + 3) = ( (\text{Matrix} + 2) ) + ( (\text{Matrix} + 6) ) / \text{Suca}; \quad // z
}

\} \text{else if ( (\text{Matrix} + 4) ) > (\text{Matrix} + 8) ) }
\text{Suca} = \sqrt{1.0 + (\text{Matrix} + 4) - (\text{Matrix}) - (\text{Matrix} + 8) )} * 2.0; \quad \text{w}
\*(\text{Quat}) = ((\text{Matrix} + 2) ) - (\text{Matrix} + 6) ) / \text{Suca}; \quad // x
\*(\text{Quat} + 1) = ((\text{Matrix} + 1) ) + (\text{Matrix} + 3) ) / \text{Suca}; \quad // y
\*(\text{Quat} + 2) = (0.25 * \text{Suca}; \quad // y
\*(\text{Quat} + 3) = ( (\text{Matrix} + 5) ) + (\text{Matrix} + 7) ) / \text{Suca}; \quad // z
\}
\}

\} \} \text{else }
\text{Suca} = \sqrt{1.0 + (\text{Matrix} + 8) ) - (\text{Matrix}) - (\text{Matrix} + 4) ) * 2.0; \quad S=4 \text{Qz}
\*(\text{Quat}) = ((\text{Matrix} + 3) ) - (\text{Matrix} + 1) ) / \text{Suca}; \quad // w
\*(\text{Quat} + 1) = ((\text{Matrix} + 2) ) + (\text{Matrix} + 6) ) / \text{Suca}; \quad // x
\*(\text{Quat} + 2) = ( (\text{Matrix} + 5) ) + (\text{Matrix} + 7) ) / \text{Suca}; \quad // y
\*(\text{Quat} + 3) = 0.25 * \text{Suca}; \quad // z
\}
\}

\} \} \text{else}
\text{Suca} = \sqrt{1.0 + (\text{Matrix} + 4) ) - (\text{Matrix}) - (\text{Matrix} + 8) )} * 2.0; \quad \text{w}
\*(\text{Quat}) = ((\text{Matrix} + 2) ) - (\text{Matrix} + 6) ) / \text{Suca}; \quad // x
\*(\text{Quat} + 1) = ((\text{Matrix} + 1) ) + (\text{Matrix} + 3) ) / \text{Suca}; \quad // y
\*(\text{Quat} + 2) = (0.25 * \text{Suca}; \quad // y
\*(\text{Quat} + 3) = ( (\text{Matrix} + 5) ) + (\text{Matrix} + 7) ) / \text{Suca}; \quad // z
\}
\}

\} \} \text{else if ( (\text{Matrix} + 4) ) > (\text{Matrix} + 8) ) }
\text{Suca} = \sqrt{1.0 + (\text{Matrix} + 4) - (\text{Matrix}) - (\text{Matrix} + 8) )} * 2.0; \quad \text{w}
\*(\text{Quat}) = ( (\text{Matrix} + 1) ) + (\text{Matrix} + 3) ) / \text{Suca}; \quad // y
\*(\text{Quat} + 1) = ( (\text{Matrix} + 2) ) + (\text{Matrix} + 6) ) / \text{Suca}; \quad // y
\*(\text{Quat} + 2) = (0.25 * \text{Suca}; \quad // y
\*(\text{Quat} + 3) = ( (\text{Matrix} + 5) ) + (\text{Matrix} + 7) ) / \text{Suca}; \quad // z
\}
\}

void \text{inverseQuat(double *InvQuat, double *RegQuat)}
\{
\quad *\text{InvQuat} = *\text{RegQuat};
\quad *\text{InvQuat} + 1 = -1.0 * (*\text{RegQuat} + 1);
\quad *\text{InvQuat} + 2 = -1.0 * (*\text{RegQuat} + 2);
\quad *\text{InvQuat} + 3 = -1.0 * (*\text{RegQuat} + 3);
\}

void \text{NormQuat(double *YesQuat, double *NotQuat)}
\{
\quad \text{mult} = 1.0 / \sqrt{(pow((*\text{NotQuat}),2) + pow((*\text{NotQuat} + 1),2) + pow((*\text{NotQuat} + 2),2))};
\quad (*\text{YesQuat}) = (*\text{NotQuat}) * \text{mult};
\quad (*\text{YesQuat} + 1) = (*\text{NotQuat} + 1) * \text{mult};
\quad (*\text{YesQuat} + 2) = (*\text{NotQuat} + 2) * \text{mult};
\quad (*\text{YesQuat} + 3) = (*\text{NotQuat} + 3) * \text{mult};
\}

void \text{Slerp(double *NewQuat, double *OldQuat, double *CurrentQuat)}
\{
\quad // SLERP code taken from http://www.gamasutra.com/features/19980703/quaterebony_01.htm
\quad \text{cosom} = (*\text{OldQuat} + 1) * (*\text{CurrentQuat} + 1) + (*\text{OldQuat} + 2) * (*\text{CurrentQuat} + 2) + (*\text{OldQuat} + 3) * (*\text{CurrentQuat} + 3) + (*\text{OldQuat}) * (*\text{CurrentQuat}) ;
\quad \text{tez} = 0.8;
\quad \text{if (cosom < 0.0)}
\quad \quad \text{cosom} = -1.0 * \text{cosom};
\quad \quad \text{tol}[0] = -1.0 * (*\text{CurrentQuat});
\quad \quad \text{tol}[1] = -1.0 * (*\text{CurrentQuat} + 1); 
\quad \quad \text{tol}[2] = -1.0 * (*\text{CurrentQuat} + 2);
\quad \quad \text{tol}[3] = -1.0 * (*\text{CurrentQuat} + 3);
\quad \text{else}
\quad \quad \text{tol}[0] = (*\text{CurrentQuat});
\quad \quad \text{tol}[1] = (*\text{CurrentQuat} + 1); 
\quad \quad \text{tol}[2] = (*\text{CurrentQuat} + 2);
\quad \quad \text{tol}[3] = (*\text{CurrentQuat} + 3); 
\quad \text{if (1.0 - cosom) > EPSILON)}
\quad \quad \text{omega} = \text{acos(cosom)};
\quad \quad \text{sinom} = \sin(\text{omega});
\quad \quad \text{scale0} = \sin((1.0 - \text{tez}) * \text{omega}) / \text{sinom};
\quad \quad \text{scale1} = \sin(\text{tez} * \text{omega}) / \text{sinom};
\quad \text{else}{
scale0 = 1.0 - tez;
scale1 = tez;
}

(*NewQuat) = scale0 * (*OldQuat) + scale1 * tol[0];
(*NewQuat + 1) = scale0 * (*OldQuat + 1) + scale1 * tol[1];
(*NewQuat + 2) = scale0 * (*OldQuat + 2) + scale1 * tol[2];
(*NewQuat + 3) = scale0 * (*OldQuat + 3) + scale1 * tol[3];
}

void Quat2Matrix(double *Matrix, double *Quat)
{
    (*Matrix) = 1.0 - (2.0 * pow(*((Quat + 2)), 2.0)) - (2.0 * pow(*((Quat + 3)), 2.0));
    (*Matrix + 1) = (2.0 * (*((Quat + 1)) + (*((Quat + 2))) - (2.0 * (*((Quat + 3)) + (*((Quat + 1))));
    (*Matrix + 2) = (2.0 * (*((Quat + 1)) + (*((Quat + 3))) + (2.0 * (*((Quat + 2))) + (*((Quat + 1))) - (*((Quat + 1));
    (*Matrix + 3) = (2.0 * (*((Quat + 1)) + (*((Quat + 2))) - (2.0 * (*((Quat + 3))) + (*((Quat + 4))) - (2.0 * pow(*((Quat + 3)), 2.0));
    (*Matrix + 4) = (2.0 * (*((Quat + 2))) + (*((Quat + 3))) + (2.0 * (*((Quat + 2))) + (*((Quat + 1))) - (*((Quat + 3))));
    (*Matrix + 5) = (2.0 * (*((Quat + 2))) + (*((Quat + 3))) - (2.0 * (*((Quat + 2))) + (*((Quat + 1)));
    (*Matrix + 6) = (2.0 * (*((Quat + 1)) + (*((Quat + 2))) - (2.0 * pow(*((Quat + 3)), 2.0));
    (*Matrix + 7) = (2.0 * (*((Quat + 2))) + (*((Quat + 3))) + (2.0 * (*((Quat + 1))) + (*((Quat + 3)));
    (*Matrix + 8) = (2.0 * (*((Quat + 1))) + (*((Quat + 2)));
}

void NormVect(double *YesVect, double *NotVect)
{
    VectBznorm = 1 / sqrt((*NotVect) * (*NotVect) + (*NotVect + 1) * (*NotVect + 2));
    (*YesVect) = VectBznorm;
    (*YesVect + 1) = VectBznorm;
    (*YesVect + 2) = VectBznorm;
}

void triad_assist(t_triad *triad, void *b, long msg, long arg, char *s)
{
    if (msg == ASSIST_OUTLET) {
        switch (arg) {
            case 0: sprintf(s, "%s", "qw_"));
                break;
            case 1: sprintf(s, "%s", "qx"));
                break;
            case 2: sprintf(s, "%s", "qy"));
                break;
            case 3: sprintf(s, "%s", "qz"));
                break;
        }
    } else if (msg == ASSIST_INLET)
        sprintf(s, "%s", "(list)");
}

void triad_free(t_triad *triad)
{
    post("Fine!....BYE!", 0);
}
Appendix
Appendix F: AHRS MAX
Library

List of Externals and Abstractions

Externals:

- ahrs_aed2xyz.mxo:
  converts azimuth and elevation coordinates to cartesian;

- ahrs_axis2quat.mxo:
  converts angle-axis coordinates to quaternion;

- ahrs_gyro_orientation.mxo:
  Estimates orientation of an object with respect of a know reference system

- ahrs_matrix2quat.mxo:
  converts a 3x3 matrix to quaternion;

- ahrs_quat2axis.mxo:
  converts a quaternion to angle-axis coordinates;

- ahrs_quat2matrix.mxo:
  converts a quaternion to 3x3 matrix;

- ahrs_quatadd.mxo:
  adds two quaternions

- ahrs_quatconj.mxo:
  conjugates two quaternions

- ahrs_quatdiv.mxo:
  division operator for quaternions

- ahrs_quatinv.mxo:
  inverts a quaternion

- ahrs_quatmagn.mxo:
  calculates the magnitude of a quaternion
Appendix

• ahrs_quatmult.mxo:
  multiply two quaternions
• ahrs_quatnormalize.mxo:
  normalises a quaternion
• ahrs_quatsub.mxo:
  subtracts two quaternions
• ahrs_triad.mxo:
  TRIAD algorithm.
• ahrs_vecadd.mxo:
  adds two vectors
• ahrs_veccross.mxo:
  cross product between two vectors
• ahrs_veccross.mxo:
  normalises a quaternion
• ahrs_vecsub.mxo:
  subtracts two vectors
• ahrs_vecmag.mxo:
  calculates the magnitude of a vector
• ahrs_vecmagcross.mxo calculates the magnitude of the cross product of two given vectors
• ahrs_vecnormalize.mxo:
  normalises a vector
• ahrs_vecsub.mxo:
  subtracts two vectors
• ahrs_xyz2aed.mxo converts Cartesian coordinates to azimuth, elevation and distance

Abstractions:

• ahrs_accel_accel.mxb:
  converts ADC accelerometer values to m/sec^2
• ahrs_accel_velocity.mxb:
  numerically integrates acceleration values to retrieve velocity
• ahrs_accel_distance.mxb:
  numerically integrates velocity values to retrieve distance
• ahrs_gyro_velocity.mxb converts ADC gyroscope value to °/s
• ahrs_gyro_angle.mxb numerically integrates angular speed to retrieve angle
• ahrs_matrixnormalize.mxb normalises a 3x3 matrix
• ahrs_offset.mxb calculates the offset for incoming sensor data
• ahrs_resolution.mxb calculates resolution for incoming sensor data
Appendix G: AGORÁ Sound Sources

Selection of forty songs from personal iPod Favourite playlist

Appendix

16. Reich, S. (1997), ‘Tehillim Part I (Fast)’, (Box Set) CD 1 - Track 40 of Steve Reich Works 1965-95, Nonesuch (Warner)


References


REFERENCES


REFERENCES


Cycling 74 (2010), MAX [online], available: http://cycling74.com/ [accessed: 11 October 2012]. [81]


REFERENCES


REFERENCES


REFERENCES


REFERENCES


REFERENCES

on New Interfaces for Musical Expression, Pittsburgh, June 4-6, 2009, Pittsburgh, PA - USA: Carnegie Mellon University, 45-46. [48] [49]


REFERENCES


222


REFERENCES


Nordic Semiconductor (2012), Nordic nRF24L01 [online], available: http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01 [accessed: 11 October 2012].


REFERENCES


REFERENCES

Rauterberg, M. (2012), *History of HCI - Sayre Glove* [online], available: http://www.idemployee.id.tue.nl/g.w.m.rauterberg/presentations/HCI-history/tsld065.htm [accessed: 11 October 2012]. 31


Santayana, G. (1922), *Soliloquies in England and Later Soliloquies*, New York: Charles Scribner’s Sons. 11


226


Shirley, S. (2006), Mote I/O Driver, Interaction Design Centre - University of Limerick, unpublished. 9 79 183


REFERENCES


REFERENCES


REFERENCES


230
REFERENCES
