MusE-XR: Musical Experiences in Extended Reality
to Enhance Learning and Performance

by

David Johnson
B.Sc., College of Charleston, 2004
M.Sc., College of Charleston, 2013

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Computer Science

MusE-XR: Musical Experiences in Extended Reality
to Enhance Learning and Performance

by

David Johnson
B.Sc., College of Charleston, 2004
M.Sc., College of Charleston, 2013

Supervisory Committee

Dr. George Tzanetakis, Supervisor
(Department of Computer Science)

Dr. Daniela Damian, Co-Supervisor
(Department of Computer Science)

Dr. Peter Driessen, Outside Member
(Department of Electrical Engineering)

Supervisory Committee

Dr. George Tzanetakis, Supervisor
(Department of Computer Science)

Dr. Daniela Damian, Co-Supervisor
(Department of Computer Science)

Dr. Peter Driessen, Outside Member
(Department of Electrical Engineering)

ABSTRACT

Integrating state-of-the-art sensory and display technologies with 3D computer graphics, extended reality (XR) affords capabilities to create enhanced human experiences by merging virtual elements with the real world. To better understand how Sound and Music Computing (SMC) can benefit from the capabilities of XR, this thesis presents novel research on the design of musical experiences in extended reality (MusE-XR). Integrating XR with research on computer assisted musical instrument tutoring (CAMIT) as well as New Interfaces for Musical Expression (NIME), I explore the MusE-XR design space to contribute to a better understanding of the capabilities of XR for SMC.

The first area of focus in this thesis is the application of XR technologies to CAMIT enabling extended reality enhanced musical instrument learning (XREMIL). A common approach in CAMIT is the automatic assessment of musical performance. Generally, these systems focus on the aural quality of the performance, but emerging XR related sensory technologies afford the development of systems to assess playing technique. Employing these technologies, the first contribution in this thesis is a CAMIT system for the automatic assessment of pianist hand posture using depth data. Hand

posture assessment is performed through an applied computer vision (CV) and machine learning (ML) pipeline to classify a pianist's hands captured by a depth camera into one of three posture classes. Assessment results from the system are intended to be integrated into a CAMIT interface to deliver feedback to students regarding their hand posture. One method to present the feedback is through real-time visual feedback (RTVF) displayed on a standard 2D computer display, but this method is limited by a need for the student to constantly shift focus between the instrument and the display.

XR affords new methods to potentially address this limitation through capabilities to directly augment a musical instrument with RTVF by overlaying 3D virtual objects on the instrument. Due to limited research evaluating effectiveness of this approach, it is unclear how the added cognitive demands of RTVF in virtual environments (VEs) affect the learning process. To fill this gap, the second major contribution of this thesis is the first known user study evaluating the effectiveness of XREMIL. Results of the study show that an XR environment with RTVF improves participant performance during training, but may lead to decreased improvement after the training. On the other hand,interviews with participants indicate that the XR environment increased their confidence leading them to feel more engaged during training.

In addition to enhancing CAMIT, the second area of focus in this thesis is the application of XR to NIME enabling virtual environments for musical expression (VEME). Development of VEME requires a workflow that integrates XR development tools with existing sound design tools. This presents numerous technical challenges, especially to novice XR developers. To simplify this process and facilitate VEME development, the third major contribution of this thesis is an open source toolkit, called OSC-XR. OSC-XR makes VEME development more accessible by providing developers with readily available Open Sound Control (OSC) virtual controllers. I present three new VEMEs, developed with OSC-XR, to identify affordances and guidelines for VEME design.

The insights gained through these studies exploring the application of XR to musical learning and performance, lead to new affordances and guidelines for the design of effective and engaging MusE-XR.

Contents

Supervisory Committee ii

Abstract iii

Table of Contents v

List of Tables ix

List of Figures x

Glossary xiv

Acronyms xvi

1 Introduction 1
- 1.1 Research Goals . . . 3
- 1.2 Extended and Mixed Reality . . . 3
  - 1.2.1 Affordances of XR . . . 5
- 1.3 Musical Experiences in Extended Reality . . . 6
  - 1.3.1 Computer Assisted Musical Instrument Tutoring . . . 7
  - 1.3.2 New Interfaces for Musical Expression . . . 10
- 1.4 Research Contributions . . . 11
  - 1.4.1 Publications . . . 13
- 1.5 Thesis Outline . . . 14

2 Background 16
- 2.1 Computer Assisted Musical Instrument Tutoring . . . 17
  - 2.1.1 Music Learning . . . 17
  - 2.1.2 Automatic Assessment of Musical Performance . . . 22
  - 2.1.3 Interfaces for Enhancing Musical Practice . . . 28

2.2 New Interfaces for Musical Expression . . . 32
2.2.1 Digital Music Instruments . . . 32
2.2.2 Hyperinstruments . . . 34
2.2.3 Open Sound Control . . . 36
2.3 Extended Reality . . . 38
2.3.1 XR Training . . . 38
2.3.2 XR and SMC . . . 44
2.4 Challenges . . . 49

3 Research Methodology 52
3.1 Research Exploration and Conceptualization . . . 53
3.1.1 Exploratory Research . . . 53
3.1.2 Conceptualization of Research Tracks . . . 55
3.2 Research Tracks . . . 56
3.2.1 Track 1: CAMIT . . . 56
3.2.2 Track 2: MusE-XR . . . 58
3.3 Research Project Methodologies . . . 59
3.3.1 Automatic Assessment of Pianist Hand Posture . . . 59
3.3.2 Evaluating the Effectiveness of XREMIL . . . 62
3.3.3 Virtual Environments for Musical Expression . . . 65

4 Automatic Assessment of Pianist Hand Posture using Depth Data 67
4.1 Introduction . . . 68
4.1.1 Pianist Hand Posture . . . 68
4.2 Approach for Hand Posture Assessment . . . 70
4.2.1 Hand Segmentation . . . 72
4.2.2 Feature Extraction . . . 74
4.3 Feasibility Assessment . . . 75
4.3.1 Prototype Description . . . 75
4.3.2 Experiments . . . 78
4.3.3 Discussion . . . 83
4.4 System Description . . . 84
4.4.1 Data Collection . . . 85
4.4.2 Hand Segmentation . . . 86

4.4.3 Hand Posture Detection 90
4.5 Experiments 91
4.5.1 Hand Segmentation 92
4.5.2 Hand Posture Detection 94
4.6 Discussion 103
4.6.1 Considerations for Interface Design 103
4.6.2 Future Work 105
4.7 Conclusion 107

5 Evaluating the Effectiveness of XREMIL 109
5.1 Introduction 110
5.1.1 The Theremin 112
5.1.2 Contributions and Outline 112
5.2 System Design 113
5.2.1 Design Guidelines for XREMIL 114
5.2.2 Prototype Design 116
5.2.3 Heuristic Evaluation of Prototype 121
5.2.4 *MR:emin* Description 124
5.3 User Study 127
5.3.1 Research Questions 128
5.3.2 Study Design 130
5.3.3 Performance Task and Evaluation Metrics 132
5.3.4 Procedure 132
5.4 Results 133
5.4.1 Quantitative Analysis of Objective Data 133
5.4.2 Quantitative Analysis of Subjective Data 136
5.4.3 Qualitative Analysis 141
5.5 Discussion 143
5.5.1 Factors of *MR:emin* on Learning Transfer 143
5.5.2 User Experience of *MR:emin* 145
5.5.3 Threats to Validity 146
5.6 Conclusion 146

6 Virtual Environments for Musical Expression 149
6.1 Introduction 150

6.2 Unity OSC Library 150
6.3 OSC-XR 153
6.3.1 OSC Controller Prefabs and Scripts 154
6.3.2 Control Data Validation 155
6.4 OSC-XR Use Cases 158
6.4.1 The Sonic Playground 158
6.4.2 Virtual Hyperinstruments 161
6.4.3 Immersive Vis Control 163
6.5 Conclusion 166

7 Conclusion **167**
7.1 Discussion 168
7.1.1 XREMIL 168
7.1.2 Design Considerations 170
7.2 Future Work 173

A Publications **178**
A.1 Publications from this Research 178
A.2 Publications not from this Research 179

B Publically Available Software **180**

Bibliography **181**

List of Tables

Table 4.1 5-fold cross validation accuracy averages for each session 80
Table 4.2 Average hand posture detection accuracy for each depth map size per descriptor type 95
Table 4.3 Class counts per participant 99

Table 5.1 Findings for each design guideline from the implementation of a Heuristic Evaluation (italics indicate a potential design issue). 122

Table 6.1 Examples of available OSC-XR controller prefabs and scripts 153
Table 6.2 Average errors for each evaluation task 158

List of Figures

Figure 1.1 Reality-Virtuality Continuum (Milgram et al., 1995) . . 3

Figure 2.1 The AMIR marker-based motion capture system for violin technique assessment (Ng et al., 2007). 24
Figure 2.2 The Conducting Tutor interface with body tracking implemented using the Kinect (Salgian and Vickerman, 2016). 26
Figure 2.3 The Yousician piano lesson interface. The colored notes in the score map to the colored keys on the piano to teach students which keys to press. 29
Figure 2.4 A xylophone hyperinstrument augmented with virtual faders (Trail et al., 2012). 36
Figure 2.5 The Music Everywhere AR environment for piano tutoring (Das et al., 2017). 43
Figure 2.6 The Wedge interface for building and performing immersive musical environments (Moore et al., 2015) ©2015 IEEE. 46

Figure 3.1 Methodology for research on musical experiences in extended reality (MusE-XR) 53
Figure 3.2 Research Methodology for Experiments on the Automatic Assessment of Pianist Hand Posture 60
Figure 3.3 Research Methodology for Evaluating the Effectiveness of XREMIL 62
Figure 3.4 Research Methodology for the Design of the OSC-XR Toolkit for Prototyping VEME 65

Figure 4.1 The three common hand postures of beginning piano students that are detected with the presented system. Figures 4.1a and 4.1b show common postures mistakes made by students, while Figure 4.1c shows the hand in the ideal posture for pianists. 69
Figure 4.2 The depth camera is positioned with an aerial viewpoint to capture both hands from overhead. Figure 4.2b shows the RGB view of the camera which is used for data annotation. Figure 4.2c shows an example of a depth map that is used for model training and detection. 71
Figure 4.3 **a)** Original depth map from the Kinect, **b)** Hands segmented from Kinect depth map, **c)** Original depth map from the Intel Realsense, **d)** Hands segmented from the Realsense depth map 76
Figure 4.4 Hand Posture Detection Accuracy Rates 81
Figure 4.5 Normalized confusion matrices for each session using HONV 82
Figure 4.6 The posture detection pipeline used for assessing hand posture from single depth maps. 84
Figure 4.7 Examples of DIF and DCF offsets for extracting features of a single pixel in a depth map used to classify the pixel as either hand or background. 88
Figure 4.8 Per pixel classification results of hand segmentation using DCF and DIF with varying radius and neighborhood sizes. 92
Figure 4.9 Individual participant posture detection accuracy of different depth map sizes when using HOG and HONV descriptors 96
Figure 4.10 Hand posture detection accuracy for HOG and HONV with different cell and block sizes 97
Figure 4.11 Hand posture detection accuracy with different oversampling methods. 100
Figure 4.12 Confusion matrices for each student posture model trained without oversampling 101

Figure 4.13 Confusion matrices for each student posture model trained using SVM SMOTE oversampling 102

Figure 5.1 Milgram's Reality-Virtuality Continuum 110
Figure 5.2 The author practicing the theremin using VRMin and a screen capture of the learning environment. 117
Figure 5.3 Performance analysis plots of practice sessions with and without VRMin 120
Figure 5.4 The *MR:emin* XREMIL Environment 126
Figure 5.5 A participant performing in each of the three different training environments. 131
Figure 5.6 Performance data from the training sessions of participants from each study sample 135
Figure 5.7 Boxplots for the performance metric, D, of each sample during training 136
Figure 5.8 Boxplots for PI from pre-test to post-test for each sample 137
Figure 5.9 Boxplots for the total NASA TLX assessment score . .138
Figure 5.10 Boxplots for individual NASA TLX subscale scores. . .139
Figure 5.11 Boxplots of individual UEQ subscale scores 140

Figure 6.1 Example Unity Inspector Interfaces from the OSC-XR toolkit 151
Figure 6.2 The results of control data validation for slider controllers and pad controllers. 156
Figure 6.3 The Sampler Zone VEME which includes OSC-XR pads to trigger audio samples and corresponding OSC-XR sliders to control sample playback rate. 159
Figure 6.4 Hyperemin, a virtual theremin hyperinstrument with real-time ASP controlled with an OSC-XR 3D Grid. . .162
Figure 6.5 The t-SNE view control interface with OSC-XR sliders to control T-SNE view parameters such as rotation and scale. 164

Figure 6.6 The t-SNE visualization parameter control interface with OSC-XR sliders to control T-SNE parameters and an OSC-XR pad to trigger visualization refresh with the new parameters. 165

Glossary

affordance the properties of an object or environment that provide an understanding of what it offers typical participants. 5, 6, 10, 12, 15, 66

depth camera a sensor that is capable of producing an 2D image representation, i.e. a depth map, that contains the distances from the sensor to points in a scene. iv, xiv, xv, 8, 11, 12, 14, 25, 27, 28, 35, 53, 55–57, 67, 70, 72, 75–78, 85, 103, 107, 174

depth map a single channel image, produced by a depth camera, in which each pixel represents a distance value rather than a color. ix, xi, xiv, 8, 11, 27, 28, 60, 61, 70–78, 84–91, 93–96, 108, 169

Kinect a depth camera and motion sensing device released by Microsoft. x, 25–27, 35, 53, 55, 57, 70, 75, 78–80

learning transfer the degree to which learning in one environment affects performance of another task (Cormier and Hagman, 1987). 39–42, 130, 133, 134, 141–143, 175

MIDI a common technical standard and communication protocol commonly used with digital musical devices to enable communication between devices and computer systems.. 27, 36, 68, 114, 117, 124, 125, 127, 132, 142

OSC a common communication protocol for NIME, that enables distributed communication between a controller device and a sound engine. iv, 12, 16, 32, 36–38, 48, 49, 51, 66, 117, 119, 150, 152–155, 160, 161, 166

RDF an ensemble classifier composed of T decision trees whose predictions are aggregated using votes weighted by the posterior probabilities to make the final prediction. 72, 73, 83, 86, 89, 90, 92

Realsense a class of depth cameras and motion sensing devices released by Intel. 70, 75, 78, 79, 85

RGB camera a traditional camera, as opposed to a depth camera that produces an image in the RGB color space. 27, 61, 70, 76, 85

SVM a linear classification model for supervised learning which represents samples as points in high dimensional space and distinguishes classes of samples by calculating the ideal hyperplane separating them. 78–80, 90, 91, 94, 95, 98–100, 108

Acronyms

ABC Applied and Basic Combined. 52, 56, 59

ADASYN Adaptive Synthetic Sampling. 99, 100, 103

AR augmented reality. 2–4, 36, 42, 43, 47, 63, 111, 144, 147, 172, 175

ASP audio signal processing. xii, 23, 24, 53, 54, 68, 162

CAMIT computer assisted musical instrument tutoring. iii, iv, 3, 7–12, 14, 16–26, 28–30, 38, 49, 50, 55–59, 62, 63, 67, 68, 103, 105, 109, 110, 112, 128, 147, 149, 168, 169, 176

CV computer vision. iv, 11, 12, 25–28, 55, 57, 59, 60, 67, 71, 72, 74, 79, 108, 169, 174

DCF depth context feature. xi, 73, 86–88, 90, 92–94, 107

DIF depth image feature. xi, 72, 73, 86–88, 90, 92, 93, 107

DMI digital music instrument. 32–34

FPS frames per second. 85, 152

HCI Human Computer Interaction. 1, 33, 34, 53–55, 115, 173, 174

HE Heuristic Evaluation. 113, 115, 121, 124

HMD head mounted display. 40, 64, 70, 114, 123, 124, 130, 162

HOG histograms of oriented gradients. xi, 74, 75, 77, 79, 80, 90, 94–98

HONV histograms of normal vectors. xi, 74, 75, 77, 79, 80, 82, 90, 94–98, 108

ML machine learning. iv, 11, 12, 24, 27, 28, 49, 53–55, 57, 60–62, 67, 71, 74, 76, 85, 169, 174

MR mixed reality. 3–5, 43, 63, 64, 111, 113, 116, 128, 144, 147

MusE-XR musical experiences in extended reality. iii, iv, vi, x, 3, 5, 7, 11, 12, 15, 16, 34, 47, 48, 51, 53, 56, 58, 65, 66, 150, 166, 167, 170–173, 176, 177

NASA TLX NASA Task Load Index. xii, 133, 136–139

NIME New Interfaces for Musical Expression. iii, iv, xiv, 7, 10–12, 14, 16, 32, 36–38, 44, 49, 55, 58, 63, 65, 66, 149, 161

NUI natural user interaction. 4, 53

RTVF real-time visual feedback. iv, 9, 11, 12, 14, 17, 21, 30, 32, 43, 50, 55, 57, 63, 64, 105, 109–111, 115, 124, 128, 129, 169, 173, 175

RV Reality-Virtuality. 3, 4, 111, 147, 172

SMC Sound and Music Computing. iii, vi, 2, 3, 5–7, 11, 14–16, 44, 52, 53, 59, 65, 66, 168, 170

SMOTE Synthetic Minority Over-sampling Technique. 99, 100, 103, 108

TELMI Technology Enhanced Learning of Musical Instruments. 68, 110

UEQ User Experience Questionnaire. 133, 136, 138, 145

VE virtual environment. iv, 4, 5, 8, 12, 39, 42, 45, 48, 113–115, 118, 121–123, 125, 126, 130, 133, 142, 144, 145, 147, 170, 176, 177

VEME virtual environments for musical expression. iv, x, xii, 10–12, 15, 36, 38, 44, 47, 50, 58, 59, 65, 66, 114, 149, 159, 170, 172, 173, 176, 177

VR virtual reality. 1–6, 38, 44, 111, 113, 116, 117, 144, 172, 173

VRMI virtual reality music instruments. 38, 47, 114, 150

WMR Windows Mixed Reality. 64, 124

XR extended reality. iii–vi, 2–12, 14–17, 30, 34, 36, 38–52, 56–59, 63–66, 70, 72, 105, 109–112, 114, 115, 121, 124, 128, 143, 147, 149, 150, 152, 161–163, 166–168, 170–176

XREMIL extended reality enhanced musical instrument learning. iii, iv, vi–viii, x, xii, 8, 9, 12, 14, 15, 17, 29, 30, 32, 36, 38, 42–44, 50, 57, 58, 62–64, 109, 111–117, 123, 124, 126, 128, 129, 143, 149, 168–170, 172, 173, 175, 176

Chapter 1

Introduction

> VR is the technology that highlights the existence of your subjective experience. It proves you are real.
>
> *Jaron Lanier* (Lanier, 2017)

In 1965, Human Computer Interaction (HCI) innovator Ivan Sutherland (1965) described his vision for the future of computing:

> The ultimate display would, of course, be a room within which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked.

Sutherland (1968) would later lay the foundations to support his vision by developing the first head mounted display for virtual reality (VR). Since then there have been significant research efforts in developing the technologies needed to realize this vision.

Recently there has been a resurgence of research of technologies for extending human capabilities and experiences through virtual augmentation and simulation. With major tech companies, such as Facebook, Google, and Microsoft, joining the space, extended reality (XR) technologies have begun to enter the mainstream. XR is a new term that encompasses the set of computer mediated experiences that extend the real world through virtual simulation or augmentation and the technologies that enable such experiences, such as display technologies, including virtual reality (VR) and augmented reality (AR), input controllers and user tracking sensors, and haptic devices. Enterprises have started to realize the benefits of XR (Rogers, 2019) but general consumer interest appears to be waning due to a perceived lack of benefits and compelling user experience design (Pettey, 2018). Although XR technology has shown potential to enhance human experiences in areas such as education and training, science, sports and exercise (Slater and Sanchez-Vives, 2016) as well as in the enterprise (Rogers, 2019), a lack of compelling applications outside of gaming has hindered its adoption amongst general consumers. Jerald (2016, p. 473) argues that VR (and related XR technologies) should be presented to the general population by engaging benefits of the technologies that

B_1 provide experiences and entertainment that no other technology can provide,

B_2 enable networked worlds for enhanced socialization,

B_3 make people's lives easier and better fulfill their needs,

B_4 improve well-being by providing immersive health care as well as physical and mental exercise, and

B_5 increase cost savings and profitability.

Researchers and developers are more likely to find compelling XR applications that drive demand by focusing on applications that support any of these benefits. By applying XR to Sound and Music Computing (SMC), this dissertation supports the promotion of XR by enabling the development of musical experiences that no other technology can provide (B_1), that make

people's live easier through enhanced learning (B_3), and that promote mental exercise by providing new methods for musical performance and learning, (B_4).

1.1 Research Goals

The overall goal of this research is to enhance applications in SMC through the application of emerging XR technologies. Through this applied approach, I aim to further the knowledge on the design and effectiveness of musical experiences in extended reality (MusE-XR) to enhance learning and performance. This results in two supporting research goals. First, I aim to gain an understanding of the challenges and limitations for designing practical computer assisted musical instrument tutoring (CAMIT) systems that integrate XR technology. Second, I aim to gain an understanding of the affordances of XR to facilitate the development and design of MusE-XR.

1.2 Extended and Mixed Reality

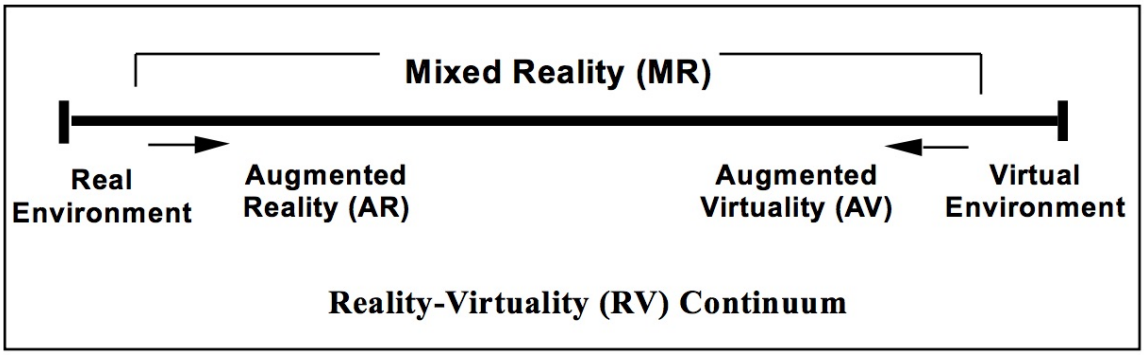


Figure 1.1: Reality-Virtuality Continuum (Milgram et al., 1995)

Composed of technologies that span the Reality-Virtuality (RV) Continuum (Milgram et al., 1995), see Figure 1.1, XR enables the extension of the human experience by combining the digital and physical worlds. This is supported through immersive display technologies, such as VR, augmented reality (AR), and MR, as well as through hardware and software technologies, including sensory interfaces and applications. Being a fairly

new term, XR, is not yet prevalent in the literature, but I use it throughout this dissertation because it best conveys the idea of extending human experiences and the breadth of technologies necessary for implementing such experiences.

At a minimum, developing XR experiences requires a display device, an input device, and a virtual environment (VE). Each of these components utilizes a range of technologies covering the entire RV continuum. The level of realism and fidelity of each component combines with the others to influence the XR system's location on the continuum.

VR and AR are two common categories of hardware technologies that enable the 3D graphics, spatialized sound and motion tracking necessary for simulated and augmented user experiences. The main difference between these two technologies is the level in which the user is immersed into a simulated environment. On the far right of the RV continuum, is a fully virtual experience, typical of standard VR, in which a user is completely immersed in simulated virtual environment (VE) and interacts only with virtual objects. On the other end of the continuum is AR, an experience in which a user is situated in the real world with the simulated environment overlaying the user's environment. During the AR experience, a user can interact with both virtual or real objects. User experiences that fall in between these two extremes are considered MR.

The discussion of the RV Continuum by Milgram et al. (1995) focuses primarily on display technology, but it is not the only component that influences the level of reality or virtuality in an XR system. Input devices play a part as well and can be oriented on the RV continuum. Towards the far right of the continuum are VR input controllers, such as the Oculus Touch controllers (Oculus, 2019), which track motion but require button presses and joystick movements to control interactions. Towards the left end of the continuum are natural user interaction (NUI), such as the Kinect sensor (Microsoft, 2019a) or the Leap Motion (2019), that enable gesture tracking, affording real-world like interactions. Integrating a NUI with VR moves the experience towards the left of the RV continuum bringing VR into the MR space. Additionally, real world objects that have their own sensing or tracking capabilities can be integrated into a VE allowing a user to interact with objects as they normally would in the real world supporting a MR experi-

ence. I clarify the definition of MR because work in this thesis has a focus on enabling MR experiences by integrating natural user interactions and physical objects into immersive VEs.

1.2.1 Affordances of XR

"The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill" (Gibson, 1979). Norman (2002) went on to expand this definition to clarify that an affordance is a relationship between the characteristics of an object with capabilities of the user. In other words, affordances are properties of an object or environment that provide an understanding of what it offers typical participants.

As technologies mature and designers gain experience with them, guidelines and principles emerge to aid in the design of systems. Emerging technologies, however, are often lacking in such recommendations to support design decisions. Designers can look to similar fields for inspiration and guidance, but first they need to know what a technology offers its users. In other words, designers using emerging technology need an understanding of the affordances of the system.

The application of XR to SMC is still emerging and lacks clear guidelines and affordances to inform the design of MusE-XR. One of the goals of this dissertation is to explore this design space and identify new methods to integrate these two fields. Without a clear set of design guidelines, a first investigation seeks to understand the different affordances of XR.

XR has some clear affordances on its own but there is limited research enumerating them (Dalgarno and Lee, 2010; Elliott et al., 2015). Dalgarno and Lee (2010) explore the affordances of learning environments and discuss the affordances specific to this space. Elliott et al. (2015) explore the affordances of VR specific to software engineering research, but they also introduce three general categories of affordances: spatial cognition, manipulation and motion, and feedback. Instead of categorizing the types of affordances, I suggest looking at the high level affordances of XR:

A_{XR_1} *a digital 3D visual layer* generated by a display device allowing for the simulation of virtual worlds or augmentation of the real world,

A_{XR_2} *spatial representation of virtual objects* allowing users to perceive objects similar to how they would in the real world,

A_{XR_3} *registration and localization of real world objects* to add new interactions to existing physical objects through visual overlays,

A_{XR_4} *reality based interactions* provided by natural input devices and realistic physics engines, allowing users to interact with objects using real world like interactions,

A_{XR_5} *non-reality based interactions*, using simulation, computer generated graphics, and the ability to defy the laws of physics, allowing users to interact in ways not possible in the real world, and

A_{XR_6} *enhanced modes of collaboration* allowing users in distributed locations to more easily work together.

Understanding these high level affordances enables research for identifying new ones specific to SMC.

1.3 Musical Experiences in Extended Reality

Jaron Lanier, an early pioneer in VR research (and often thought to have coined the term Virtual Reality), performed *The Sound of One Hand* (Lanier, 2017), a live improvisation of VR music instruments. One of the notable aspects of this performance was that Lanier performed multiple instruments using only one hand, creating a performance that could not be accomplished in the real world. Lanier saw the potential to use XR technologies for the creation and performance of music not otherwise possible. This experience demonstrates the potential to support B_1, but little research applying XR to music has occurred since.

It is clear through years of research that musical education and performance lead to increased learning and cognitive development. Yet as the benefits are clear, public access to musical education is in decline (Kratus, 2007; Aróstegui, 2016). With the decline in decline in music eduction, a report by the Associated Board of the Royal Schools of Music (ABRSM,

2014) indicates that the onus is now on individuals to pursue musical education but the associated costs of music education are a major barrier for students from lower socioeconomic groups. Furthermore, students without musical instruction are less motivated to continuing playing (ABRSM, 2014). As Lanier demonstrates, XR affords new musical experiences to address these challenges by making musical learning and performance more motivating and accessible. At the same time, identifying new techniques and tools for MusE-XR supports the adoption of XR by facilitating experiences that engage benefits B_1, B_3, and B_4.

SMC covers a wide breadth of research on computational approaches for developing innovative musical experiences. Two aspects relevant to this dissertation are computer assisted musical instrument tutoring (CAMIT) and New Interfaces for Musical Expression (NIME).

1.3.1 Computer Assisted Musical Instrument Tutoring

Learning to play a musical instrument is challenging and requires years of disciplined practice to master. Typically, aspiring musicians rely on lessons with a professional teacher to supervise their learning. In order to improve their playing abilities, students must augment lessons with daily practice where they are expected to gradually be able to self analyze their performance. Without a teacher present, however, students must wait until their next lesson to have the teacher verify they are practicing properly. The Internet and tools such as YouTube have made it easier for students to find resources for self-teaching. These tools, however, lack the feedback and personal guidance provided by a trained professional. Research in the field of CAMIT attempts to improve the learning process through automated training and assessment of musical performance (Percival et al., 2007). CAMIT research aims to enhance self-teaching and teacher led instruction by augmenting daily practice sessions with additional feedback about the quality of student performance. The success of computer based music education platforms (Yousician, 2019; Skoove, 2019), show that there is demand for new musical learning methods. There are challenges and limitations, however, to their approaches which use traditional 2D displays to present users with feedback on their performance, discussed further in Section 2.1. To

address these limitations, I explore methods to integrate XR with CAMIT to enhance the musical learning process through the study of **extended reality enhanced musical instrument learning (XREMIL)**.

Automatic assessment of musical performance is a core component of CAMIT research. Most assessment systems, however, only focus on the musical quality of the performance, including the identification of pitch and timing errors (Dannenberg et al., 1993; Schoonderwaldt et al., 2005; Lu et al., 2008) as well as timbre quality (Giraldo et al., 2019). Proper technique is also an important component of musical instrument learning, but one that has seen less attention in CAMIT research. I believe this may have been due to a lack of accessible technologies with the capabilities needed for tracking musicians' body movements. Early research on technique assessment used expensive equipment not available to the general population (Mora et al., 2006; Ng et al., 2007). With the emergence of commoditized sensory interfaces for motion tracking, such as depth cameras, automatic assessment of technique has become more accessible to everyday consumers. A depth camera is a sensor capable of producing a 2D image representation, i.e. a depth map, that contains the distances from the sensor to points in a scene. They are an important technology for XR, enabling natural interaction VEs without the need for physical controllers. Employing a depth camera with a CAMIT for the automatic assessment of musical technique affords capabilities to integrate XR with the system. To this end, I pose the following research question:

RQ_1 Can XR related sensory technology be used effectively for the automatic assessment of musical technique?

To address this question, I research the development of a CAMIT system for the automatic assessment of pianist hand posture using depth camera data, i.e. depth maps. Through the research process, I aim to understand the technical needs for implementing a CAMIT system using accessible technologies that can be integrated with an XR experience.

CAMIT systems that automatically assess performance require a well designed interface to provide students results of their assessment. Some CAMIT systems provide offline feedback (Schoonderwaldt et al., 2005; Lu et al., 2008; Blanco and Ramirez, 2019) which allows students to analyze

the results without having to also focus on their musical performance. Students with limited musical experience, however, may not understand the results or know when exactly the errors were made during the performance. Providing assessment results alongside a recording of the performance has been employed to address this limitation (Ng et al., 2007). Another method to overcome the limitation is to provide students with real-time visual feedback (RTVF) as they are practicing. This is an approach taken by publicly available CAMIT systems, Yousician (2019) and Skoove (2019). A limitation with this method is that it requires students to look at separate 2D display as they are practicing, constantly shifting focus between the instrument and the display. XR affords a new technique with visual overlays displaying RTVF directly on the musical instrument. While there have been a few systems that implement this technique (Huang et al., 2011; Chow et al., 2013), there is no known research on the effectiveness of XREMIL with RTVF. To this end, I pose the following research question:

RQ_2 Is real-time visual feedback (RTVF) with XREMIL effective for musical learning?

To address this question, I administer a user study to evaluate the effectiveness of this approach using a novel XREMIL environment to train participants to play specific notes on a theremin. Through this research, I aim to gain an understanding of how well learning in an XREMIL with RTVF transfers to the real world. Additionally, I aim to understand the limitations, challenges, and benefits associated with XREMIL.

XR training has shown success in a number of fields (see Section 2.3.1) which may inform the design of new XREMIL environments, but the design challenges posed by musical tutoring are significantly different then those of other fields. To this end, I pose the following research question:

RQ_3 What are the affordances and guidelines for designing effective and engaging XREMIL?

To address this question, I follow the interaction design process for the development of an XREMIL environment for learning the theremin (used in the previously described user study). Understanding the affordances of XR for XREMIL and developing design guidelines will facilitate the design and

development of learning environments that are effective as well as engaging for the user.

1.3.2 New Interfaces for Musical Expression

In the *Art of Noise: A Futurist Manifesto* (Russolo, 1913), Luigi Russolo suggested that "we must replace the limited variety of timbres of orchestral instruments by the infinite variety of timbres of noises obtained through special mechanisms." Russolo's manifesto motivated a number of musicians to experiment with new methods and interfaces for producing and manipulating sound. At the time, however, most of the methods were based on mechanical or analog technology. Little did Russolo know how the computer would expand the possibilities of sounds that could be produced with a single device. Since the beginning of the digital age, however, scientists and artists have seen this potential. In fact, the first recording of computer music can be attributed to Alan Turing nearly 70 ago (Copeland and Long, 2016). Since then a lot has changed. New digital synthesis and physical modeling techniques provided tools for programmers and musicians to generate and control infinite timbres of sound in real time (Cook, 2002; Smith, 2010), leading the way for a new field of research, New Interfaces for Musical Expression (NIME). NIME research is focused on applying and developing new technologies to enhance musical performance and overcome the limits of traditional instruments as described by Russolo (1913). XR has the potential to create new modes of musical interaction not previously possible. My research explores this space through the design and development of **virtual environments for musical expression (VEME)**.

Similar to CAMIT, NIME has had limited research exploring the application of XR and identifying design guidelines. Existing design knowledge from traditional NIME (Cook, 2001, 2009; Wanderley and Orio, 2002; Wanderley et al., 2016) can inform design of new environments, but XR presents new challenges and affordances for the design of novel VEME. Through my research, I aim to increase knowledge of the VEME design space by addressing the following questions:

RQ_4 What are the affordances and guidelines for designing effective and engaging VEME?

RQ_5 How can the development and prototyping of VEME be made more accessible to the NIME community?

With limited previous work to build on, addressing RQ_4 requires the development of various categories of VEME. A major challenge in VEME development, however, is the lack of a workflow for supporting rapid prototyping that easily integrates sound design tools with the XR environment design workflow. To address RQ_4 and RQ_5, I identify design needs and implement a novel XR toolkit to enhance the VEME development workflow; thus, enabling designers to more easily explore the design space.

1.4 Research Contributions

By addressing the research questions posed earlier with applied and basic research, this thesis provides several contributions to the fields of XR, CAMIT and NIME. This section highlights the main contributions to provide SMC researchers with new tools and knowledge to facilitate research efforts for the design of novel and innovative MusE-XR.

Addressing limitations in the current CAMIT literature, I contribute research informing the design and implementation of CAMIT systems. First, research to address RQ_1 results in a novel CAMIT system to automatically assess pianist hand posture using commodity sensor technologies. The outcome of this work demonstrates the viability of such a system using depth data from commodity depth cameras. The work contributes details on the application of existing computer vision (CV) and machine learning (ML) techniques to extract hands from a depth map that contains a scene in which the hands are in direct contact with another object (i.e. the piano). Furthermore, the work demonstrates that ML models trained with standard CV image descriptors are successful in hand posture detection with depth data from the extracted hands. This approach enables individually customized hand posture assessment models with limited training data. This work is further discussed in Chapter 4.

The second major contribution of this thesis to the CAMIT literature is the results of a user study conducted to answer RQ_2 and RQ_3. The user study evaluates the effectiveness of RTVF for learning notes on the

theremin. By performing the first known experiment for XREMIL effectiveness, the user study provides significant contributions toward understanding the challenges and affordances of employing RTVF in XREMIL. The results of the study indicate that providing RTVF may hinder a students' improvement by increasing the cognitive demands required for practice. The XREMIL environment, however, lead to more accurate performances during training and increased participant engagement and confidence. In addition to the results of the user study, this work contributes a novel XREMIL environment for the theremin as well as newly identified affordances and guidelines for the design of these environments. Full details of the study are described in Chapter 3.

The third contribution presented in this thesis come from addressing RQ_4 and RQ_5. Research on the affordances and guidelines for VEME design led to an open source XR toolkit, called OSC-XR, which integrates Open Sound Control (OSC) for rapidly prototyping VEME. OSC-XR is freely available to the NIME community and aims to enable research through improved VEME design workflows by simplifying the integration of OSC into VE development. Additionally, three new VEME, developed using OSC-XR, are presented along with identified affordances and guidelines identified through the design experience. Full details on the toolkit and th novel VEME are discussed in Chapter 6.

To summarize, the major contributions of this thesis are:

C_1 a novel CAMIT system for the automatic assessment of hand posture using CV and ML methods with commodity depth camera data,

C_2 the results of the first known user study on the effectiveness of RTVF for XREMIL, and

C_3 OSC-XR, an open source XR toolkit integrating OSC into the development workflow to facilitate the development and research of MusE-XR.

1.4.1 Publications

The previously discussed research contributions led to the following publications.

[1] D. Johnson, D. Damian, and G. Tzanetakis. Evaluating the Effectiveness of Mixed Reality Music Instrument Learning with the Theremin. *Virtual Reality*, July 2019.

[2] D. Johnson, D. Damian, and G. Tzanetakis. OSC-XR: A Toolkit for Extended Reality Immersive Music Interfaces. In *Proceedings of the 2019 Sound and Music Computing Conference*, May 2019.

[3] D. Johnson, D. Damian, and G. Tzanetakis. Detecting Hand Posture in Piano Playing Using Depth Data. *Computer Music Journal*, To Appear 2019.

[4] D. Johnson, I. Dufour, G. Tzanetakis, and D. Damian. Detecting Pianist Hand Posture Mistakes for Virtual Piano Tutoring. In *Proceedings of the International Computer Music Conference*, pages 168-171, 2016.

[5] D. Johnson, and G. Tzanetakis. VRmin: Using Mixed Reality to Augment the Theremin for Musical Tutoring. In *Proceedings of the 2017 Conference on New Interfaces for Musical Expression*, pages 151-156, 2017.

1.5 Thesis Outline

Chapter 2: In this chapter, I present background information on the three major research topics discussed throughout this thesis. First, the field of CAMIT is presented in terms of the challenges of musical learning. Then I introduce the field of NIME, including three main areas that have inspired and influenced my research: digital music instruments, hyperinstruments, and the open sound control protocol. Third, I cover XR research as applied to training and SMC. I close the chapter by summarizing and synthesizing the challenges and influences to this from these three fields.

Chapter 3: In this chapter, the methodologies employed to achieve the goals of this thesis are describe. I then discuss the history of my research and how my early research led to the two research tracks explored through this dissertation. I go on to describe how the conceptualization of the research questions from each track. Finally, the specific methodologies used with each research project are presented.

Chapter 4: In this chapter, I investigate the application of existing technologies to design a CAMIT system for the automatic assessment of pianist hand posture using depth camera data. First, I discuss my proposed approach for implementing the system. Next, the development of a prototype system is presented to explore the viability of the proposed approach. After validating the approach, I present a modified approach addressing limitations that were identified during prototyping. Then, I discuss the development of the assessment model using a real world data set. I close the chapter with a short discussion on the challenges of the system as well as ideas on the design of interfaces for presenting assessment results.

Chapter 5: In this chapter, the effectiveness of RTVF with XREMIL is evaluated. First, I discuss the design and evaluation of a novel system for teaching students to play notes on the theremin, leading to a few design guidelines for XREMIL. Next, a user study is administered for evaluating the effectiveness of RTVF with this system. Then, I present the results of performing data analysis on the objective and subjective data obtained through the study. Finally, I present my thoughts on the implications the results have on XREMIL design.

Chapter 6: In this chapter, An XR toolkit, called OSC-XR, is presented

to facilitate the development of VEME. First, I describe the implementation details and instructions for using the publicly available API. Then, I cover how OSC-XR can be used to enable rapid prototyping for VEME. I evaluate the toolkit by implementing three use cases in the design of different categories of VEME. I close with a discussion on the affordances and design guidelines learned through the use cases.

Chapter 7: In this chapter, the work presented in this thesis is summarized. I then discuss design considerations for applying extended reality (XR) technologies to SMC based on the experience of designing XREMIL and VEME throughout this thesis. Finally, I provide my thoughts on future directions for the research of MusE-XR.

Chapter 2

Background

The interest in using computing technologies to create music goes back to the emergence of computers themselves when Alan Turing discovered he could play musical notes on an early computer (Lewis, 2016). Since then scientists and musicians have continued to find ways to employ the newest technologies for novel musical experiences. This has lead to a field of research called Sound and Music Computing (SMC). Computer assisted musical instrument tutoring (CAMIT) and New Interfaces for Musical Expression (NIME) are two areas of research within SMC that explore the application of the latest technologies to enhance musical learning and performance, respectively. The emergence of XR affords new opportunities and capabilities to enhance musical learning and performance new musical experiences in extended reality (MusE-XR).

This chapter highlights the significant literature relating to the three main threads of the research in this thesis: CAMIT, NIME, and XR. The chapter starts with a literature review of CAMIT research in the context of the music learning process. I then describe the field of NIME including the three main areas of influence on this work: digital music instruments, hyperinstruments and Open Sound Control (OSC). Next, I introduce the concept of XR and its application to the areas of training and music. I close the chapter with a discussion on the limitations of the current state of the art in MusE-XR.

2.1 Computer Assisted Musical Instrument Tutoring

This section discusses state-of-the-art CAMIT research presenting techniques that enhance the music learning process. In Section 2.1.1, I provide a high level overview of the music learning process and three supporting areas: music lessons, music practice, and motivation. I discuss how CAMIT systems have provided tools to enhance each of these areas. A major focus of CAMIT research that supports all three areas is the development of new computational techniques to automatically assess a students musical performance. Section 2.1.2 provides a description of the CAMIT research that proposes new techniques for assessing musical performance, including both the quality of the musical output and the quality of the physical technique playing the instrument. I also discuss state-of-the-art research for tracking pianists' hands that can be used for assessment of pianist hand technique, in support of RQ_1. Due to the high cognitive demands of learning music, presenting the results of automatic assessment requires carefully designed feedback mechanisms, in Section 2.1.3 I discuss offline and real-time methods for presenting feedback to students. Emerging XR technologies afford new methods for integrating RTVF in the learning process, Section 2.3.1 discusses emerging research in XREMIL, in support of RQ_2 and RQ_3. Finally, I conclude the section with a discussion of the gaps and limitations of the current state of the art in CAMIT research.

2.1.1 Music Learning

The musical learning process typically consists of a student receiving either a lesson from a professional teacher, in a one-on-one or group setting, or a lesson using self-teaching resources such as music books and Internet tools, such as YouTube. After the lesson, the student is expected to practice what they learned during their lesson. With teacher led training, the student performs for the teacher, after a week or two of practice, to show their progress and receive feedback about their performance. The teacher then decides if the student should continue practicing the previous material or if they should move on to a new lesson. On the other hand, with self-teaching a user never receives professional feedback and must rely on

their own judgment to determine when to move on to a new lesson. The lesson-practice cycle continues until "students build independence, aural discrimination, and the ability to plan and evaluate their own practicing, at some point becoming their own teachers" (Kostka, 2004).

This is a lofty goal as there are a number of challenges with the music learning process. The cost of lessons can make private lessons inaccessible. Group lessons are not tailored to a specific individual and there may be little time for individualized feedback. When self-training a student does not receive professional feedback on their performance. Furthermore, all three of these learning methods require students to practice effectively during their time away from an instructor, but students may not know proper methods for effective practice. Finally, motivating students to practice on a daily basis is a challenge of its own. Percival et al. (2007) discuss three main areas of music pedagogy to categorize these challenges: lessons, practice, and motivation. Research in CAMIT systems has the potential to address each of these challenges.

Music lessons can be expensive and students often look to different methods to teach themselves how to play an instrument. In this past, this may have required purchasing a music book or two and creating self organized lesson plans. The emergence of computers and the Internet have opened the door for alternative methods for engaging the music learning process. The recent concept of Massively Open Online Courses (MOOCs) has led to a number online courses and video tutorials to replace standard music lessons (Berklee, 2019; Udemy, 2019). Additionally, market ready tutoring applications, such as Yousician (2019) and Skoove (2019), provide users with pre-designed lesson plans that allow users to learn at their own pace. Online courses and tutoring applications, however, currently lack the capabilities for individualized lessons that take students' abilities, or lack of, into account when developing the lesson plans. Research in CAMIT has resulted in systems that are able to automate the selection of practice tasks based on evaluation of student performance. The Piano Tutor project (Dannenberg et al., 1993, 1990) used an expert system to tailor lesson plans based on assessment of the students skills. The Piano Tutor provided students with practice tasks that were selected to improve assessed weak points in a student's learning. Similarly, Kitamura and Miura (2006)

developed a system for self-learning the piano with the intention of replacing expert instruction. The system employed existing pedagogy methods from common music learning texts. Using these methods, the system was able to observe weak points in a students practice and automatically generate practice tasks using curriculum from the texts. The IMUTUS project, a CAMIT for teaching the recorder, took a simpler approach. Instead of selecting tasks to improve specific weak points, students unlocked lessons when they succeeded in meeting prerequisite skills. In addition to generating lessons plans and practice tasks, teachers often demonstrate performance techniques to students as part of the weekly lesson. To this end, Lin and Liu (2006) presented an intelligent piano tutor that was able to demonstrate to a student the correct fingering of a score using a 3D virtual pianist. These systems have the potential to change the way a student approaches learning music and taking lessons. Improved lessons facilitate learning, but practice is still required to become a better musician. Students must have the motivation to practice on a regular basis to significantly improve.

Getting students to practice regularly is common challenge in music education, especially when students find the work frustrating or challenging. It is often the case, that students are propelled to learn an instrument because of some extrinsic motivation, such as a parental requirement or a child's desire for increased social status. Whatever a student's extrinsic motivation for learning an instrument, intrinsic motivation is needed to sustain and enjoy practice (Csikszentmihalyi et al., 2014b). Without proper intrinsic motivations, students may be quick to quit a practice task if they find it too challenging or frustrating. Csikszentmihalyi et al. (2014a) suggest that individuals becoming intrinsically motivated when they are able to reach a state of *flow*, a state in which the individual is absorbed into the activity they are performing. The authors outline three preconditions for achieving flow:

1. the activity contains a clear sets of goals,
2. there is a balance in the perceived challenges of the activity and perceived skills of the individual,
3. and, finally, there should be "clear and immediate feedback".

CAMIT systems may support these preconditions by providing students with a clear curriculum, personalizing the curriculum based on the student's (perceived) skills, and by designing thoughtful methods for feedback.

There are a number of CAMIT systems that aim to address the issue of motivation in music pedagogy. While not explicitly stating the concept of flow, many of these systems address one or more of the preconditions. CAMIT systems described in the previous paragraph (Dannenberg et al., 1993, 1990; Kitamura and Miura, 2006; Yousician, 2019; Skoove, 2019) indirectly support flow by providing students with specific practice activities; thus, defining clear goals for the students. Fukuya et al. (2013) considered student motivation as the core factor in their piano tutoring system. The authors developed a piano tutoring system that kept students motivated by decreasing the perceived challenges of practice for beginning students and allowed the students to select a learning method that corresponds with their own skills. The system implemented two methods for reducing the perceived challenge of practicing a musical piece. First, the system projected keying information directly onto the keyboard, making the activity of reading a score easier for beginning students. Second, the system, made it easier to play a complex piece by correcting for keying errors. When a student keyed an incorrect note, the system would output the correct note if the error was within a specific error margin. Students were also able to select from multiple learning methods, each with different error margins, that corresponded to their skill level thus balancing the difficulty of the task with the student's skills. With the knowledge that flow is often achieved while playing video games, it is possible that the gamification of music tutoring can improve motivation. Jaime et al. (2016) expand on this idea by presenting a music tutoring system that gamified the music learning process using concepts from rhythm games, such as Guitar Hero and Rock Band. The interface of the tutoring system mimicked the design of these but took steps to address problems that limit their training abilities. Gamification has potential to keep students motivated but the authors do not study the effects of their game on student motivation. Another way to make music learning more fun for a student is to enable collaboration through duets or other techniques. The Family Ensemble (FE) system from Oshima et al. (2007) supported student motivation in this way by making it

easier for parents, even with limited musical skills, to perform a duet with the student. FE allowed a parent to join their student through a system that used note-replacement techniques to correct a parent's performance to match the duet. While CAMIT research that focuses on improving motivation is relatively limited, most CAMIT systems indirectly support motivation by enhancing the practice process with techniques for providing guidance and feedback (discussed in section 2.1.2) to improve a student's confidence that they are practicing correctly. Getting students to practice their instruments is a challenge in an age of distraction but with proper motivation, facilitated by CAMIT, students are likely to practice more. Research presented in this thesis investigates novels methods that facilitate flow and motivation by enhancing musical practice thorgh automated performance assessment and real-time visual feedback (RTVF). More practice, however, doesn't necessarily mean they will practice more effectively.

It is a common belief that increasing the amount time practicing will lead to better performance, but Kostka (2002) suggests this is not necessarily true. Instead the author suggests that practice is more effective with more deliberate methods for practicing, such as focusing on specific tasks. Deliberate practice according to Ericsson et al. (1993) consists of the activities that teachers and experts have found to be the most effective in increasing performance and should be tailored to each students needs. To promote deliberate practice, Kostka (2004) suggests that teachers should work more on teaching effective practice techniques during their scheduled lessons. To improve students' practice time the author suggests that teachers

- teach students how to practice,
- use an aural model to allow the student to hear the correct sounds,
- select music that is interesting to the student,
- teach creativity,
- and teach students how to self evaluate their practice sessions.

These techniques may provide students with a framework for effective practice but without a teacher present it is challenging to ensure students,

especially beginning students, are practicing as the teacher expects. Similarly, Ericsson et al. (1993) suggests that teachers should design individualized training activities and explicitly instruct students on how to train in between meetings. Music teachers and students generally agree on the importance of deliberate practice but in many cases teachers assume that students learn effective practice techniques during their lessons and apply them afterwards during practice. A survey of college students, however, indicates that a majority of the students are not practicing regularly and effectively (Kostka, 2002).

CAMIT systems enhance musical practice by providing tools to support deliberate practice as well as tools to ensure students practice correctly when a teacher is not available. Generally this is accomplished by enhancing students' abilities to self-evaluate with tools enabling the automatic assessment of musical performance. Taking Kostka's 2004 advice, my research supports deliberate practice by investigating new methods to assess performance and provide feedback with the intention of improving students' abilities to self evaluate.

2.1.2 Automatic Assessment of Musical Performance

As previously discussed self-evaluation is an important skill to improve musical abilities. This skill is not well developed in beginning students and with the help of a teacher, they must gradually learn to evaluate their performance to identify and correct errors. Students, however, often forget or do not understand what was taught during the lesson. If students do not learn to properly self-evaluate they may practice incorrectly until they next lesson when a teacher corrects their mistakes. This requires the student to go back and relearn what they have been practicing. This repetitive practice can be frustrating for a student. Additionally students attempting to teach themselves, may never catch their mistakes and learn to play incorrectly. To help improve self-evaluation CAMIT systems employ computational techniques to automatically assess performance when a teacher is not present. Performance of musical instruments has two main quality attributes that students must evaluate during performance: the quality of the musical output and the quality of their physical playing technique. In

general, errors can be categorized into musical mistakes, such as missed notes or poor sound quality, and technique mistakes, such as poor posture. To enhance students' practice time, is is import to effectively present feedback to the student based on the results of the automatic assessment of their performance. The rest of this section, discusses the techniques used by CAMIT systems to automatically assess musical quality and technique.

Automatic Assessment of Musical Quality

Assessment of musical quality deals with the aural component of a musical performance; most commonly, this means playing the correct note at the correct time. With some instruments, such as the stringed instruments, the timbral quality of the sound is just as important. Most CAMIT systems that assess musical quality have focused on assessing pitch and timing errors. This is usually achieved by listening to the performance with audio signal processing (ASP) and then comparing the performance with a ground truth score to identify errors. One of the first CAMIT research projects to use ASP was the Piano Tutor Project (Dannenberg et al., 1993), an intelligent multimedia system to teach beginners to play the piano. The Piano Tutor was a complete tutorial system intended to supplement traditional musical pedagogy with a professional teacher. Using ASP the Piano Tutor implemented score following to assess how a student was performing by listening to the student's performance and comparing it with a score (Dannenberg et al., 1990). The IMUTUS project (Raptis et al., 2005) was a music tutoring system for teaching the recorder to beginning students. Similar to the Piano Tutor, IMUTUS listened to students' performances using ASP for audio recognition to assess the musical output. Audio recognition was integrated with score matching to detect errors in the performance. By listening to a performance, the IMUTUS interface was able to detect melodic, timing and articulation errors (Schoonderwaldt et al., 2005). iDVT (Lu et al., 2008) was a system for violin tutoring that transcribed a student's performance through onset detection and pitch estimation. To improve the quality of onset detection ASP was fused with video data. A student could then compare the transcribed performance to a reference score. Melodic correctness is not the only aural attribute when assessing the musical quality of a per-

formance, timbral quality is just as important for some instruments. In addition to listing for melodic errors, the IMUTUS system listens for sound quality issues that show lack of instrument control (Schoonderwaldt et al., 2005). Research performed with the TELMI project used ASP and ML methods to analyze violin performance but rather than focus on pitch and onset errors, the system assessed tone quality (Giraldo et al., 2019). Their system implements methods to train student specific tone quality models to overcome the subjectivity in timbre perception that makes generalization a challenge. Similarly, the CAMIT system presented in Chapter 4 of this thesis employs customizable student specific assessment models but for piano playing technique. ASP plays an important role in the automatic assessment of musical performances but can only assess the musical quality of the performance; other methods are needed to assess performer playing technique.

Automatic Assessment of Playing Technique

Figure 2.1: The AMIR marker-based motion capture system for violin technique assessment (Ng et al., 2007).

Assessment of playing technique requires watching physical characteristics of the performer to identify poor form during a performance. Tech-

nique errors that teachers often watch for include posture related errors, such as poor hand or body posture, as well as problems with performance gestures, such as bowing technique. Automatic assessment of playing technique requires methods and systems to capture body positioning and movements during practice. CAMIT researchers have employed optical systems, such as optical motion capture and camera technologies, to capture the needed performance data. For piano pedagogy, Mora et al. (2006) employed a motion capture system to track the movements and body posture of a pianist. The system used eight infrared cameras and an average of 79 positional markers to record positional data to construct a 3D skeleton model which could be overlaid on video recording of the practice session. The i-Maestro project (Ng et al., 2007) used a motion capture system to capture and analyze the performance of stringed instruments for the 3D augmented mirror (AMIR) application. AMIR used twelve infrared cameras and markers attached to the performer, the bow, and the instrument to capture performer and instrument positional data, see Figure 2.1. The data was used to provide assessment and feedback on the performer's bowing technique and posture. Motion capture systems, however, are complicated and expensive, limiting their use outside of laboratory settings. Figure 2.1 demonstrates the complexity of using a marker-based approach which could also be intrusive to instrument playing. Thus, more accessible methods, such as computer vision or signal processing with low cost sensors, are needed to capture motion for technique assessment for practical or at-home settings. Dalmazzo and Ramirez (2019) used the Myo armband, which tracks muscle movement in the forearm using electromyography (EMG), for the classification of violin bowing gestures. The Myo data was combined with audio data for real-time gesture recognition using a Hierarchical Hidden Markov Model (HHMM). Salgian and Vickerman (2016) proposed a computer vision (CV) based CAMIT system for conducting students that used a Kinect depth camera to track students' physical conducting performance. Using depth data, the system was able to detect common conducting errors, calculate tempo and perform articulation recognition (the Conducting Tutor interface is shown in Figure 2.2). Similar to Salgian and Vickerman (2016), I use CV methods with depth data to assess playing technique for piano players as discussed in Chapter 4. These works show that assess-

ment of playing technique is an important component to music pedagogy and can be integrated in CAMIT systems using technologies such as motion capture, CV and signal processing.

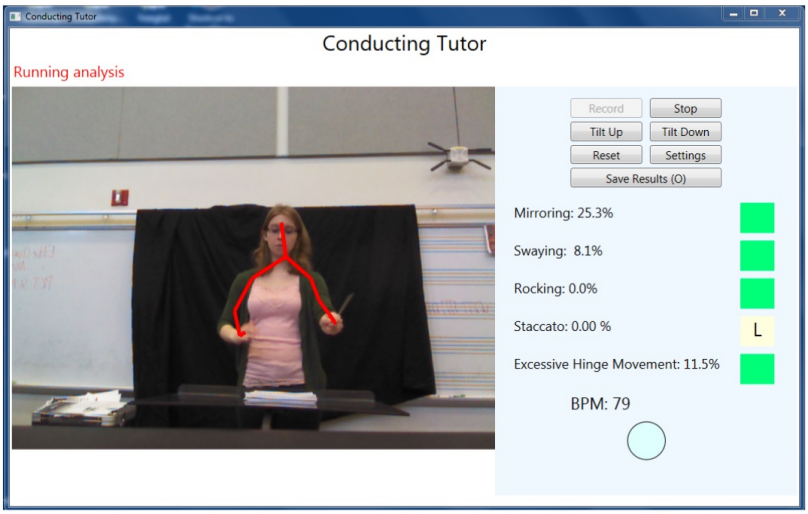


Figure 2.2: The Conducting Tutor interface with body tracking implemented using the Kinect (Salgian and Vickerman, 2016).

Pianist Hand Tracking

The research presented in Chapter 4 requires methods for tracking pianists' hands during performance to address RQ_1 regarding the assessment of pianist hand posture. Capturing pianists' hands for piano playing technique assessment presents a number of challenges making it an interesting problem. To name a few, there are variations in size, shape and color making it difficult to create generalized models, performance requires precise motor skills requiring high resolution, and the hand is interacting with another object (i.e. the piano) making it difficult to track at a granular level. There has been some previous research attempting to solve the problem of pianist hand performance assessment. Tits et al. (2015) used a marker based motion capture system to analyze pianists' hands and finger gestures to determine the performer's level of expertise. Their system employed 12 infrared cameras for tracking 27 reflective markers that were placed on each hand. Using a marker based approach affords techniques for obtaining precise hand data but are generally intrusive, expensive and not readily

available to non-researchers. As an alternative, markerless approaches for hand tracking use standard RGB cameras or depth data from depth cameras. Hadjakos et al. (2009) presented three methods for using RGB video to detect which hand played a note, and Oka and Hashimoto (2013) used a combination of depth maps from a Kinect and Musical Instrument Digital Interface (MIDI) data to identify pianists' fingering mistakes. A depth camera was employed by Hadjakos (2012) to capture the motion of key points from a pianist's entire body, such as head, shoulders, wrists and hands. Liang et al. (2016) used a depth camera and ML methods to detect individual finger tapping for playing a virtual piano, but in this case, the hand was interacting with a flat surface not the piano. Additionally, the camera was placed very close to the hand making it impractical for at-home practice environments. These works all provided techniques for hand tracking, but none provided techniques to capture the data needed for the assessment of hand posture in a real-world practice setting.

There were two studies that presented more practical systems for hand tracking in piano playing. MacRitchie and McPherson (2015) developed a more practical system for automatic fingering detection that fused data from a high speed RGB camera and touch sensors. A camera placed at an aerial viewpoint tracked painted markers on the pianist's hands to capture the XY coordinates of each finger. While the data was only 2D, the coordinates were used to calculate a curvature index, CI. CI was calculated as the ratio between the distance of two points at a given time with distance of the same two points in a reference frame. While this provides relative information about the curvature of each finger, there is not enough information to fully discriminate between various categories of hand posture as needed for hand posture assessment proposed in Chapter 4. In another more practical setup, Li et al. (2014) proposed a system for pianist hand posture analysis that detected key regions of the hand using depth data. They used CV with depth cameras to find regions of the hand, such as the hand center, the middle finger, and the wrist. The key points were used to derive features for analysis: the hand center height to hand arch height ratio and the horizontal and vertical wrist angles. Using these features, a histogram analysis was performed for assessing the range of hand motion during a specific piano piece. The histograms used for analysis were

generated from data over the entirety of the performed piano piece rather than for real-time classification of posture mistakes. Furthermore, the authors provide little information about precisely how to replicate their hand tracking algorithms or a discussion about it's accuracy. While these systems provided more precise methods for pianist hand tracking, the methods meet the needs of a system for real-time assessment of pianist hand posture. Similar to Liang et al. (2016), my proposed hand posture assessment system places the depth camera above the hands for an overhead view, but, like MacRitchie and McPherson (2015), places it above an actual piano to capture both of a pianist's hands while they are playing. Further, like Li et al. (2014) I use CV methods to extract features of the hand from depth maps but rather than extracting specific regions, I extract general image descriptors from a depth map containing just a hand that has been segmented from the original depth map. Moreover, I use ML methods with the image descriptors to detect hand posture from single depth maps rather than performing a posture analysis over the entirety of a performance.

2.1.3 Interfaces for Enhancing Musical Practice

There are two general approaches employed in CAMIT interfaces to enhance student performance during musical practice. The first approach uses real-time visual cues to provide the students guidance about how to play. For example, a CAMIT system may overlay keys on a virtual keyboard with visual signifiers indicating which key should be played and by which finger, similar to the approach taken by Yousician, see Figure 2.3. Another approach is to provide students with feedback based on the results of performance assessment. Designers of interfaces for CAMIT systems should take each of these approaches, including their benefits and drawbacks, into consideration.

Realtime Guidance

One method used by teachers during a music lesson to teach a student how to perform a specific practice exercise is to simply demonstrate the correct performance to the student. The teacher may ask the student to watch them perform as they demonstrate or the teacher may play along in

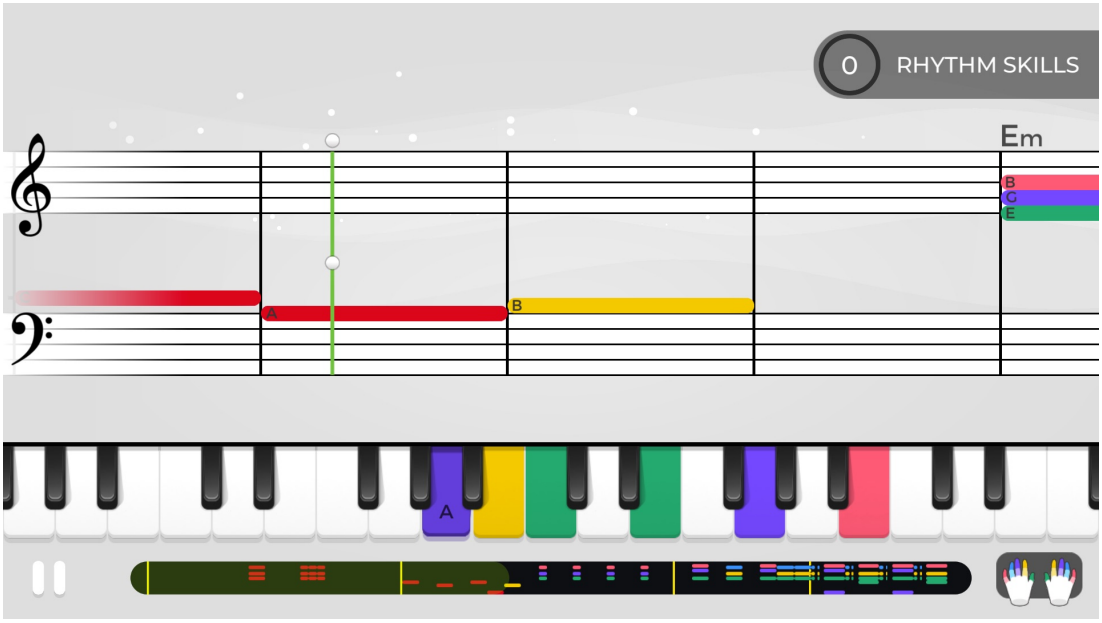


Figure 2.3: The Yousician piano lesson interface. The colored notes in the score map to the colored keys on the piano to teach students which keys to press.

real time with the student, allowing the student to mimic what the teacher is doing. Emerging technologies afford new techniques to demonstrate a performance without a teacher being present.

Many CAMIT systems designed for performance guidance provide real-time visual cues for students to follow along as they are playing. This is one of the approaches taken by of market ready tutoring systems (Yousician, 2019; Skoove, 2019). For example, Yousician piano lessons, shown in Figure 2.3, overlay virtual piano keys with color coded visual cues to indicate when and where a student should place their fingers, making the process of reading a score easier for the student. A challenge with the 2D approach is that a student must cognitively map the visual cues on the 2D display to the physical keyboard and constantly shift focus between the display and the instrument. To overcome this problem, the MirrorFugue and Andante (Xiao and Ishii, 2016) use a projector placed over a piano to project recordings of a performer and animations directly onto the piano. Using these visual cues, the student is able to imitate the performer or follow the animations to learn a specific piece. Similar to the work of Xiao and Ishii (2016), the theremin XREMIL environment developed for my research,

called *MR:emin*, overlays the theremin space with visual cues to guide students to the correct notes at the correct time. Instead of using a projector, *MR:emin* uses XR technology to implement the visual cues. There is other emerging research that employs XR to overlay a physical instrument with visual cues (Huang et al., 2011; Chow et al., 2013; Das et al., 2017), discussed more in Section 2.3.1. Instead of generating real-time guidance, Kitamura and Miura (2006) proposed an offline system that used a 3D model of a hand virtually performing to show students correct fingering for a given piece. These works proposed various methods to provide students with performance guidance when a teacher is not present, but there is little work studying the effectiveness of such approaches.

Carefully designed visual guidance can enhance practice by helping improve students' confidence that they are performing correctly. Using offline guidance allows the student to view a correct playing technique, but requires that a student remembers the instruction while they are practicing. RTVF addresses this constraint by allowing a student to play alongside visual cues, but with this approach designers need to ensure that the student learns to actually play the instrument and not just follow visual cues (Percival et al., 2007). In other words, the skills learned in the training environment should transfer to a real world scenario. A lack of research studying the effectiveness of either approach is a clear gap in the knowledge of CAMIT research. I attempt to fill this gap with research performed in Chapter 5 to address RQ_2 and evaluate a XREMIL with RTVF.

Feedback Methods

Section 2.1.2 discussed computational techniques used for automatic assessment of a music performance to identify errors without a teacher present. The results of the assessment are generally intended to be presented through a CAMIT interface with the goal of helping students evaluate their performance. Feedback from the assessment can be presented to students either offline, when they are done with a performance, or in real-time, as they are performing.

Interfaces with offline feedback first monitor a student's performance during a practice task and then present feedback from the assessment af-

ter the task has been completed. The IMUTUS recorder tutoring system (Schoonderwaldt et al., 2005) used offline feedback to inform a student about the mistakes made while practicing. To limit the amount of information presented to the student the system selected the top three errors to present to the student. Additionally, the system presented the student with an overall quality score to provide the student with an quick impression of how well they performed. iDVT (Lu et al., 2008) simply presented students with a piano roll display of their performance alongside a ground truth score allowing the students to see how correctly they performed compared with an expert. The AMIR system (Ng et al., 2007) extends the common practice technique of recording a practice session by augmenting the recording of a violin practice session with assessment visualizations regarding the quality of a student's bowing technique. Blanco and Ramirez (2019) studied the effectiveness of offline feedback for violin learning. Beginning violin students were assessed on the quality of a sustained tone on the violin. The students were presented with a visualization to indicate the quality of the tone by extracting audio descriptors from the generated sound. The authors found that only students presented with the visualization continued to improve after the first set of training trials. They hypothesize that the feedback presented allowed the students to experiment with new ways to generate sounds. Offline feedback can be helpful, but when an assessment system identifies specific errors, students may find it difficult to correlate feedback with the specific occurrences of mistakes.

To overcome this limitation, interfaces with real-time feedback immediately present students feedback about their performance. Systems for real-time feedback are less researched in the literature. The Yousician interface (Yousician, 2019) provides students with real-time feedback indicating when timing and pitch errors are made. The Violin Timbre Navigator (Perez-Carrillo, 2019) was a system for analyzing bowing information that provided the users with real-time feedback. Their work included the Violin Palette which was a visualization providing a student with details about their bowing technique. The visualization was fairly complicated and may be too cognitively demanding for use in real-time settings. Without an evaluation of the interface, it's not clear how effective the system was for improving bowing technique. Instead of using a visual display to present

feedback, Ferguson (2006) used sonification to provide musicians with aural feedback for their performance. The system was able to identify errors in timing, pitch, and loudness and would play auditory cues when errors were identified. The *MR:emin* learning environment presented in this thesis provides feedback most similar to the Yousician (2019) by simply providing a visual indicator to inform the student if they are performing correctly or not. Real-time feedback has the potential to improve musical practice by helping students self evaluate and address errors as they play. Because music practice is already cognitively demanding without the added multimodal feedback, designers need to carefully evaluate the effectiveness of various techniques. More research is needed in this field to better understand how each method affects the learning process. To address the question of the effectiveness of RTVF method for instrument learning, I perform a user study, discussed in Chapter 5, using the *MR:emin* XREMIL environment.

2.2 New Interfaces for Musical Expression

The field of NIME covers a wide breadth of topics related to creating and performing music, but in this section I focus on three topics that have guided the research presented in this thesis. Section 2.2.1 provides a brief introduction to digital music instruments (DMIs) and a discussion on the design of control interfaces. The concept of hyperinstruments is introduced in Section 2.2.2. Finally, in Section 2.2.3 I discuss OSC, an important NIME interface protocol.

2.2.1 Digital Music Instruments

A DMI is an "instrument that uses computer-generated sound and consists of a control surface or gestural controller, which drive the musical parameters of a sound synthesizer in real time" (Miranda and Wanderley, 2006). DMIs differ from physical musical instruments in the degree of coupling between physical control and sound generation. With traditional instruments, the controller and the sound generation source are the same, i.e. the physical instrument, resulting in direct coupling between the control

and sound. On the other hand, a DMI controller typically has no inherent sound generation capabilities and must rely on a separate sound generation source, i.e. the computer, resulting in a decoupling between control and sound. The decoupling of sound generation from control actions requires designers to carefully consider the control-to-sound mappings for their design (Hunt et al., 2003).

This definition divides DMIs into three distinct but tightly bound categories for research; 1) control interfaces, 2) mapping strategies, and 3) sound generation. Each of these categories has independent research attempting to solve low level challenges. However, designers of DMIs should take each into consideration. Work performed in this dissertation, discussed in Chapter 6, is focused the control interface component.

Control interfaces for DMIs can be anything from GUI applications using standard HCI metaphors, such as direct manipulation, to everyday objects like a coffee mug, as in Perry Cook's JavaMug (Cook, 2001). Miranda and Wanderley (2006) classify control interfaces into four categories based on the degree of similarity to acoustic instruments,

- *Augmented Musical Instruments*, acoustic instruments augmented by sensors to provide additional musical control,
- *Instrument Like Controllers*, controllers that attempt to reproduce the features of an acoustic instrument,
- *Instrument Inspired Controllers*, controllers inspired by acoustic instruments that try to overcome some limitations of the acoustic instrument,
- *Alternate Controllers*, all other controllers that don't have a resemblance to an acoustic instrument.

This wide breadth of possibilities presents many challenges and considerations for designing innovative and expressive DMI controllers.

To help constrain the design space, there has been a substantial research on design guidelines and principles gained from experience as well as the integration of HCI interaction metaphors for DMI controller design. Cook (2001) presents a set of design principles for DMI controllers based on

years of experience designing and performing with DMIs; he later revisits the principles and adds a few new ones (Cook, 2009). Wessel et al. (2002) presented metaphors for musical control that were used to inform their design process for expressive musical gestures. Morreale et al. (2014) proposed MINUET (Musical Interfaces for User Experience Tracking) as a design framework for reducing the complexity of the controller design space. MINUET defines a conceptual model, modeled after the PACT (People, Activities, Contexts and Technologies) framework for interactive systems, to help designers better understand the elements involved in the DMI design process. Wanderley et al. (2016) discussed two models from HCI research for the design and evaluation of controllers; Rasmussen's (1986) *Human Information Processing* framework conceptualizing musical interaction possibilities and the *Instrumental Interaction* model of Beaudouin-Lafon (2000) to describe how users interact with objects as instruments (i.e. tools). The guidelines and metaphors presented in the previous works help to facilitate the design process of DMI controllers.

The design space for XR interfaces and environments brings about many new challenges not covered with typical HCI theories. There is emerging research on XR interface design, covered in Section 2.3, to facilitate the design process but it has not yet matured. Thus, this dissertation focuses on tools that enable rapid prototyping of MusE-XR to support iterative design allowing designers to quickly explore new design ideas.

2.2.2 Hyperinstruments

Just as the goal of DMIs is to enhance musical expressivity by designing new control interfaces the goal of hyperinstrument research is to enhance musical expressivity by extending the capabilities of existing instruments. Hyperinstruments, also known as augmented instruments, are traditional instruments that are extended by augmenting them with sensors to capture performance data. The performance data is then computationally processed in real-time to generate a new musical result (Machover, 1992). By adding computational data processing to a traditional music instrument, hyperinstruments enable performers the same level of control and sound generation afforded by DMIs that is typically not available with traditional

music instruments.

Early research in hyperinstruments by Paradiso and Gershenfeld (1997) studied the use of electric field sensing to track physical parameters of bows for stringed instruments. In this work, the authors built two hyperinstruments, the *Hypercello* and *Hyperviolin*. The bows of each instrument were augmented with sensors for tracking bow pressure, position and acceleration. The sensor data was used to detect musicians' gestures, as they performed, which were mapped to algorithmic parameters for extended musical output. Young (2002) extended the work to design the *Hyperbow* which included additional sensing for the downward and lateral strains on the bow. The sensor data afforded musicians new expressivity in their performance by allowing the violinist to alter the acoustic sound of the violin in real time using gestures performed with the bow. Typically, hyperinstruments added sensors to existing instruments. Overholt (2005) took augmenting the violin one step further with the *Overtone Violin*. Designed from the from the ground up, the *Overtone Violin* was built with specialized hardware and sensors directly embedded into the instrument. The previous hyperinstruments required physical modification of the musical instrument. The *Electrumpet* (Leeuw, 2009), on the other hand, was a device that could be mounted and removed from any trumpet. While the augmentation was not completely invasive to the original instrument it did require some technical expertise of microcontrollers to mount to the trumpet. The device augmented a trumpet with sensors and buttons for added control of the acoustic output as well as an LCD screen so the player could look at the trumpet and not a computer display. The previous works required invasive modifications that were expensive, hard to implement, and difficult to adopt by musicians.

The emergence of depth camera technology, such as the Kinect, has led to developments in non-invasive augmentation. Trail et al. (2012) augmented a xylophone with virtual faders by using a Kinect to track mallet tips, see Figure 2.4. The system turns the bars of the vibraphone into virtual faders that are controlled by the tracked mallets. Using sensors like the Kinect affords more accessible and affordable instrument augmentation since all that is needed is a Kinect and computer running the special middleware. Such as setup is limited, however, by the need for an addi-

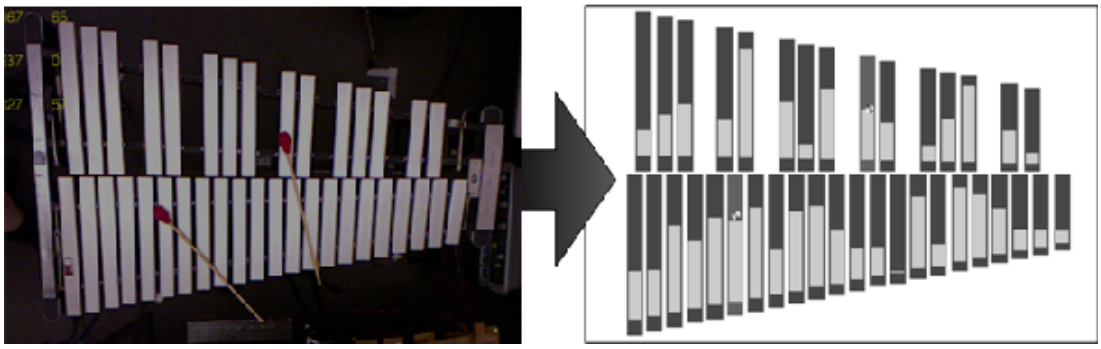


Figure 2.4: A xylophone hyperinstrument augmented with virtual faders (Trail et al., 2012).

tional computer display to view control data, such as the position of the faders in the previous example. Visual signifiers overlaid on the bars, using augmented reality (AR) for example, would allow the player to focus on the instrument, instead of an additional display, as they perform. XR enables non-invasive sensing techniques similar to the work of Trail et al. (2012), and it extends the idea by affording visual cues overlaid directly on the instrument being augmented. This has influenced the design of virtual hyperinstruments for XREMIL environments and VEME presented in this thesis.

2.2.3 Open Sound Control

NIME systems often have complex and distributed configurations requiring communication between multiple multimedia devices. In their simplest form, NIME consist of a controller and a sound synthesis engine. Decoupling the controller and sound engine affords separation of concerns but presents challenges in communication between devices with minimal latency. The OSC protocol (Wright and Freed, 1997) was developed to address these concerns and free NIME designers from the constraints of MIDI.

OSC is a networking protocol that enables communication of real-time streams of musical control data between distributed devices. The basic unit of data is the OSC message which contains an OSC address and a set of arguments (i.e parameter values). Messages are sent, via UDP or TCP, from a device configured as an OSC transmitter to one or more devices that are configured as OSC Receivers. An OSC receiver then routes the message

to a component or function as specified with the OSC address. This design provides lightweight and flexible communication between any number of devices with limited latency (Wright and Freed, 1997).

The NIME community has adopted OSC as a standard protocol for the development of systems with distributed communication needs. There are now OSC libraries for most programming languages and the inclusion of OSC is practically required for all audio programming environments. The OSC ecosystem has enabled numerous innovative NIME systems and more (Wright, 2005).

OSC Controllers

Researching the musical control of computers, Wessel and Wright (2002) discussed the affordances of using digitized tablets for musical control as well as potential mapping strategies for gestural control of music using these devices. The emergence of wireless networking and handheld devices has opened the door for a class of multi-touch OSC controllers that were inspired by this work.

TouchOSC Hexler (2019) is one of the most popular multi-touch controller applications specifically for mobile devices on iOS and Android. It provides users with prebuilt layouts using various control widgets, such as knobs, sliders and buttons. In addition to the set of existing control interfaces, users may also use the TouchOSC Editor to build their own interfaces from the standard library of widgets. The authors of two other multi-touch toolkits cited the influence of TouchOSC for their flexible and customizable control interfaces. Control (Roberts, 2011) was a mobile application that, similarly, let users design custom interfaces from a set of prebuilt widgets using JSON to define the interface structure. Control was set apart from from other interfaces by giving users the ability to add customized functions to their widgets using JavaScript. While TouchOSC and Control were developed for specific mobile platforms. Argos was an application and programming library for building multi-touch OSC control interfaces for custom devices (Diakopoulos and Kapur, 2010). Using Argos users were able to design control interfaces from a library of prebuilt widgets, similar to those of TouchOSC. Additionally, Argos provided developers a set of

C++ classes, built on openFrameworks, for creating their own widgets. The popularity of multi-touch OSC controllers, especially TouchOSC, show that OSC based applications with flexible design support needs of designers.

Flexible OSC controllers have also been designed for environments other than multi-touch devices. Hamilton (2011) developed a OSC library for the Unreal Development Kit (UDK) for use with VR environments, discussed more in Section 2.3.2. Further, Di Donato et al. (2018) presented an OSC library for mapping data from a Myo armband to OSC messages. Similar to multi-touch controllers, the system facilitated development through prebuilt components but instead of visual widget, the Myo Mapper implemented prebuilt feature extraction methods.

The OSC for XR toolkit presented in Chapter 6 was inspired by the underlying theme of flexible design presented in the previous works. Similar to these systems, OSC-XR implements prebuilt objects and customized scripting facilitating a simplified design workflow.

2.3 Extended Reality

The previous sections described two facets of music which XR may be greatly impacted with continued research: musical learning (CAMIT) and musical performance (NIME). Currently, however, there is limited work applying XR in these areas. There have been a few proof of concept XREMIL environments, but to my knowledge there have been no studies on the effectiveness of such environments. There is substantial research on the use of XR for training in other fields that has influenced my work. Section 2.3.1 covers the state of the art in XR based training as well as the few XREMIL systems that have been developed. Research on the application of VEME design has been limited to a few virtual reality music instrumentss (VRMIs) which are discussed in Section 2.3.2 along with audio synthesis techniques for VEME.

2.3.1 XR Training

Through virtual simulation or augmentation, XR has capabilities to enhance existing training techniques or to provide new ones in areas where

traditional methods are expensive, impractical, or are otherwise infeasible. Various fields have utilized these capabilities to improve training processes. Research has shown the potential for effective XR training environments in surgery and medicine (John et al., 2016; Lehmann et al., 2005; Cook et al., 2013), production assembly and maintenance (Borsci et al., 2016; Murcia-Lopez and Steed, 2018; Werrlich et al., 2018; Gonzalez-Franco et al., 2017), spatial navigation (Waller et al., 1998; John et al., 2018), neurorehabilitation (Adamovich et al., 2009; John et al., 2018), and sports (Miles et al., 2012), to name a few. The effectiveness of a given training environment is evaluated through an assessment of the learning transfer facilitated by the environment.

Learning transfer is the degree to which learning in one environment affects performance of another task (Cormier and Hagman, 1987). In XR, learning transfer deals with the amount in which training in a VE affects performance of the task in the real world. Early research on learning transfer theorized that the transfer of knowledge is a function of "*identical elements* in common between learning and transfer tasks" (Cormier and Hagman, 1987). One of the first known studies evaluating the learning transfer of an XR training environment indicated that there may not be positive transfer from immersive training environments to the real world. The authors attributed the lack of transfer to the state-of-the-art technology of the time (Kozak et al., 1993). Since then, however, there have been numerous studies that have shown positive results in XR training.

The focus of the studies is often on either the transfer of procedural knowledge or the transfer of psychomotor skills learned through the training environments. When training on procedural knowledge, the goal is to improve knowledge retention of a given task or process. For example, Gonzalez-Franco et al. (2017) implemented an XR environment for training the procedure of manufacturing an aircraft door. Trainees were expected to retain both general knowledge of the procedure, and be able to demonstrate the steps involved in manufacturing the door. On the other hand, there are a number XR training environments with the goal of improving trainees' psychomotor skills related to a given task, such as the work of John et al. (2016) in which the authors evaluated an XR environment for neurosurgery training. In their work, the participants were trained on the

ventriculostomy procedure and evaluated on the precision of their catheterizing results, in other words they were evaluated on how well they were able to control the needle used in the procedure. While the research in this thesis is focused on the transfer of psychomotor skills related to musical performance, procedural training transfer is also of importance to music pedagogy.

Research of the transfer of procedural knowledge is common in manufacturing training where the focus of training is on assembly tasks. Researchers often use burr puzzles as a proxy for actual assembly tasks when evaluating the transfer of learning in XR. Burr puzzles are a type of mechanical puzzle made of interlocking notched sticks which afford researchers a cognitively complex assembly task with simple instructions for completion (Oren et al., 2012). Carlson et al. (2015) compare XR training to physical training for burr puzzle assembly. While initial test results showed that XR training was outperformed by physical training in terms of puzzle assembly time, results of retention tests performed two weeks later indicate that XR training with additional color cues was just as effective as physical training (Carlson et al., 2015). Also using burr puzzles, Murcia-Lopez and Steed (2018) sought to understand the effects of XR training when haptic devices or physical objects are not available during the training process. They found that there was no significant difference between success rates for participants trained in XR with virtual Burr puzzle blocks and those trained by paper instructions with only physical blocks. The authors claimed this to be an important finding because the results provide evidence that learning transfer in XR without physical objects can may match that of traditional training with physical objects. While these results show positive results for learning training for assembly tasks, many in industry feel that the results may not transfer to real world situations (Werrlich et al., 2018).

To validate the learning transfer of XR in manufacturing training, researchers performed studies on industry specific assembly and maintenance tasks. Gonzalez-Franco et al. (2017) performed a study to evaluating the learning transfer of complex manufacturing training in XR compared to traditional face-to-face training. In their work, XR training was collaborative training where both the trainer and the trainee wore XR HMDs and

interacted with a virtual model of the manufacturing elements; whereas, face-to-face training employed a scaled physical model during the training session. According to Gonzalez-Franco et al. (2017) their results support the use of collaborative XR training for a complex manufacturing procedure. While these results are positive for observation based training, the benefits of XR lie with its abilities for experiential training. Borsci et al. (2016) compare two XR based experiential training environments with video based observational training. One XR environment employed a CAVE for immersive training and the other used a holographic 3D table. All groups received similar instructions about how to complete the maintenance process but the XR groups had additional training through performing trial procedures in the XR learning environments. Post tests revealed that both XR trained groups outperformed the video trained groups and there was no significant difference between the two XR groups (Borsci et al., 2016). While it is possible that the benefits simply arise from the additional experiential learning provided in the XR groups, real world experiential training may not always be feasible without XR. These works provide evidence to support the transfer of procedural knowledge from XR training to real world tasks but they do not address an important aspect of musical training, the learning of psychomotor skills.

In a general study on the learning transfer of psychomotor skills, Rose et al. (2000) evaluated XR for training hand steadiness. To evaluate hand steadiness, participants were required to move a metal ring attached to a rod along a curved wire without touching the ring to the wire. Performance on the task was judged by the number of times the participant touched the ring to the wire. Participants were trained by performing trials on either the physical device or a virtual model of the device with a hand held controller; additionally a control group received no training. The authors found that real world training and XR training resulted in equivalent performance on the steadiness test and both outperformed the group that received no training (Rose et al., 2000). This work provides strong evidence in support of positive learning transfer with XR training environments. Research in surgical training provides some practical results on the use of XR training for psychomotor skills. Lehmann et al. (2005) evaluated the learning transfer of psychomotor skills necessary for endoscopic surgery by compar-

ing a Virtual Endoscopic Surgery Trainer (VEST) with a Conventional Video Trainer (CVT) (a standard surgical training technique). The authors found that participants trained with VEST had comparable performance on the CVT as those trained only with the CVT. Thus, the skills learned in VEST successfully transferred to another device, the CVT. This study used a non immersive VE for training. More recently, Thomsen et al. (2017) investigated learning transfer with a stereoscopic training simulator for cataract surgery to the operating room. In this work, the authors conclude that surgical skills learned in the XR simulator improve performance in the operating room. The positive results of learning transfer for surgical tasks provide evidence that XR training has the ability to improve psychomotor skills.

These studies provide strong evidence of learning transfer in their specific disciplines, but music learning requires additional cognitive processing through auditory input. Thus, in addition to procedural knowledge and psychomotor skills, XREMIL should be evaluated in terms of auditory learning transfer. This means that the visual cues and other affordances of XR training need to be carefully considered so as not to take the focus away from the auditory learning required for music pedagogy. To the best of my knowledge there are no studies that evaluate the effectiveness of music training with XR training or the factors that affect the music learning process.

XREMIL Environments

Emerging research in XR has recently shown potential to use XR for various training tasks (Section 2.3.1 describes the research in more detail). The research into the use of XREMIL environments is limited but there have been a few studies that proposed using XR to assist with music pedagogy. Early work on XREMIL systems used non-immersive AR to add a visual layer to the learning process. This includes the work of Mora et al. (2006) in which they employ AR to assess body posture of a piano player as well as Liarokapis's (2005) work with non-immersive AR for for guitar learning. With XR technology improving, there has been an emergence of work employing immersive AR for music pedagogy. Chow et al. (2013) proposed

Figure 2.5: The Music Everywhere AR environment for piano tutoring (Das et al., 2017).

a head mounted AR system for teaching musical notation. The authors implement a Guitar Hero-like setup in which visual cues, overlaying the piano, flow down from the top of the display to indicate which notes are to be played. The interface provides the user with RTVF by color-coding the visual note cues to indicate the type of error. After a practice session, the student is provided with feedback on the correctness of their performance in a simple text-based view. The authors performed a small user study but only obtained subjective feedback about they system. The effectiveness of the system was not tested. Huang et al. (2011) also implemented an AR system for teaching piano, called Piano AR. Similar to work of Chow et al. (2013), Piano AR overlays visual cues, in this case virtual fingers, on the piano to indicate to a student what note should be played. The system was used to provide guidance on how to play a given piece but did not provide feedback to the student about their hand posture or playing technique. Similarly, Das et al. (2017) used AR to create interactive lessons by overlaying the piano with visual instruction placed perpendicular to the piano keys, shown in Figure 2.5. Most of the work for XR has been for piano pedagogy but Kweon et al. (2018) proposed an MR system that combines a virtual environment with a physical drum kit for drum training. These works all demonstrate proof of concept ideas for XREMIL but do not study the effectiveness of such approaches.

The previously discussed research shows that XR affords environments

in which visual cues are directly overlaid on musical instruments being learned; therefore, eliminating the transfer function required when trying to map visualizations display on a 2D display to the physical instrument. Adding real-time visual cues, however, may increase the cognitive demand required to learn an instrument which is cognitively challenging. More research is needed to understand the effectiveness of this approach and to identify effective methods for applying XR to music pedagogy. Work in this thesis is intended to begin filling this gap in research by evaluating the effectiveness of an XREMIL environment for the theremin.

2.3.2 XR and SMC

As Lanier demonstrated with *The Sound of One Hand* (Lanier, 1992), XR affords novel techniques for creating NIME that cannot be replicated in the real world. The adoption of XR, however, has not gained much traction in the NIME community and research of XR based musical interfaces is limited . The rest of this section discusses existing XR interfaces for musical expression and ends with a brief discussion on audio programming environments available for XR musical interfaces.

Virtual Environments for Musical Expression

The emergence of XR has created a wide range of new possibilities for the design of VEME. From virtual versions of real instruments (Mäki-Patola et al., 2005) to virtual worlds endowed with musical capabilities (Hamilton et al., 2011), XR technologies enable musical experiences that cannot be achieved in the real world.

Mäki-Patola et al. (2005) explored the idea of mimicking real world instruments in VR by developing virtual music environments based on real musical instruments: the virtual xylophone, the virtual air guitar, a virtual membrane, and the gestural FM synthesizer. In their findings, they reported that mimicking traditional instruments with VEME may not result in a more expressive performance because VR is a different medium compared to the real world. Rather than mimicking the real world, the authors suggest designing instruments specific to the medium, noting ”virtual instruments allow for intelligent visual feedback. A physical instrument’s ap-

pearance is static, but a virtual instrument can modify its visual features in " (Mäki-Patola et al., 2005). The affordances of XR suggest designing interfaces that cannot be implemented in the real world or augmenting traditional interfaces with visual overlays. The work in this thesis has been influenced by the thoughts of Mäki-Patola et al. (2005) on using visual feedback to create experiences not available in the real world.

In one of the first known works utilizing XR technology, Mulder et al. (1999) designed virtual 3D musical instruments that afforded gestural control of multiple sound parameters simultaneously. Interaction with the virtual objects followed a sculpting metaphor in which users performed a "claying" gesture to the object; thus, modifying sound parameters. Users wore Cybergloves to interact with the virtual object but the system lacked 3D visualization making interaction somewhat challenging. During their evaluation the authors noted that control of the virtual object took some effort to master as users needed to learn the limitations for controlling the object. This could have been mitigated by implementing an immersive environment to visualize the objects. Building on the ideas of 3D objects, Berthaut et al. (2011) proposed 3D reactive widgets in an immersive environment. The 3D widgets allowed for musical performance that went beyond what is possible in the real world. The reactive widgets represented complex multi-process sounds with many parameters that are difficult to interact with in the real world. Implementing a VE with carefully designed gestures and audiovisual mappings allowed the user to easily interact with multiple widgets for an expressive musical interaction. The presented ideas of 3D widgets demonstrate the capabilities of extended reality (XR) for designing expressive musical environments with interactions that would be difficult using physical objects.

Rather than using shapable virtual objects for generating rich, complex sounds, some virtual interfaces implement simple static musical objects that generate basic notes or chords. The notes or chords are then triggered by striking or selecting. The ChromaChord (Fillwalk, 2015) was an immersive musical environment composed of twelve cube shaped keys for playing single notes on the chromatic scale. The system employed a Leap Motion for hand tracking and notes were played striking the key with a hand. In addition to virtual keys, the system included a modulation window to alter

the effects of a filter. Playing a range of notes in this interface could be a challenge since only one octave is displayed at a given times.

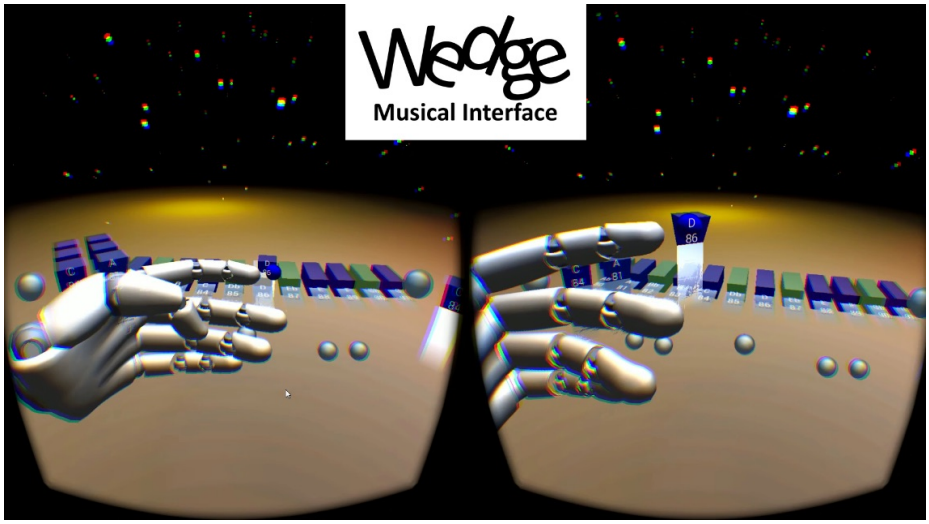


Figure 2.6: The Wedge interface for building and performing immersive musical environments (Moore et al., 2015) ©2015 IEEE.

To address this challenge and to make compositions easier to perform, some environments with single note objects allow for customization that could only be afforded with XR. Taking advantage of this affordance, Moore et al. (2015) proposed an immersive music interface called Wedge. The interface was composed of cube shaped virtual objects that each represented individual notes, see Figure 2.6, similar to the note objects in ChromaChord (Fillwalk, 2015). Striking the object triggered the note to play, similar to keys on a piano. The core feature of Wedge was the ability to build a customized performance environment by selecting and combining note objects to form musical chords and sequences affording composition specific environments. Similarly, Zielasko et al. (2015) developed a virtual environment for blowing on bottles allowed users to customize their performance space. Users of the interface were presented with a set of virtual bottles they could blow to generate sound at specific pitches. The bottles could be moved around the performance environment to suite the performer's needs and the pitch of each bottle could be adjusted by virtual adding and removing water. These interfaces have limited capabilities for generating sounds as rich and expressive as the previously discussed reactive widgets but showed how XR can be used to quickly build customized interfaces for

musical performance.

The previous works all rely on interaction with virtual objects through standard XR input devices, such as hand controllers. Böttcher et al. (2005) instead proposed a VRMI for interacting with tangible music controllers. In this work, the authors built physical flute and drum like controllers which were represented in the virtual environment as 3D objects. Interaction with the controllers was mapped to the parameters of a physical model. By moving the controllers, the user was able to change the dimensions of the physical model, and it's virtual representation in real-time, while simultaneously using it for a musical control. Using tangible interfaces as controllers for virtual music performance provided users with a clear understanding of the affordances and constraints for interaction.

As suggested by Mäki-Patola et al. (2005), there has also been research in augmenting musical interactions. Andante (Xiao et al., 2014) used a projector to overlay virtual animated objects on the piano. The user, however, did not interact directly with the objects, instead the objects walked along the keys suggesting which piano keys to the user should press. Revgest (Berthaut et al., 2017) used augmentation to enhance gestural music instruments by adding virtual objects, using projection technology, that performers could interact with. Gestural instruments are typically performed without physical objects, using only body movements to control the sound output. Adding virtual objects to the instrument affords enhanced interactions and feedback. Instead of using AR as a means for musical interaction, Bukvic and Lee (2017) developed an AR system using Google Glass, called Glasstra, to assist the conductor of a laptop orchestra. Glasstra projected real-time visual data onto the AR display to provide the conductor with information about orchestra status.

With the wide breadth of possibilities for VEME facilitated by XR research is needed to provide XR luthiers with design guidelines and best practices. Serafin et al. (2016) recently surveyed the current state of the art in VRMIs and proposed nine design principles. The principles were used to develop an evaluation framework for the VRMIs surveyed. To the author's knowledge the work of Serafin et al. (2016) is the only work that has identified design principles and guidelines specifically for MusE-XR. Influenced by Serafin et al. (2016) the research in this thesis expands on their work

by identify additional affordances and guidelines for general MusE-XR.

Audio Programming in XR

Developing MusE-XR requires a design workflow that facilitates sound design and VE design. VEs are typically developed using game engines, such as Unity (Unity, 2019) or Unreal Engine (Unreal, 2019a), designed to simplify the workflow for developing 3D environments through a suite of tools that include advanced graphics rendering pipelines and physics engines. They also include sound engines for playback of sound files with mixing, added effects and sound spatialization. There is, however, minimal support in game engines for audio synthesis capabilities desired by sound designers. Unreal Engine has an experimental package for sound synthesis (Unreal, 2019b) but the limited features of the environment may not provide designers with the full tool set supported by existing sound design environments. To support the design of immersive music environments there is a need for more robust audio synthesis capabilities.

Currently there are a some audio programming languages that support audio synthesis and processing with Unity. Faust (Orlarey et al., 2009), for example, can compile to a C library for integration through Unity's plugin API and LibPD (Brinkmann et al., 2011) has a C# wrapper that can be integrated with Unity. Most recently, Atherton and Wang (2018) presented Chunity, a Unity plugin to support the integration of ChucK within the Unity development environment. While these systems all add support for audio synthesis to Unity, designers must use Unity scripting to setup parametric control of the patches. Furthermore, integration of the tools and languages into a game engine workflow can be challenging making it difficult to quickly iterate during the sound design process.

The distributed communication facilitated by OSC, discussed in Section 2.2.3, provides designers of MusE-XR with the ability to design sounds in their preferred audio programming environment. By integrating OSC into a game engine, designers can map XR interactions to OSC messages which are sent to the designer audio programming tools of choice. Hamilton (2011) used OSC in the design of UDKOSC, an immersive musical performance environment for the Unreal Development Kit (UDK). With this

system Hamilton was able to perform in an immersive environment using avatars that interacted with objects in the virtual environment. The UnityOSC library Garcia (2019) enables OSC support in Unity but requires knowledge of C# scripting to use effectively. Furthermore, it only enables standard OSC support and it is up to the designer to mapping interactions to OSC messages. To enable rapid prototyping an easier to use OSC client is needed. To this end, in Chaper 6 I propose a novel XR toolkit to integrate OSC with Unity.

2.4 Challenges

CAMIT and NIME research both look to emerging technologies for new techniques to enhance the musical experience. The emergence of XR presents new opportunities to enhance musical experiences in these fields, but there are still a number of questions and challenges that need to be addressed.

Applying emerging technologies to enhance the musical learning process goes back almost 30 years to the Piano Tutor Project (Dannenberg et al., 1990, 1993). Due to technological constraints, however, CAMIT systems have been mostly impractical for everyday use until recently. Advances in computing power and the maturity of the Internet, have driven the emergence of CAMIT systems that are publicly accessible and easily implemented with everyday technology, such as Yousician (2019) and Skoove (2019). These systems perform pitch and onset tracking using built-in computer microphones to identify performance errors in musical practice. Learning to play music, however, is about more than just playing the correct notes at the correct time. Learning to play an instrument with proper techniques is critical to reduce performance fatigue and possible injury. Furthermore, many instruments require that students know what constitutes good sound quality in the tones they are producing. State-of-the-art CAMIT research is taking advantage of advances in sensory technologies and ML research, to generate new techniques for the automatic assessment of playing technique and sound quality.

With large research projects, such as the TELMI project (Ramìrez et al., 2019), there is a reemerging interest in developing CAMIT technologies, es-

pecially systems for assessing more than just pitch and times. There are still, however, large gaps in knowledge that need to be filled for truly effective and practical CAMIT systems. Limiting the practicality of CAMIT systems is cost and accessibility of sensors being used to track students as they perform. Systems that use specialized setups, such as motion capture or specialized sensors, limit the practicality of CAMIT systems for everyday use. Instead, researchers should begin to focus on using inexpensive setups, including easily accessible sensor or technologies already built into users' devices, such as camera technology. In this thesis, I expand on these ideas to implement a novel CAMIT system for the automatic assessment of pianist hand posture using commodity sensor technologies that are easy to set up for everyday use.

Much of the CAMIT research focuses on these computational techniques for automatic assessments but neglects to study what constitutes an effective user interface to provides students with guidance and feedback. Feedback of performance assessment is often presented to the student via 2D displays which can suffer from an inherent transfer function that requires mapping visual feedback from the 2D display to the physical instrument. Integrating XR into CAMIT has the potential address this challenge by providing methods for overlaying visual feedback directly on the music instrument being learned. To improve the state of CAMIT systems for everyday use more research is needed to better understand effective techniques and interfaces for music pedagogy. In this thesis, I aim to fill this gap by evaluating the effectiveness of an XREMIL environment with RTVF for theremin learning. Furthermore, I explore the XREMIL design process to identify affordances and guidelines, facilitating future development of XREMIL environments.

The emergence of XR has also started a small trend in the design of VEME. With the field being so young, there exist a number of questions about how to design engaging musical experiences. More research is needed to provide basic design guidelines and principles. Developing guidelines and principles, however, requires the design and evaluation of many interfaces which can be a challenge with the lack of a good environment and sound design workflow. The level of expertise currently needed to design XR environments make XR environment development inaccessible to non

programmers. Designers need enhanced abilities with a low barrier for entry to enable rapid prototype of new environments. In this thesis, I aim to address this limitation by presenting an open source XR toolkit that enables rapid prototyping of MusE-XR using OSC.

Chapter 3

Research Methodology

This chapter describes my approach to conceptualizing and answering research questions explored through this thesis. Shneiderman (2016) argues, in *The New ABCs of Research: Achieving Breakthrough Collaborations*, that projects combining applied and basic (pure) research result in higher impact work, and more significant advances, than performing either one alone. The research methodologies used in this dissertation take inspiration from Shneiderman's proposed Applied and Basic Combined (ABC) research model. The overall intent of this dissertation, the application of emerging XR technology to SMC, falls under applied research, but basic research is needed to fully integrate the new technologies into the research area.

I first describe the exploratory approach I take in this thesis to find areas of SMC where I could contribute the most. This leads to a discussion of two major research tracks investigating the use of XR for enhanced musical pedagogy and for enhanced musical performance. Next I describe the methodologies used for each of the three main projects exploring the application of XR to these areas. The high level research approach taken for this dissertation is outlined in Figure 3.1 while Figures 3.2, 3.3, and 3.4, provide a detailed look at the individual methodologies used for each research project.

3.1 Research Exploration and Conceptualization

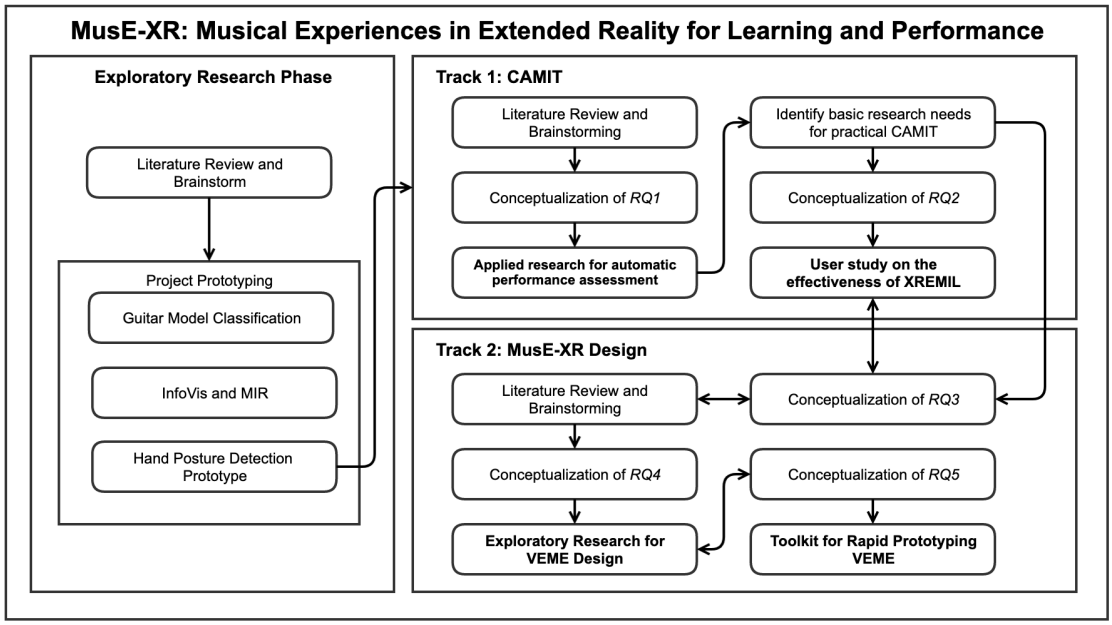


Figure 3.1: Methodology for research on musical experiences in extended reality (MusE-XR)

3.1.1 Exploratory Research

During the exploratory phase of my research, I performed literature reviews and implemented projects in various areas of SMC to find open problems to work towards addressing. Previous research I performed during my master's degree was focused around HCI and interaction design for multimodal interactive music and art systems with natural user interaction (NUI). This research was enabled through depth camera technology, such as the Microsoft Kinect (Johnson et al., 2014). My goal was that my research would build on these competencies and expand my knowledge of new topics.

With the field of music information retrieval (MIR) being my supervisor's core research field, it was an obvious first choice to explore. This area was of interest due to is interdisciplinary nature that integrates research from fields such as machine learning (ML) and audio signal processing (ASP).

Around the time I was reviewing the MIR literature, I had a conversation with a peer about the sound quality of expensive guitars and how well a computer model could distinguish the subtle differences in timbre between different guitar models. This felt like an interesting idea to explore in more detail. After a brief literature review, I found that there was significant MIR research in the classification of music instruments but research in classifying models within a single instrument class was limited. This led to an experiment to test if standard ML methods would work for the classification of guitar models from audio recordings (Johnson and Tzanetakis, 2015). Using an ML approach required a data set to train the classification model. Since this was new research, there was no existing data set to work with. Instead, the project required that I create a new data set with audio recordings from a wide range of guitar models. Using the newly created data set, I applied ASP to the recording to extract features to train the classification models. The system had varying levels of success with different classification algorithms. I found this to be an interesting project that taught me new skills including ASP, ML and data set creation, but, ultimately, was lacking in topics around HCI and not the direction I wanted to proceed in.

Another project I completed during the exploratory phase combined the field of information visualization (InfoVis) with MIR to visualize the musical qualities of a song over time. The goal of this project was to explore the potential of visually representing a song in such a way that would help a user find music they like, for example, to add to a playlist, without having to listen to the song. At the time, music visualizations in the time domain was limited to spectrograms and waveforms which only visualize one quality of a audio signal, frequency and amplitude, respectively. The general musical listener needs more information to create a mental image of music using a visual representation. For this project, I experimented using timbre, pitch, and loudness data provided by the Echonest Track Analyzer API (now part of Spotify) to create a visual representation of a song over time. Each attribute was visually encoded, using either bars or colors, then organized using rhythmic information to produce a final song glyph. This project explored interesting ideas in InfoVis and MIR but at the time I was more interested in another experiment I started around the same time.

At that time, previous work from my supervisor in computer assisted musical instrument tutoring (CAMIT) caught my interest (Percival et al., 2007). Performing a literature review in this area, see Section 2.1, revealed emerging work in the automatic assessment of musical performance to provide students with feedback about their practice performance. Most of the work focused on the identification of pitch and timing errors in audio data Schoonderwaldt et al. (2005); Dannenberg et al. (1993), but there was also emerging research in analyzing performer technique using sensory technology, such as motion capture, cameras and depth sensors Mora et al. (2006); Ng et al. (2007). This inspired me to integrate my previous experience with depth cameras with recently learned ML knowledge to implement a system for automatic assessment of musical performance using the Kinect. This idea was the starting point for my research which ultimately led to the two tracks taken in this dissertation.

3.1.2 Conceptualization of Research Tracks

The exploratory phase led to the **first research track investigated in this dissertation: computer assisted musical instrument tutoring (CAMIT)**. This research track started with the design of a CAMIT system to automatically assess musical technique using the Kinect. Implementation of such a system requires interdisciplinary research in computer vision (CV), ML, and HCI. CV and ML are employed for the underlying assessment models, and HCI informs the design of the interface to present students with performance feedback about their performance. This opened the door for two possible directions to further my contributions to CAMIT research: advancing the state of art in hand detection using CV and ML, or advancing the state of HCI interface design for musical tutoring. Due to my expertise and interest in HCI, I decided to focus my research efforts on methods for enhancing musical tutoring with real-time visual feedback (RTVF) of musical performance assessment data.

A literature review revealed limited results on interface design for performance assessment, but other research fields provided inspiration. Research in the field of New Interfaces for Musical Expression (NIME) exposed me to work on hyperinstruments Machover (1992), or instruments

augmented with sensors for added functionality. Integrating the idea of hyperinstruments, inspired thoughts on augmenting a piano to enhance musical learning. While this would be challenging to implement through physical methods of augmentation, advances in XR afford techniques for virtual augmentation that could proof beneficial. This idea would inspire **the second research track investigated in this dissertation: musical experiences in extended reality (MusE-XR)**.

3.2 Research Tracks

3.2.1 Track 1: CAMIT

CAMIT research has been going on for nearly 30 years (Dannenberg et al., 1990, 1993) but there still exist challenges hindering the development of practical CAMIT systems. ***The research goal of this track is to gain an understanding of the challenges and limitations for developing practical CAMIT systems***. My overall objective is to output practical solutions for the design of effective CAMIT systems. To achieve this goal, the track follows the ABC model (Shneiderman, 2016) approach through applied and basic research approaches.

The first research project in this track employed an applied research approach with an objective of implementing a practical CAMIT system using emerging technology. Brainstorming topics to advance CAMIT research, led to the idea to integrate depth camera technology to enhance the musical learning process. To expand on this idea, I performed an initial literature review (see Section 2.1) to identify areas to apply depth camera technology. The review revealed a promising area of research, the automatic assessment of musical performance. The goal of this area is to use computational techniques to analyze musical performance, and generate feedback on the quality of the performance being assessed. Most of the literature focused on identifying pitch and timing errors using audio data, but there was some emerging research on using motion tracking methods to assess student technique. One of major limitations was the need for expensive equipment not readily available outside of a laboratory environment. To make CAMIT more accessible, assessment systems should employ cheap,

commodity sensor technologies. To this end, I hypothesized that commodity depth cameras could be utilized to address this limitation and make advanced CAMIT more accessible.

Discussing common mistakes made by piano students with Isabelle Dufour, a piano teacher and Ph.D. student in my research lab, and Tom Arjannikov, another Ph.D. student in the lab, we conceptualized an idea for assessing pianist hand posture using the Kinect depth camera to help beginning students' improve their playing technique. This led to the conceptualization of RQ_1. To answer this question I designed an applied research project exploring the application of state-of-the-art CV and ML research to implement a system for identifying and analyzing hands in recordings of piano students practicing. More concretely, I applied existing research to address the technical challenges of tracking and assessing pianists hands for *the first project of this dissertation: the automatic assessment of pianist hand posture*. The methodology used for this project is described in detail in 3.3.1.

The research project was designed with the expectation of identifying research gaps where basic research is needed to fulfil the design of a practical CAMIT system. The main gap identified through this research was the limited evaluation and design guidelines for effective methods for providing students with assessment feedback. The hand posture assessment system was developed using technologies that enabled the use of XR to display assessment feedback. To that end, I was inspired by the concept of hyperinstruments and the capabilities of XR to develop learning environments with real-time visual feedback (RTVF) mapped directly onto musical instruments. There was no research, however, on the effectiveness of such environments, leading to RQ_2. I aim to answer this question with *the second project in this dissertation: evaluating the effectiveness of XREMIL for the theremin*. To answer this question and add to the general body of XR training and CAMIT knowledge, I performed a controlled experiment described in Section 3.3.2.

3.2.2 Track 2: MusE-XR

XR is a set of emerging technologies that have led to the creation of interesting applications in a number of different areas, such as gaming, training, and marketing to name a few. Research on the use of XR for CAMIT and other musical applications, however, was limited which prompted me to consider other methods for applying XR to enhance musical experiences. This led to the conceptualization of the second track of this dissertation: musical experiences in extended reality (MusE-XR). ***The research goal of this track is to understand the affordances of XR to enhance music experiences.*** The overall objective is to output general guidelines and solutions for developing MusE-XR.

A brief literature review on MusE-XR resulted in relatively few results, aside from research regarding spatialized sound. Most of the available literature, however, came from two areas that fit into my competencies: NIME and CAMIT. The publications that did exist in these areas were generally limited to proof of concept systems that add little general knowledge to the overall design of MusE-XR. To this end, I performed two experiments to gain knowledge of the design space. To gain an understanding of the MusE-XR design space I posed RQ_3 and RQ_4.

The two tracks of this dissertation have differing goals, but there is overlap in the projects designed to achieve them. The second project of this thesis, regarding evaluation of the effectiveness of XREMIL, was also intended to address RQ_3. In addition to administering a controlled experiment, I obtained data regarding the design of XREMIL through the interactive design process while developing the learning environments as well as through user questionnaires and interviews administered with the user study, discussed further section 3.3.1.

The second project in this research track moves away from musical learning and looks to the application of XR for musical expression. An initial literature review revealed only a few instances of VEME in the NIME literature, resulting in limited insights towards design guidelines to inform the design of new environments. This gap in literature lead to RQ_4. To answer this question I started a project developing multiple categories of VEME to gain insights on the design space. While working on this project, I

realized the readily available options to integrate sound synthesis capabilities needed for VEME development were limited, leading to RQ_5. Research efforts to answer RQ_4 and RQ_5 added a goal to support rapid prototyping. This influenced *the third project of this dissertation: the development of an open source toolkit for building and prototyping VEME*. Enabling rapid prototyping would make it easier for myself, and other researcher, to explore the design space.

3.3 Research Project Methodologies

The approach taken for this thesis employs a creative exploration through the SMC space to find new methods to integrate XR technologies. Using the ABC approach from Shneiderman (2016), I designed and administered three main projects to answer research questions posed in the thesis: RQ_1, RQ_2, RQ_3, RQ_4, and RQ_5. Each project attempted to answer specific research questions and used methodologies specific to the questions being addressed. This section describes the research methodologies for each project.

3.3.1 Automatic Assessment of Pianist Hand Posture

Taking an applied research approach, shown in Figure 3.2 and discussed in full detail in Chapter 4, the project consisted of two phases aimed at addressing RQ_1. The first phase explored the feasibility of the CAMIT system and resulted in a prototype model to demonstrate the proof of concept. The second stage resulted in a more practical implementation by refining the model based on the identified shortcomings of the initial prototype.

In phase one, after defining the research question, I reviewed the literature for existing methods to employ in the development of a hand assessment model. The initial review resulted in only a few related works for tracking pianists' hands. The research, however, did not provide the applicable techniques needed for a complete hand posture assessment model. Instead, I looked to other CV research regarding image segmentation, which deals with extracting object from images, and object detection, which deals

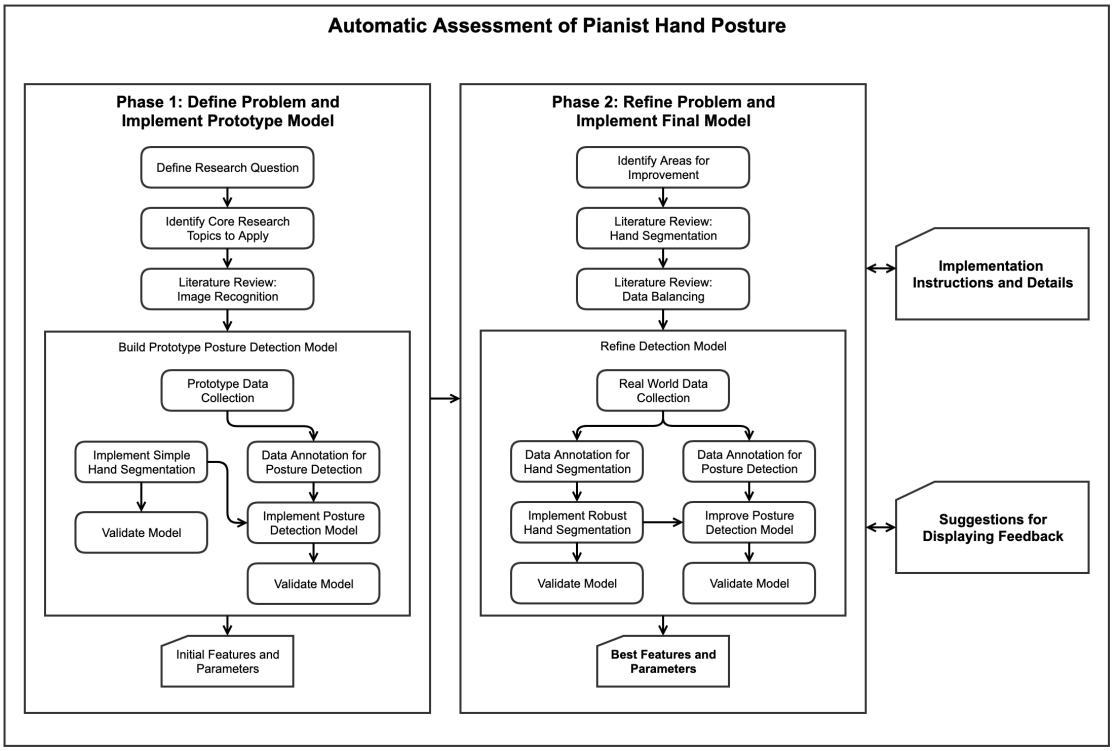


Figure 3.2: Research Methodology for Experiments on the Automatic Assessment of Pianist Hand Posture

with identifying object classes. This research provided the technical details needed to implement a hand posture system using CV and ML methods.

Employing ML approaches requires a data set to build the hand segmentation and detection models needed for the system. During the proof of concept phase, I decided to employ an artificially created data set for initial validation of system feasibility. Collaborating with Isabelle Dufour, we created an artificial data set by recording ourselves while playing piano exercises for beginners. We performed each exercise three times, each time using one of three hand postures. This afforded a contrived data set that was easily annotated with posture classes. Using this data set allowed me to quickly begin experiments applying the techniques learned through the initial literature review.

Using this data set, I employed a basic approach for image segmentation which worked well enough with the artificial data to identify and segment the left and right hands from the depth maps. With the hands segmented from the depth map, features descriptors were extracted to design the pos-

ture detection models. The initial experiments employed two feature descriptors from the object detection literature; one was a general approach originally designed for RGB cameras, while the other was specifically designed for object recognition in depth maps. Using these descriptors, I developed hand posture detection models using techniques from the literature review. Validation of of the model demonstrated the potential for a working hand posture detection system. Since the data was contrived, however, research using a more realistic set of data was required.

The goal of the second phase of the project was to implement a more practical system by using realistic data and improving upon limitations identified during the initial prototyping phase. In addition to acquiring real world data, a new method for hand segmentation was need as the basic approach required RGB data and was not robust to varying conditions.

To create a new data set that more realistically represents beginning piano students, I worked with Isabelle to recruit her piano students that we would record while playing practice exercises. We were able to recruit six of her students. During the recording sessions, students were not asked to perform with specific hand postures, instead we asked them play as they normally would during practice. As they played, we made notes on the correctness of their hand posture. After creating all the recordings, Isabelle and I annotated the data from the notes using an annotation interface developed to simplify the process of annotating individual depth maps.

As expected, the original hand segmentation technique did not work well with the smaller hands of the piano students. Therefore, a more robust segmentation model was required. A literature revealed that state-of-the-art hand segmentation employed a per pixel classification approach. To implement this approach, a subset of frames from the recordings was annotated and a ML model was developed to classify individual pixels as either left hand, right hand, or background. Validation of this approach showed the model to be more robust to different hand characteristics affording a generalized model for all students.

I was now able to develop and validate the posture detection models using techniques from the prototype. First, I attempted to build one generalized model for all the students, but this did not fare well due to a wide variation in hand characteristics and playing styles. To account for this

this, individually trained models were developed for each student. This worked well in most cases, but one challenge did arise. In a few cases, the data from some of the posture classes were too imbalanced with a large percentage of the data from only one class. This required additional research on addressing imbalanced data which was integrated into hand posture detection pipeline.

The contribution of this project is a detailed set of instructions, including the ideal feature descriptors and ML parameter values, for developing a practical hand posture assessment system. Additionally, experience developing a CAMIT system, allowed me to provide insights into methods for displaying of assessment feedback. Evaluation of feedback methods, however, was left for a following experiment.

3.3.2 Evaluating the Effectiveness of XREMIL

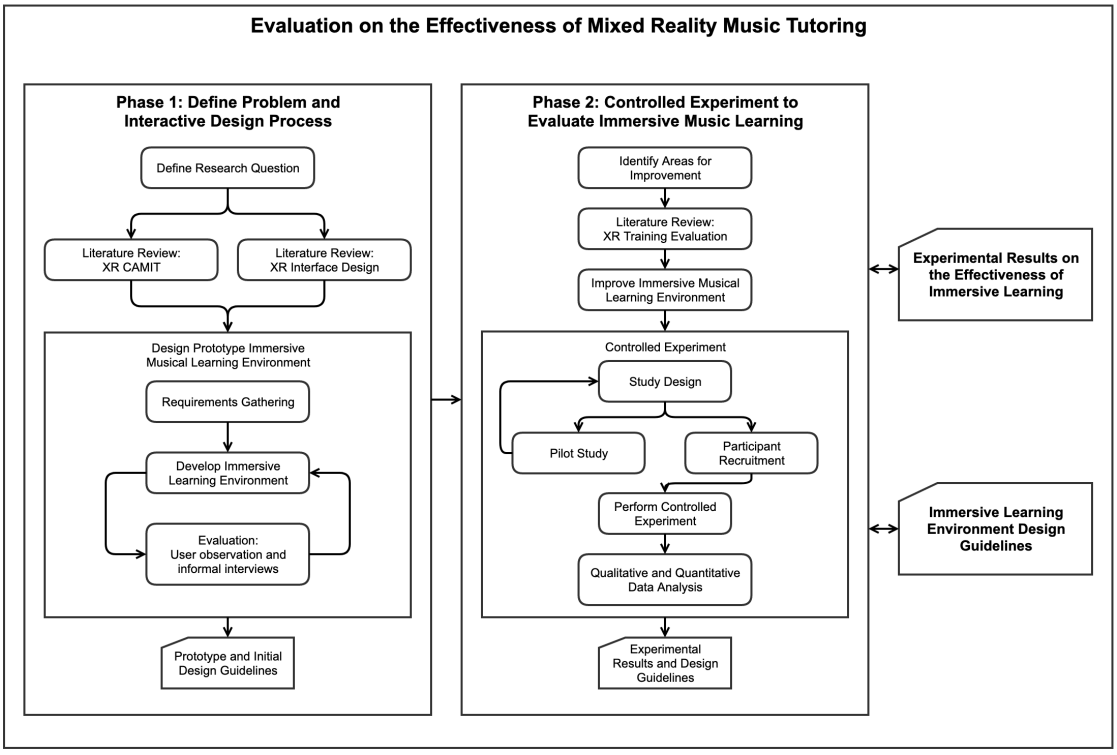


Figure 3.3: Research Methodology for Evaluating the Effectiveness of XREMIL

The previous experiment shows the application of emerging technology

for pianist hand posture assessment, but a full CAMIT system requires an effective method to present assessment results to students. Influenced by the idea of hyperinstruments, I decided to use XR as a method for presenting RTVF but there was no research on the effectiveness or design of such systems. Intending to fill this gap and address RQ_2, I take a controlled experiment approach for this project. The project also includes the administration of questionnaires and interviews with participants to obtain subjective design feedback to help address RQ_3.

To answer these questions, this project takes a two phase approach, shown in Figure 3.3 and discussed in full detail in Chapter 5. The first phase involved following the interaction design process (Rogers et al., 2015) to develop an XREMIL for use in a controlled experiment. Before conducting the experiment, however, I performed an initial evaluation of the environment through informal evaluations and demo sessions. Phase two involved performing a controlled experiment to evaluate a version of the XREMIL revised based on earlier evaluation.

To implement my previously discussed system for pianist hand posture assessment with XR for a controlled experiment requires resources not readily available to me, such as immersive AR. Using resources that were available, I designed an MR environment to teach users to play notes on the theremin. This setup afforded the development of an immersive MR environment to simulate an AR experience.

The first prototype developed through the design process, was implemented using the Google Daydream mobile VR platform. I was able to evaluate this environment on two different occasions. First, I held a demo session at a local tech exposition, *Discover Tectoria*, which was open to the general public. Here I was able to observe users from the general public interact with the system and held informal discussions with them about their experiences. Second, a demo session with a slightly revised edition was held at the 2016 International Conference of New Interfaces for Musical Expression (NIME). During this demo, I held informal discussions about the system with users that had advanced musical technology knowledge. Developing the environment using an interaction design process with input from multiple groups of users led to a set of design guidelines for immersive musical learning environments. Additionally, the informal obser-

vations and discussions, led to a number of design improvements for the next version that would be used with a controlled experiment in phase two.

In phase two of the project a controlled experiment was conducted to evaluate the effectiveness of RTVF with XREMIL. Before administering the experiment, I revised and improved the learning environment based on evaluation from phase one. The biggest change was moving to a more powerful, PC connected MR system, the Samsung Odyssey HMD for the Windows Mixed Reality (WMR) platform (Microsoft, 2019). The changes included refactoring the original implementation work with this device as well as making some environmental design changes to make the interface more usable. At the same, I performed a literature review of XR training research to better understand techniques used for the evaluation of XR training environments. With the new environment completed and the literature reviewed, I designed and conducted the user study.

The user study was designed for both objective and subjective evaluation by obtaining quantitative and qualitative data about the training experience. Quantitative objective data provided a measurable basis for comparing three training environments. While quantitative and qualitative subjective data provided an understanding of user needs and limitations. After the initial experimental design, I performed a pilot study with six participants to work out any issues in the experiment methodology. Addressing changes from the pilot study, I then recruited participants to partake in the experiment. Recruitment was performed on a rolling basis over a month to ensure I was able to recruit at least thirty participants in a timely manner.

For each participant, I administered the experiment and collected quantitative data related to their learning performance. To obtain subjective quantitative data about their experience, I had participants complete three questionnaires, that are common in the XR training literature. Afterward, the participants were interviewed for a qualitative assessment of the environment. After all participants had completed the study, the data was organized and coded as needed to prepare for analysis.

The outcomes of this project are experimental results which contribute to the literature on the effectiveness of XR training particularly in a musical setting. Additionally, I identify design guidelines from the data analysis to aid the future development of XREMIL.

3.3.3 Virtual Environments for Musical Expression

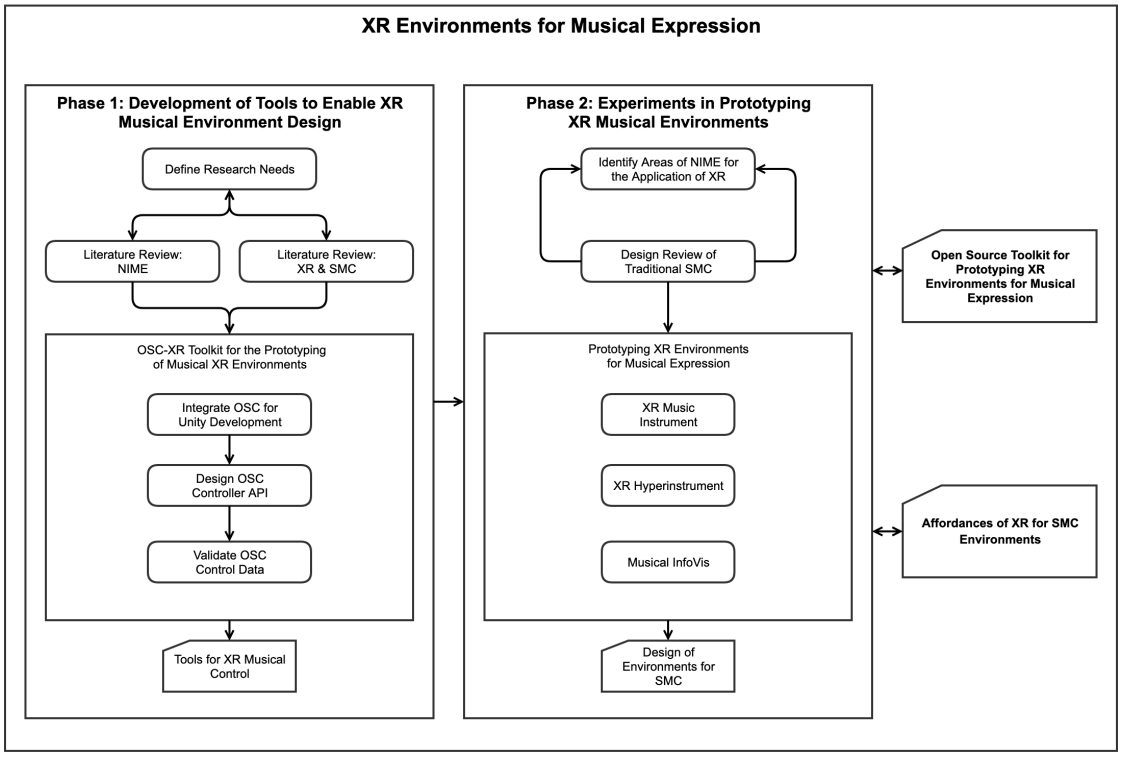


Figure 3.4: Research Methodology for the Design of the OSC-XR Toolkit for Prototyping VEME

The previous project inspired the integration of XR with SMC for MusE-XR. In that project, the application was specifically for musical learning. The aim of this project, on the other hand, is to expand XR into other areas within SMC. The project employs an exploratory approach, shown in Figure 3.4 and discussed in full detail in 6, to address RQ_4 which also led to the conceptualization of RQ_5.

To address RQ_4, I planned to develop a series of VEME to discover the XR affordances that contribute to the VEME design space. I review literature related to traditional NIME for design inspiration as well as XR based SMC for the latest techniques on applying XR for VEME development. This review revealed interesting works that provided inspiration for the development of various VEME, but work supporting design and development of MusE-XR was limited, especially in regard to the integration of sound design capabilities.

Sound design and synthesis tools are essential for the development of NIME and VEME. There are numerous audio programming languages, libraries and toolkits available for traditional NIME development but integrating these tools into the VEME workflow is still a challenge. To enable the rapid prototyping workflow to support RQ_4, a better tool set was needed to integrate sound design into the development workflow, raising RQ_5.

Addressing RQ_5 lead me to explore the use of standard NIME technologies, specifically Open Sound Control (OSC), to integrate sound design tools with the XR development workflow. OSC is a well known communication protocol that has supported the development of countless NIME by enabling communication between control interfaces and sound engines. Using this core technology, I design and develop an OSC toolkit for XR, called OSC-XR, to make the development of VEME more accessible to NIME designers of all skill levels. Using this toolkit, I implement three use cases to showcase its capabilities and explore the affordances of XR for VEME design. The three use cases provide experience in the the design of VEME that mimic traditional musical controllers and extend NIME into the virtual world, enabling interactions not possible in the real world.

The main outcome of this project is an open source toolkit shared with the SMC community to support VEME development. Additionally, I identify affordances and guidelines for the design of engaging MusE-XR. My hope is that these outcomes will facilitate future development and research on MusE-XR

Chapter 4

Automatic Assessment of Pianist Hand Posture using Depth Data

This chapter presents a novel CAMIT system for the automatic assessment of pianist hand posture to address RQ_1. I approach this question through application of existing computer vision (CV) and machine learning (ML) methods to develop a novel hand posture detection system. Because this is a new approach with limited research to build on, I first develop a prototype system to explore the feasibility and learn the limitations of hand posture detection using depth cameras. After validating the approach, a more robust hand posture detection system using a realistic data set is implemented and evaluated. Based on my experiences developing the system, I touch on implications and design considerations for integrating the system into a CAMIT interface. The work presented in this chapter has resulted in two publications [1,2].

[1] D. Johnson, I. Dufour, G. Tzanetakis, and D.Damian. Detecting Pianist Hand Posture Mistake for Virtual Piano Tutoring. In *Proceedings of the International Computer Music Conference*, 2016.

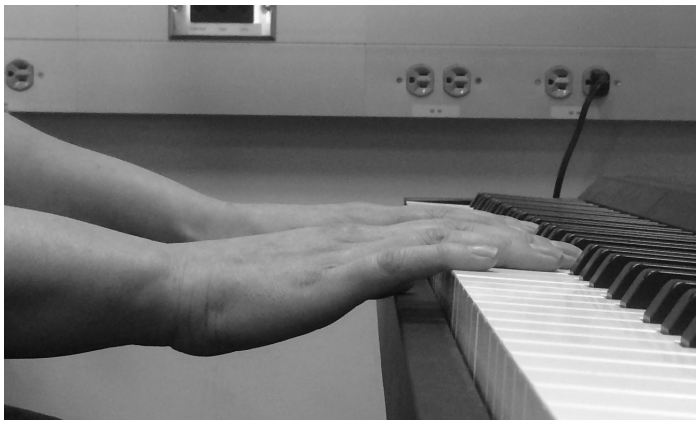
[2] D. Johnson, D.Damian, and G. Tzanetakis. Detecting Hand Posture in Piano Playing Using Depth Data. *Computer Music Journal*, To appear Spring 2019.

4.1 Introduction

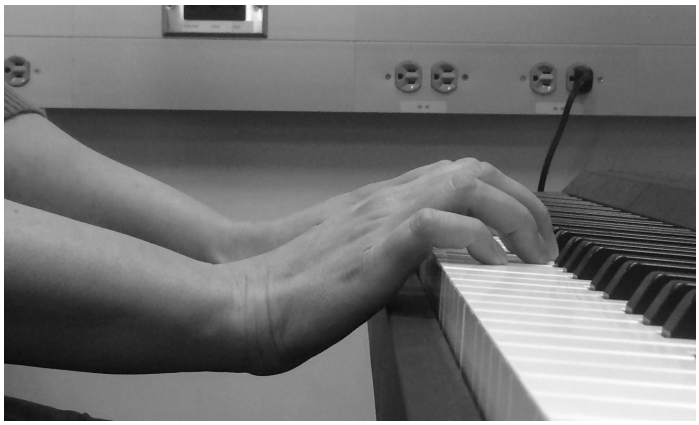
A significant portion of computer assisted musical instrument tutoring (CAMIT) research aims to enhance music pedagogy with tools and interfaces to automatically assess musical performance affording personalized feedback even when professional teachers are not present. Many CAMIT systems rely on audio signal processing (ASP) to assess the musical quality of a performance, omitting evaluation and feedback of a student's physical playing technique. Recently, CAMIT researchers have recognized the importance of assessing physical technique as well. Projects such as i-Maestro (Ng et al., 2008) and Technology Enhanced Learning of Musical Instruments (TELMI) (Ramìrez et al., 2019) have implemented methods for the automatic assessment of playing technique in stringed instrument practice. The major contribution of this chapter is a CAMIT system for the automatic assessment of piano playing technique, specifically the quality of the student's hand posture. The assessment is intended to be integrated with a CAMIT interface to present students with feedback regarding the assessment. Such an interface will enhance piano learning by providing students with immediate feedback about their performance without the need for an expert analysis.

4.1.1 Pianist Hand Posture

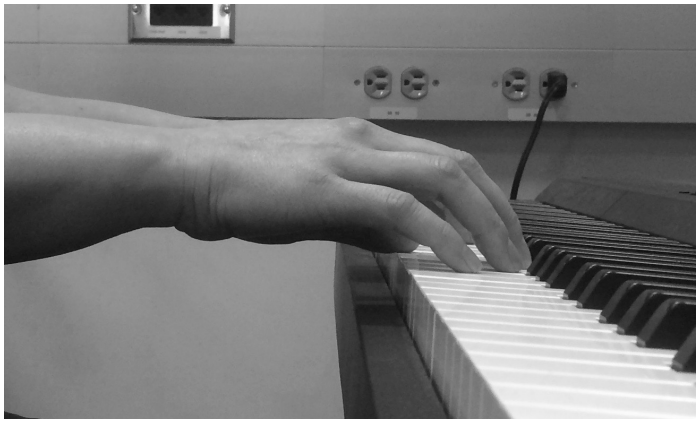
Body and hand posture are fundamental to proper technique in piano playing. Providing musicians with feedback regarding their technique is essential for musical skill acquisition, especially in the case of repetitive practice where consistent bad technique may lead to injuries in extreme cases (Riley et al., 2005). To assess the impact of multimodal feedback on pianist technique, Riley et al. (2005) provided pianists feedback of their performance through the analysis of Musical Instrument Digital Interface (MIDI) data, video recordings, and Electromyography (sEMG). The latter was added to augment video analysis after they found that, even for experienced pianists, reviewing videos frame by frame did not help identify issues. Augmenting the system with sEMG improved the results, but they noted that analysis required time and patience from both the student and the instructor.



(a) Flat Hands



(b) Low Wrists



(c) Correct

Figure 4.1: The three common hand postures of beginning piano students that are detected with the presented system. Figures 4.1a and 4.1b show common postures mistakes made by students, while Figure 4.1c shows the hand in the ideal posture for pianists.

In contrast, our system is intended to generate data for immediate performance feedback without complex analysis.

For correct hand posture, the hand should be arched and the fingers curled as illustrated in Figure 4.1c. Working with a piano teacher we identified two common posture mistakes observed in students: playing with flat hands, Figure 4.1a, and playing with low wrists, Figure 4.1b. Because most of a student's practice time occurs between lessons, bad habits can quickly become chronic. Providing students with a tool that can identify and help correct these mistakes during daily practice would reduce the probability that they become ingrained in the student's playing technique.

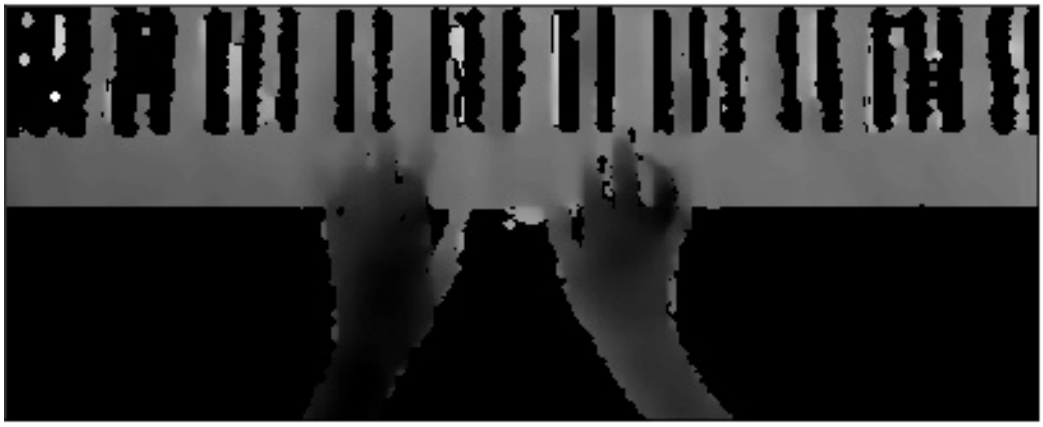
4.2 Approach for Hand Posture Assessment

To automatically assess pianist hand posture I present a novel approach that employs a 3D structured light camera, also known as a depth camera, such as a Microsoft Kinect or an Intel Realsense SR300 (I experimented with both cameras during this project). A depth camera is used to calculate the distance (or depth) of objects from the camera sensor. Using the depth data obtained from a depth cameras affords detailed hand geometry for inferring hand posture that a standard RGB camera cannot capture; additionally, the use of a depth camera affords a non-invasive setup with easy installation in any practice space.

To this end, I propose an approach that places the depth camera above the pianist hands and uses the obtained depth data for posture detection. Figure 4.2 shows the camera placement (Figure 4.2a), the scene captured by the camera (Figure 4.2b), and an example of a depth map (Figure 4.2c), an image in which each pixel represents a distance value rather than a color value. While it may be intuitive to capture a side view of the hand similar to the images in Figure 4.1, this would require two cameras to on each end of the piano to capture both hands. The proposed setup affords the recording of both hands with a single camera and it mimics the view a camera built into an XR HMD might have. Furthermore, placing the camera above the piano allows for additional information to be captured that could be integrated into a tutoring system, such as the location of the

(a) Camera setup

(b) Overhead view



(c) Example depth map

Figure 4.2: The depth camera is positioned with an aerial viewpoint to capture both hands from overhead. Figure 4.2b shows the RGB view of the camera which is used for data annotation. Figure 4.2c shows an example of a depth map that is used for model training and detection.

error being made, which notes are currently being played and by which fingers.

Using this configuration, my proposed approach requires applied computer vision (CV) and machine learning (ML) methods for hand segmentation followed by hand posture detection. The first step in the approach is to isolate the left and right hands from scene captured in the depth map using image segmentation techniques. Once the hands are isolated from the scene captured in the depth map, the posture of each hand is assessed using supervised learning to classify it as one of the three postures classes shown in Figure 4.1. Applying existing research to this novel approach

raises two new questions to address:

RQ_{1_1} Can existing image segmentation methods be used to segment hands that are in constant motion and in direct physical contact with the piano?

RQ_{1_2} Does the depth camera capture hand depth with enough detail to discriminate subtle changes in hand posture for supervised learning?

These questions are addressed through the research presented in this chapter. Before discussing the research, I present a brief overview of the two CV tasks, required for my proposed hand posture detection approach. First, state-of-the-art techniques for the task of hand segmentation are discussed. Next, I discuss the feature extraction process, including the two image descriptors employed to create a feature set for training the posture detection model.

4.2.1 Hand Segmentation

With the emergence of XR, researchers are exploring new methods to interact in more natural ways. This has led to an emergence of research on detecting the location of body and hand parts in 3D space using camera based technologies. This has afforded users new modes of interaction without the use of a physical controller. For example, hand pose recognition research employs depth cameras to identify detailed spatial information about key features of the hand, such as joint locations in the 3D space. The first step in the process for pose recognition is to segment the hand from the depth map. A similar process is needed for hand posture assessment since the pianist's hands must first be segmented from the depth map scene in which the hands are interacting with a piano.

Body part segmentation from depth data is a well researched problem in computer vision. One of the original needs was to identify body parts from depth maps to find specific joint locations for body pose recognition. To label the 31 parts of the body, (Shotton et al., 2011) employed per pixel classification with a random decision forest (RDF) trained with custom depth image features (DIFs). Similar approaches have been employed for hand

segmentation for use with hand pose recognition. Keskin et al. (2013) used the same approach as Shotton et al. (2011), including the same DIFs, to identify 21 hand parts from a depth map. Tompson et al. (2014) used this approach as well, but to segment the entire hand from the depth image rather than individual hand parts. Liang et al. (2014) also employed per pixel classification to parse hands parts from a depth image but they implemented a new feature descriptor, depth context feature (DCF), for each pixel. The new pixel descriptors improved segmentation accuracy compared with the DIFs of Shotton et al. (2011). In all of these works, there was just a single hand in the scene and the hand was not in physical contact with any other objects. In contrast, my work involves a depth map scene containing two hands which are both in physical contact with a piano.

There has also been research into segmenting a hand from a depth map in which the hand is interacting with another object. Liang et al. (2016) extended their earlier work (Liang et al., 2014) with a system for playing a virtual piano. In their work, fingertips are tracked while tapping on a flat surface to mimic piano playing. To segment the hand they employed a skin detection model integrated with the RANSAC algorithm for plane fitting to improve segmentation accuracy. To avoid the added complications of skin detection and integrating color and depth data, my proposed approach for hand segmentation requires only depth data. Furthermore, the use of the RANSAC algorithm may cause problems when a pianist's hands or fingers move below the identified plane. Kang et al. (2016) demonstrated that the per-pixel classification approach can be successfully employed to segment a hand interacting with an object using the DIFs of Shotton et al. (2011). Using the per-pixel approach, RDF hand segmentation models were developed using pixel descriptors extracted from depth maps of participants interacting with various objects. In my work, the scene captured in the depth maps poses a novel challenge in that it contains two hands that are in contact with a piano to be segmented. To address this challenge, I employ the per-pixel classification approach taken by both Shotton et al. (2011) and Liang et al. (2014) and compare the performance of their image descriptors, DIF and DCF respectively, to identify which works best for the unique hand segmentation task presented in this work.

4.2.2 Feature Extraction

Employing an ML approach for posture detection requires a method for extracting features from the hand depth map to create a vector based representation of the data. Generally in CV this involves calculating descriptors of each image, or depth map, that describe features such as color or shape. In this chapter, two image descriptors from the CV literature are compared to create a discriminative feature set to use for developing hand posture detection mode histograms of oriented gradients (HOG) and histograms of normal vectors (HONV). HOG and HONV both provide techniques for extracting descriptors from depth maps, but only HONV were designed specifically with depth information in mind. HOG were designed for and are typically used for object and human recognition with RGB and gray scale images but have been applied to image recognition using depth maps (Spinello and Arras, 2011; Lai et al., 2011). On the other hand, HONV descriptors, while influenced by HOG, were specifically designed for depth data to describe the geometry of the surface of objects.

Histograms of oriented gradients (HOG) capture local shape through edge strength and direction. In the RGB space HOG features are calculated by approximating the derivative of color intensity in the X and Y directions of an image. The gradients are converted to polar form to generate orientation angles and corresponding magnitudes for each pixel in the image. Next, histograms are generated for the image through sliding non-overlapping windows (or cells). For each cell, orientation angles are voted into bins with the votes weighted by the magnitudes, thus, capturing both the direction and strengths of change. HOG extraction also includes a process to normalize gradient strengths over a block of cells. Dalal and Triggs (2005) explored four normalization schemes, *L1-norm*, *L1-sqrt*, *L2-norm*, and *L2-Hys*. They found that all work equally well except *L1-norm*, which reduces performance by 5%.

While the work of Dalal and Triggs (2005) was performed on RGB images, HOG has also been shown to work for object and human detection with depth data (Spinello and Arras, 2011; Lai et al., 2011). Although the data is not in the RGB space, HOG calculate the orientation and magnitude of the change in depth values. Thus, when applied to depth maps these

features capture the shape of an object not only via edge direction but also by capturing the depth gradients over the surface of the object. For example, when a pianist is playing with their wrist too low, the gradients of the top of the hand will be greater than when playing in correct form in which case, the top of the hand is flat.

Histograms of normal vectors (HONV) descriptors on the other hand, were developed specifically for depth data to provide a geometric representation of objects (Tang et al., 2012). With HONV, the X and Y gradients are used to calculate the azimuth and zenith angles of normal vectors of unit magnitude. The angles of each pixel in a window are voted into two dimensional histograms. Experiments performed by Tang et al. (2012) showed that HONV generally perform better than HOG in object recognition using depth maps. While the authors don't apply block normalization to their implementation, we've added *L1-sqrt* normalization to explore its effects on hand posture detection and more closely replicate the HOG implementation.

4.3 Feasibility Assessment

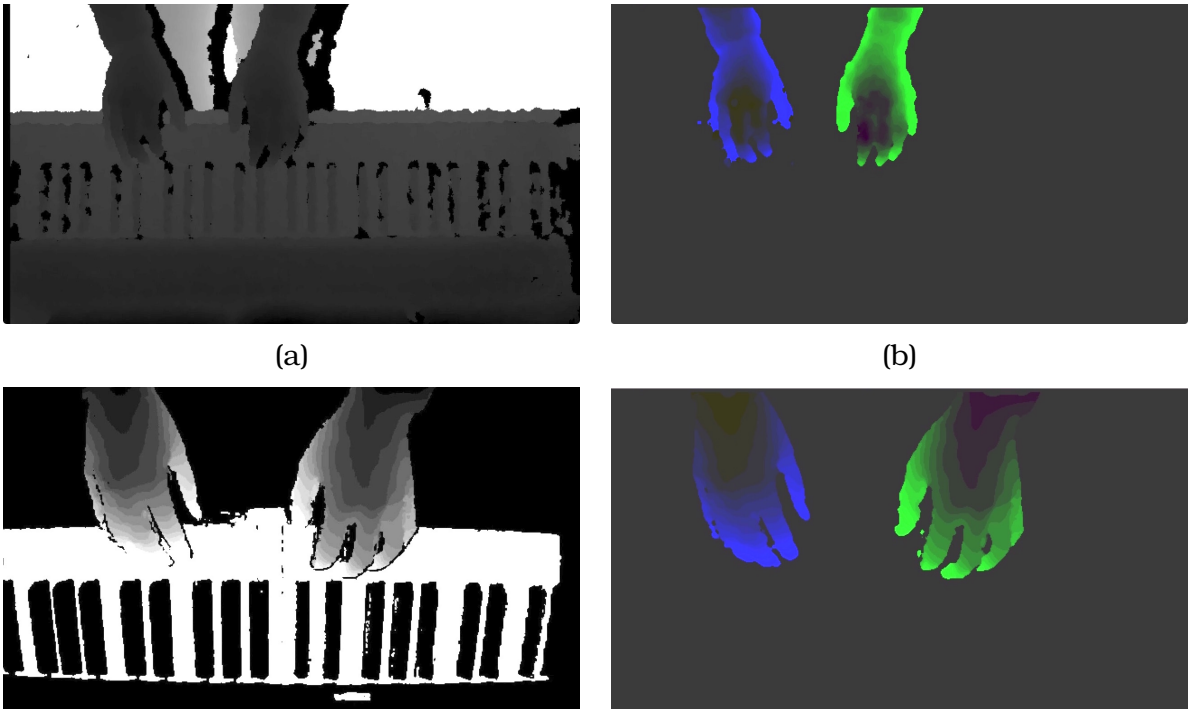
Data acquisition can be an expensive process. This is especially true with pianist hand posture assessment since an expert, i.e. a piano teacher, is required to assist with data annotation. Before expending resources to create a realistic data set needed for a fully fledged hand posture assessment model, the feasibility of the proposed approach should be validated and limitations should be identified. During this phase, I also experiment with two different depth cameras, the Kinect and the Realsense SR300, to evaluate their performance in regard to posture detection. This section describes the development of a prototype hand posture assessment system using an artificially created data set and the two depth cameras.

4.3.1 Prototype Description

The proposed approach for hand posture assessment requires two main steps. First, hand segmentation is performed to isolate the hands from the scene captured in the depth map. Second, feature descriptors are extracted

from the hand depth map and used to develop a hand posture detection model.

Hand Segmentation



(a) (b)

(c) (d)

Figure 4.3: **a)** Original depth map from the Kinect, **b)** Hands segmented from Kinect depth map, **c)** Original depth map from the Intel Realsense, **d)** Hands segmented from the Realsense depth map

To avoid the cost of data annotation, the prototype posture detection system employs a basic background subtraction technique for hand segmentation as opposed to the ML approach described in Section 4.2.1. The goal of background subtraction is to create a foreground image in which the piano and any other objects other than the hands are removed from the scene. Once the hands are isolated, the location of each hand is identified. To accomplish this, I take advantage of the RGB camera that is often included with depth cameras. Using the color data, the hands are segmented through a combination of background subtraction and thresholding.

To account for variation in practice spaces, the first step of segmentation is to remove the piano and other static objects from the scene. This is done by generating a foreground mask using Gaussian Mixture Model based background subtraction (Zivkovic and van der Heijden, 2006). Next, morphological opening (i.e. erosion then dilation) is applied to the mask to remove noise objects. The foreground is obtained by applying the generated mask to the original depth map.

Depending on the range of the camera, additional data such as the pianist's legs may be included in the foreground mask, as seen in Figure 4.3a. To remove these additional objects, I take advantage of the fact that the hands will be the closest object to the camera. Since the piano has already been removed form the scene via background subtraction, there is a gap between the hands and thighs (which are the next closest object). Using this observation, thresholding is performed to remove depths greater than the depth at that gap. The thresholding value is obtained by finding the bin at the local minima after the first peak of a depth histogram (not including bin zero). Applying the threshold generates a foreground containing only the hands.

Once the hands are segmented from the depth data, the location of each hand needs to be identified. First, a smoothing operation is applied to the image using a Gaussian blur for noise reduction. Next, the bounding box of each hand is derived through Canny Edge detection followed by a contour analysis of the edges (Canny, 1986). The final output is the bounding box coordinates for each hand. Figures 4.3b and 4.3d show images of hands segmented using this process with each of the depth cameras being tested. The segmented right hand is colored blue and the left hand is colored green.

Feature Extraction

As discussed in Section 4.2.2, HOG and HONV are compared as feature descriptors for building the hand posture detection models. In the prototype phase, I make a slight adjustment to each descriptor to account for variation in the size of the extracted hand depth maps.

Due to the varying state of the hand while performing, bounding boxes may change in size from frame to frame, which makes it difficult to use the

standard block approach in this work. For example, while playing a pianist may need to stretch their fingers to reach keys, thus, making the detected hand image wider than normal. The varying width of the hand means that histograms cannot be calculated using standard blocks without scaling all depth maps to be the same size. There is much less variability in the length of the hand while playing with a specific hand posture. For example, the wrist is usually the same distance from the longest fingertip. To account for the variable hand width, instead of using sliding blocks with scaled depth maps, histograms for the features of each hand are calculated using horizontal slices of the detected hand image. Furthermore, normalization only occurs at the cell level rather than applying block normalization which does not work with this customization.

4.3.2 Experiments

To initially test the proposed approach for hand posture detection, data was collected by recording the performances of two pianists as they played material selected from lesson plans for beginning piano students. Two depth cameras were tested for data collection, the Kinect and the Realsense. The Realsense is optimized for close-range interaction allowing for much more precise depth measurements, whereas the Kinect is designed for longer range tracking. While the Kinect has been shown to work for hand identification in piano playing by Hadjakos (2012), it is not clear if the depth data used with their system is precise enough to model the subtle differences of hand postures. To help improve close range precision, the *near-mode* option is enabled for the Kinect, affording a camera position closer to the piano. For these experiments, separate hand posture detection models are trained for each the left and right hands. Each model is a support vector machine (SVM) with a linear kernels implemented using Scikit-learn (Pedregosa et al., 2011). A one-vs-all strategy is implemented for multiclass classification.

Data Collection

Implementing hand posture assessment with a data driven approach requires a data set composed of recordings of pianists playing the piano

with their hands held in the three postures being detect. Data acquisition for such a data set can be expensive and time consuming. Before going through this process to create a real world data set from real piano students, an artificial data set was created by recording two experienced pianists playing piano exercises. Pianist, P1, is a piano teacher that plays at an advanced level and Pianist, P2, plays at an intermediate level.

The pianists performed exercises for beginning piano students, they played each exercise three times, and each time they performed the exercise with a specific hand posture simulating the errors that students with make. Depth recordings were captured for each of the exercises. In the first session, P1 is recorded with the Realsense depth camera. For sessions two and three, the Kinect was used to record P1 and P2 respectively. In each session, the pianist performed the following set of exercises.

During the first session P1 performed four different exercises with three hand posture categories. Exercise **A** consisted of P1 holding their hands in static pose for each category (correct, flat hands, and low wrists). For exercise **B**, P1's hands were held in a static pose but this time with keys pressed. In exercises **C** and **D**, motion was added for a more realistic dataset: exercise **C** was a C major scale and exercise **D** was a technical exercise from the popular piano lesson book series *A Dozen a Day* Burnam (2005). For the second and third sessions, by P1 and P2, respectively, exercises **E** and **F** from the same lesson book were added and performed in each posture category.

Results

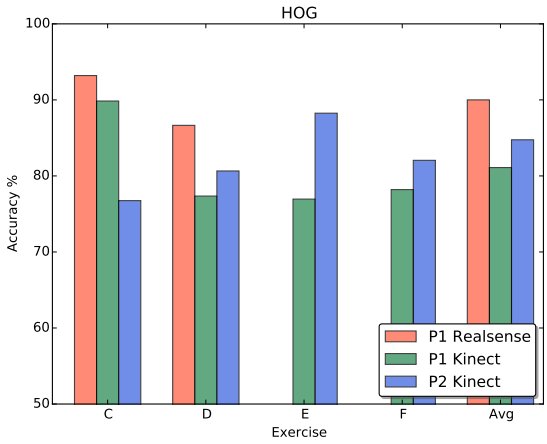
To validate my proposed CV pipeline, separate SVM posture detection models are developed from data of each of the three recording sessions: P1 with a Realsense, P1 with a Kinect, and P2 with a Kinect. The models are then evaluated using 5-fold cross validation. For each recording session, separate left hand and right hand data sets are created by processing each frame of the recordings through through the hand segmentation and feature extraction pipeline resulting in a set of feature vectors for the left hand and a set of feature vectors for the right hand. Both HOG and HONV data sets are created.

The generated data sets are then used to train and test individual SVM posture detection models. Testing is done using 5-fold cross validation. Additionally, to evaluate the potential for hand posture detection models trained with data from multiple sets of hands, the cross validation process was also performed on a single data set composed of data from sessions two and three with the Kinect. The average cross validation results for each session and the combined data set are are shown in Table 4.1.

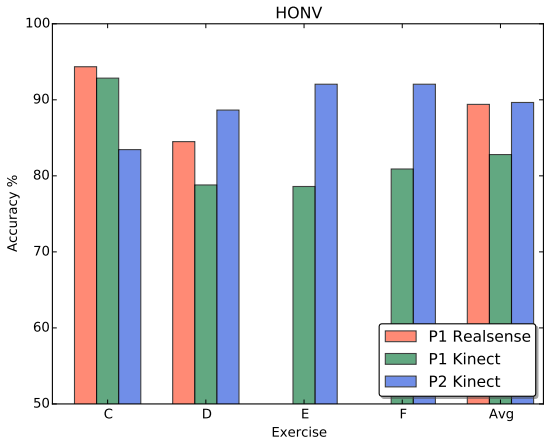
Session	HOG	HONV
P1 Realsense	94.8%	93.4%
P1 Kinect	92.4%	93.6%
P2 Kinect	97.2%	98.9%
Combined Kinect	93.7%	96.0%

Table 4.1: 5-fold cross validation accuracy averages for each session

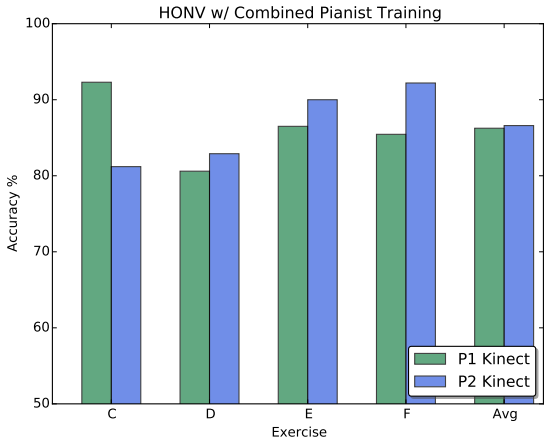
While the results are promising, cross validation using individual frames is prone to overfitting. Because each frame represents only a fraction of a second, neighboring frames will have little variation. Using cross validation, neighboring frames end up divided between the test set and training set so it is likely that data in the testing set was already seen by the trained model. To reduce the potential of overfitting, I next evaluate the performance of models trained with only static hand postures. For this evaluation, separate models are trained for each session using only recordings of the static hand poses, exercises **A** and **B**. Hand posture predictions are then made for each frame of all remaining exercises using the trained models. Posture detection accuracy rates of each exercise averaged over both hands using the HOG and HONV feature sets are shown in in Figures 4.4a and 4.4b, respectively. Figure 4.4c shows the results of detecting hand posture for each pianist, P1 and P2, using models trained with data combined from the static hand postures of both pianists recorded by the Kinect. Confusion matrices for all three sessions using HONV are shown in Figure 4.5.



(a)

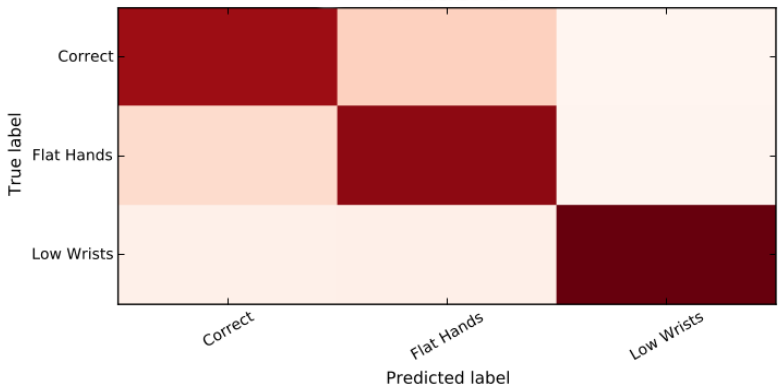


(b)

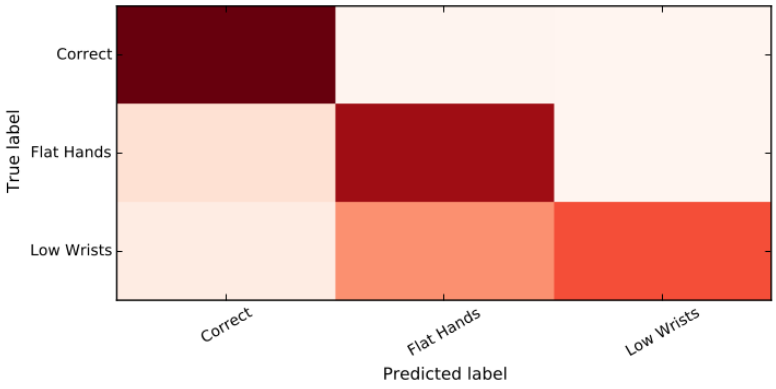


(c)

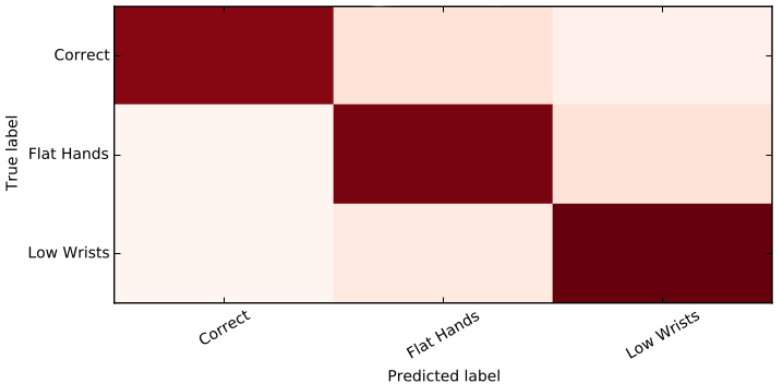
Figure 4.4: Hand Posture Detection Accuracy Rates



(a) Results from P1 with Realsense data



(b) Results from P1 with Kinect data



(c) Results from P2 with Kinect data

Figure 4.5: Normalized confusion matrices for each session using HONV

4.3.3 Discussion

The experiments show the effectiveness of the presented approach for hand posture detection. Additionally, the results of the combined model trained with multiple hands (Figure 4.4c) show the potential for a generalized posture detection model to replace individualized model training.

While performing a detailed visual analysis of misclassified images it was observed that some errors occur due to poor hand segmentation. Additionally, the background subtraction method used is not robust to varying conditions. Therefore, a more robust model with better accuracy is needed. As discussed in Section 4.2.1, RDF per pixel classification has been shown to work for hand segmentation in the current state-of-the-art hand pose recognition Tompson et al. (2014). My approach, however, poses the unique challenge that there are two hands that are in direct contact with the piano making it difficult to distinguish hands from piano where contact is made. Kang et al. (2016), however, demonstrated the per pixel approach is effective for segmenting individual hands interacting with objects they model is trained on.

The results show some variation in accuracy between each session and exercise. This is due in part to a limitation of our data collection process. To guarantee data for each hand posture, the pianists were asked to deliberately play with specific hand postures for each exercise. This presented a challenge to the pianists since poor posture is not natural. This is shown in the confusion matrix for the P1 Kinect session, Figure 4.5b, where the Low Wrist posture was often misclassified as Flat Hand. A visual analysis of the errors shows that during training P1 exaggerated the Low Wrist posture but had a difficult time keeping this posture while performing.

Evaluation of prototype implementation of the hand posture detection system validated the feasibility of the proposed approach. There are, however, some limitations that need to be addressed for a practical implementation:

L_1 Color based hand segmentation may not be robust enough for a realistic data set with more variation in hand characteristics,

L_2 Using experienced pianist to create the data set resulted in exaggerated error postures; thus, results may not translate to a realistic data

set where the differences in postures are more subtle.

4.4 System Description

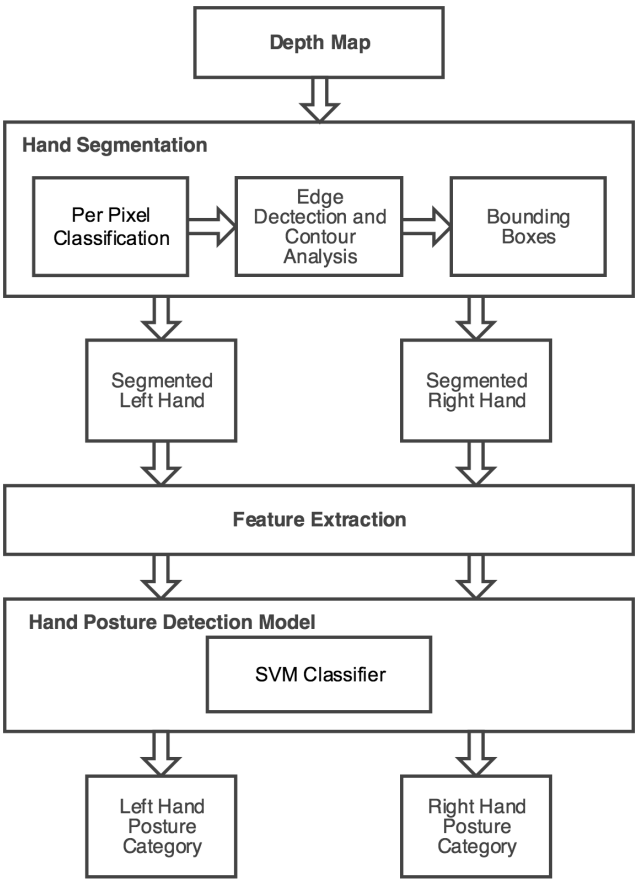


Figure 4.6: The posture detection pipeline used for assessing hand posture from single depth maps.

The validation of hand posture assessment prototype showed the feasibility of the proposed approach but with two limitations, L_1, regarding the robustness of hand segmentation, and L_2, regarding the quality of the data set. To address L_1, I employ a new method for hand segmentation, described in Section 4.4.2; and to address L_2, I create a new data set by recording beginning piano students performing a set of practice exercises, described in Section 4.4.1.

For the final hand posture assessment system, I propose the image processing pipeline in Figure 4.6 to detect hand posture from single depth

maps. The first step in the pipeline is hand segmentation in which the left and right hands are individually identified in the depth map and the background is removed using per-pixel classification, discussed in Section 4.4.2. This results in two image masks for each of the right and left hand. Edge detection with contour analysis is performed on each of the masks to find the bounding regions of each identified hand resulting in two depth maps each containing an extracted hand. Feature descriptors, described in Section 4.4.3 are then extracted from the segmented hand depth maps and used to develop the hand posture detection model.

4.4.1 Data Collection

Using a data driven approach for hand segmentation and posture analysis requires a diverse set of data for hand segmentation and posture detection models. Since the system is being initially designed for beginners, I recruited six piano students between the ages of 9 and 12 for data collection. Recordings of two of the participants had to be discarded; one for technical issues resulting in corrupted data and the other because the student's body posture resulted in them occluding their hands.

Data was collected using an Intel Realsense SR300 depth camera. The Realsense uses a short range structured light system to measure depth values at a resolution of 640×480 pixels and additionally has an 1080i RGB camera. The camera is capable of providing synchronized color, depth, and IR data at up to 60 FPS with depth range of .2 to 1.5 meters (Carfagni et al., 2017). The ML algorithms described through this work exploit only the depth information. Color data was employed to generate hand masks used to annotate depth pixels as either left hand, right hand, or background. For each recording described below, I captured the depth data and color data at 30 FPS and a resolution of 640×480.

Using the Realsense, I recorded the remaining four students playing a variety of piano exercises. At the time of the study, Participant 1 (P1) was 12 years old, participant 2 (P2) was 11 years old, participant 3 (P3) was 9 years old and participant 4 (P4) was 11 years old. The exercises they performed ranged from basic scales to technical exercises from the popular piano lesson book series A Dozen a Day (Burnam, 2005). Isabelle Dufour,

assisted in the data acquisition process by watching the students perform and taking notes on their posture as they played. After acquiring the data, it then had to be annotated to develop hand posture detection model as well as the hand segmentation model.

To annotate the data for use in the hand segmentation model, a subset of the data was created by sampling depth maps from the recordings every second. This resulted in a set of 661 depth maps for training. For each depth maps, a masks were created to label the pixels as either left hand, right hands or background.

To annotate the data with hand posture labels, I developed an application to simplify the annotation procession. Isabelle then went through each recording using the annotation application to annotate the hands in the each frame. Both the right and left hands were labeled as one of the three posture categories described in Section 4.1.1.

4.4.2 Hand Segmentation

In this work, per-pixel classification using an RDF is employed to segment the left and right hands from the depth map scene. I experiment training the RDF with both depth image features (DIFs) (Shotton et al., 2011) and depth context feature (DCF) (Liang et al., 2016) to find the optimal descriptors. The rest of this section discusses, in detail, the process and descriptors used to isolate each hand from a single depth map.

Per Pixel Classification

The task of per-pixel classification is to predict a category for each pixel in an image or depth map. For each pixel, features are extracted and used to train a classification model, in this case an RDF. The rest of this section presents the process of training the RDF for per-pixel classification with three classes, the left hand, the right hand and the background, from a depth map scene containing both pianist's hands while playing the piano. First, I discuss the DIFs proposed by Shotton et al. (2011) for body part inference; then, I discuss the more recent DCFs proposed by Liang et al. (2014). Experiments of training the RDF with either of two feature descriptors are also presented, Section 4.5.1.

Depth image features (DIFs) (Shotton et al., 2011) are discriminative descriptors that compare the depth values of N pairs of pixels, in a neighborhood, to capture a representation of the surrounding context of a given pixel p. For each pair of p, two offset parameters, u and v, are randomly selected which are used to determine the pixel locations of each offset. Each feature is computed as the difference in depth values at each offset location calculated by

$$f_\theta(I, p) = d_I(p + \frac{u}{d_I(p)}) - d_I(p + \frac{v}{d_I(p)}) \tag{4.1}$$

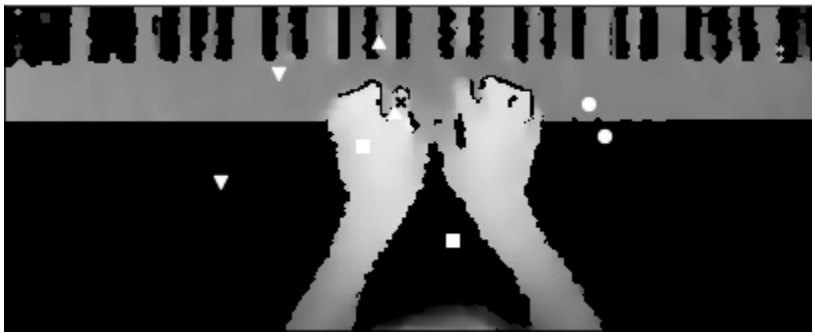
where $d_i(p)$ is the depth value of pixel p in image I. To ensure depth invariance the offsets are normalized to the depth of p using, $\frac{1}{d_I(p)}$. A large constant value is given to any offset pixel that lies on the background or outside the bounds of the image.

A set of offset parameters, u and v, are randomly sampled from a uniform distribution and used for each pixel in all depth maps. The range of the offset sampling affects the size of the neighborhood to examine; a small sampling range for the offset values represents a narrow context that is close to the pixel, whereas a large range increases the area being captured by the features.

Figure 4.7b shows an example of a subset of four randomly sampled pairs of offset locations for a pixel, marked with a black "x", located on the left index finger. Each pair of feature offsets is represented by a distinct shape. For each offset pair, the difference in depth is calculated using Equation 4.1. In practice, the number of offset pairs is much higher. Here I use a small value for the purpose of visualization.

Depth context features (DCFs) (Liang et al., 2014) are discriminative descriptors that provide a more structured approach to examining the context of a pixel's neighborhood. Instead of using randomly generated context points, Liang et al. (2014) assert that points nearer to the classification pixel better describe the context of the pixel compared to points that are further away. Thus, they propose a distance adaptive sampling scheme that samples offset pixels, or context points, more densely from points closer to the classification pixel. The distance of the context points from the current pixel is defined by maximum range value r and the the parameter M de-

(a) Original depth map



(b) A subset of DIF offsets

(c) DCF with $M = 4$ and $r = .15$

(d) DCF with $M = 5$ and $r = .15$

Figure 4.7: Examples of DIF and DCF offsets for extracting features of a single pixel in a depth map used to classify the pixel as either hand or background.

fines the number context pixels to sample in each direction. Figures 4.7c and 4.7d show examples of the selected context points using the distance adaptive approach with different M values, $M = 4$ and $M = 5$ respectively, for a pixel on the left index finger (marked with a black "x").

To handle depth invariance, depth context point offsets are defined in the 3D space rather than on the image plane. The location of the 3D context points relative to pixel p with 3D coordinates v can be defined as $v_d = [a_d, b_d, 0]^T$. Therefore, to find the pixel coordinates, p_c, the depth context point is projected back to the image plane with $p_c = \Psi_p(v + v_d)$. The feature value for a context point is thus calculated as the difference between the depth of the current pixel and the depth of the context points at the projected pixel coordinate:

$$f_\theta(I, p, v_d) = d_I(p) - d_I[\Psi_p(v + v_d])], \tag{4.2}$$

where $d_I(p)$ is the depth at the given pixel as found in the depth map.

Random Decision Forest Classification

To predict a category for each pixel in a depth map, RDF classification is employed. An RDF is an ensemble classifier composed of T decision trees whose predictions are aggregated using votes weighted by the posterior probabilities to make the final prediction. Each decision tree, t, is composed of split and leaf nodes. A split node contains a feature and threshold value used to determine the branching direction. And a leaf node contains a learned probability distribution $P_t(c|I, x)$ for labels c where I is the image and x is the pixel to classify.

To train an RDF, a random subsample of the training data (sampled with replacement) is selected to train each tree in the forest. Additional randomness is applied when finding the split parameters of a node during construction of an individual tree. At each node, a random subset of features is selected for consideration when calculating the criteria for splitting. This approach helps to improve accuracy and reduce over-fitting (Breiman, 2001).

For per-pixel classification of depth maps in my proposed system, training samples for each pixel category (left hand, right hand, and background)

are generated by randomly sampling N pixels representing the respective category from each annotated depth map. Feature values, either DIFs or DCFs are then calculated for the sampled pixels. This is done for each depth map in the training data to generate a complete training set for the RDF.

To segment the hands from the depth map, each pixel is assigned a label by evaluating all trees in the forest and calculating the weighted average using,

$$P(c|I,x) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|I,x). \tag{4.3}$$

A label l is then assigned to each pixel x of image I by $l = \arg\max_c P(c|I,x)$.

Per-pixel classification results in an image mask in which every pixel has a label l representing left hand, right hand or background. The masks are then denoised to reduce mislabeled pixels. For each hand label, a bounding box is calculated to identify the hand region. Each region of the original depth map is then extracted into a new depth map with the background removed. This results in two new depth maps, one for each hand, in which the pixels with depth values greater than zero correspond to the identified hand.

4.4.3 Hand Posture Detection

Feature Extraction

After the hands are segmented from the original depth map, the next step in the hand posture assessment pipeline is to extract feature descriptors from each depth maps. The descriptors capture information about the shape and geometry of the hand which is used to train a posture detection model. During feasibility assessment, described in Section 4.3, the performance of two feature descriptors, HOG (Dalal and Triggs, 2005; Felzenszwalb et al., 2008) and HONV (Tang et al., 2012), was compared and found to be similar. In the new data set, the smaller hands of the beginning students result in more subtle differences between hand postures so previous descriptor performance may not transfer. Therefore, I experiment using both descriptors for SVM based classification of hand posture. In this case, however,

instead of using the customized vertical slices for histogram calculation, as discussed in 4.3.1, I employ standard method of cell based histogram calculation. This requires that all depth maps are scaled to the same size. In section 4.5.2 I experiment with various scaling techniques to ensure that information lost through image scaling isn't a factor in model performance. Furthermore, for this phase I employ *L1-sqrt* for block normalization.

Training Student Specific Models

The extracted descriptors, as described above, are used for training the posture detection models. Developing one generalized model for all beginning piano students requires a large data set covering a wide variation in hand characteristics as well as a variety of hand positions and postures. With the limited amount of data collected due to resource limitations, I chose to develop student specific hand posture detection models.

For each student, I train individual SVM classifiers using the depth maps that have been obtained through the hand segmentation process. Right hand depth maps are all flipped horizontally to have the same orientation as the left hand. This transformation affords one data set per student, rather than a data set for each hand. The final student data set is then composed of feature vectors, representing the image descriptors, for each hand depth map. This data set is then used to train the SVM for the corresponding student. Validation of the individual models using cross validation and various train-test split configurations is discussed in Section 4.5.2.

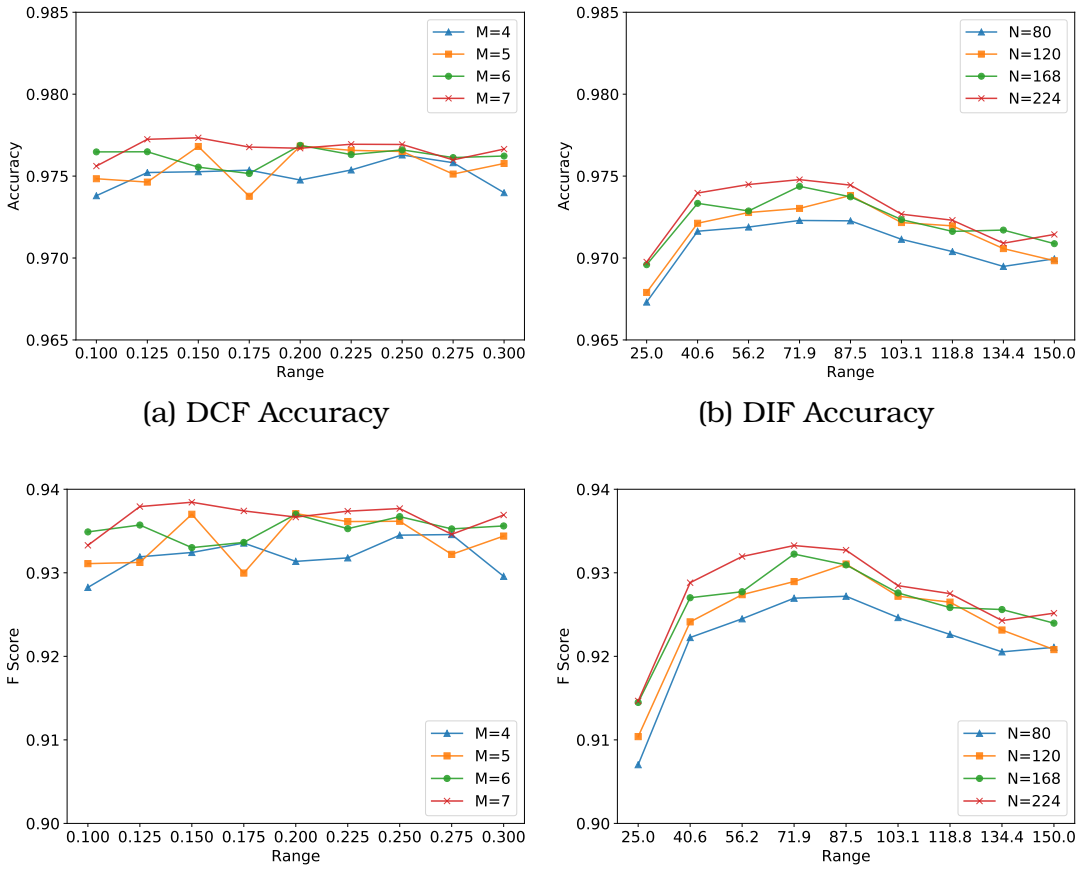
4.5 Experiments

This section presents the results of experiments validating the main components of the hand posture assessment pipeline: hand segmentation and hand posture detection. Evaluation of the proposed approach for hand segmentation is presented in Section 4.5.1 and evaluation of the proposed approach for hand posture detection is presented in Section 4.5.2.

4.5.1 Hand Segmentation

To test the hand segmentation approach discussed in Section 4.4.2, experiments are performed comparing DIF and DCF descriptors and their various parameters. An RDF classifier, implemented using Scikit-Learn (Pedregosa et al., 2011) and configured with 10 trees with a maximum depth of 20, is employed for the experiments. The rest of this section presents the results of the hand segmentation approach on the piano student data set using the two feature descriptors.

Results



(a) DCF Accuracy

(b) DIF Accuracy

(c) DCF F-Score

(d) DIF F-Score

Figure 4.8: Per pixel classification results of hand segmentation using DCF and DIF with varying radius and neighborhood sizes.

The performance of each feature descriptor for hand segmentation is

evaluated by tuning two main parameters which affect offset point selection. DCF include a parameter, M, which influences the density of points sampled from the neighborhood (i.e. number of features per pixel), and a range parameter, r, which affects the size of neighborhood being sampled. DIF have similar parameters; N, the number of features and the range, r, indicating the size of the neighborhood for offset selection. Different values for each the parameters will have varying effects on the performance of the segmentation model and should be evaluated.

Different values of M and N are evaluated for DIF and DCF, respectively. M and N are analogous features affecting the resulting number of points selected per pixel within the specified range r. M is a value that affects the grid size for the DCF context points and correlates with the total number of features for the given pixel, p. N, on the other hand, is the number of offset pairs to calculate $f_\theta(I, p)$, as in Equation 4.1 and also corresponds to the number of features in the resulting feature vector. Figure 4.7 shows examples offset and context points for both feature descriptors. For an equivalent comparison between the two descriptors, I choose values for N such that it equals the number of features for each M tested.

To evaluate the effects of each descriptor's range value, I experiment with different values of r that range from a very detailed region of the depth map close to the evaluated pixel and a large r that covers a larger neighborhood with offset points that spread further away from a given pixel. The r values for the respective descriptors use different units, but I attempted to choose corresponding r values for each descriptor that were comparable through empirical observation.

To ensure that the segmentation models are not over-fitting to data that has already been seen, I use a participant-based leave one out cross validation technique. In this scheme, I train the segmentation models with all but one participant's data and use the left out participant's data for testing the model. The classification accuracy and F-score of each round of cross validation are then averaged. Figure 4.8 lists the results for each descriptor and with the varying parameter values.

Overall, the results show that DCF consistently perform better in terms of both classification accuracy and F-score. Furthermore, the plots show that the greater the number of features generally results in better classi-

fier performance. These parameters not only effect the performance of the segmentation model, but also have implications on the computational efficiency of the segmentation process. Finding the optimal parameters thus requires a balance between accuracy and runtime performance. For real-time prediction, values that balance both accuracy and prediction time are ideal. Based on observed evidence of runtime the number of calculated features is directly correlated with runtime (although a more detailed experiment on runtime is needed to confirm this). For this reason, I find DCF descriptors with $M = 5$ and $radius = .2$ to be the best configuration for the hand segmentation model. These values will be throughout the following experiments validating the full hand posture detection approach.

4.5.2 Hand Posture Detection

In this section, I describe experiments for parameter tuning and evaluation of hand posture detection models trained using both HOG and HONV descriptors. The first set of experiments evaluates the effects of tuning different parameters toward classifier performance. In this case, individual student detection models are trained and evaluated using a cross validation scheme. In the second set of experiments, I train and evaluate the models using a leave one exercise out approach to reduce the effects of overfitting. Due to the high dimensionality of the data, an SVM classifier was employed for posture detection in all experiments.

Hyperparameter Tuning

Hyperparameter tuning is the task of evaluating and selecting the optimal values for the detection models. In this case, I am tuning the parameters that affect the resulting features for training the SVM. Specifically, I perform experiments to test changes in depth map size and aspect ratio as well as the cell and block size for HOG and HONV histogram calculations.

To train and validate the models in this section, the data is segmented into one second windows which are then distributed into training and testing sets using 3-fold cross validation. This scheme is meant to reduce the over-fitting effects seen with standard cross validation in which neighboring frames, that have very little variation, may be split into the training and

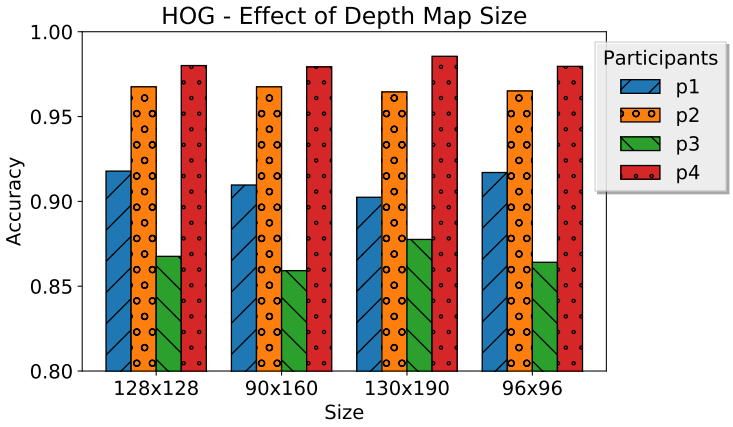
testing data. The rest of this section describes the different experiments that were performed to validate the the approach of using student specific models for hand posture detection.

Depth Map Size: General image processing algorithms often rescale input images to a standard size, such as 128×128, to calculate feature vectors of a consistent size. Hand posture detection may require more fine grained information than general object recognition becuase of the subtle differences in postures. Therefore, rescaling the segmented hand depth maps to different aspect ratio could lead to information loss or hand deformation with negative effects on detection performance. Selecting a consistent size and aspect ratio is a challenge with hand posture detection because a pianist's hands are always changing position which results in variations in the aspect ratio of the extracted hand regions from frame to frame. In the piano student data set, the average aspect ratio of the segmented hands is $9 : 16$. Rescaling to this ratio may represent the shape of the hand more accurately but could result in information loss and hand deformation for depth maps that are not naturally this ratio. I experiment with this ratio as well as two square aspect ratios. Further, the largest hand region was found to be 130×190; to keep all of the original hand data of each depth map, I experiment with increasing all depth maps to this size by padding the front of each axis with zeros resulting in a consistent size without rescaling the image. For this experiment I employ a default cell size of 8×8 and a default block size of 3×3 and an SVM with a linear kernel.

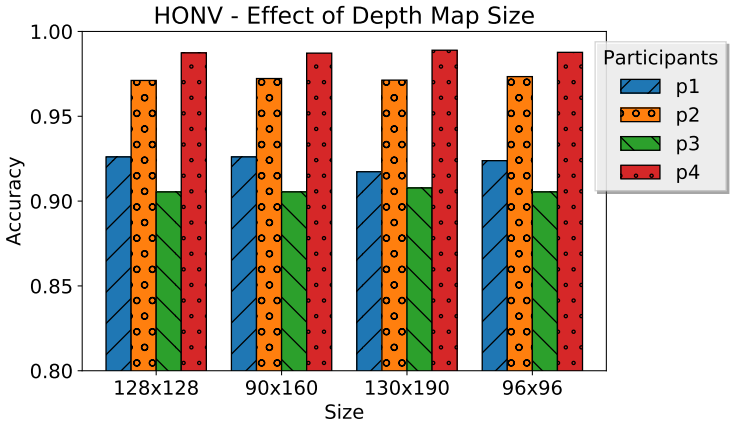
	128×128	90×160	130×190	96×96
HOG	93.3%	92.9%	93.3%	93.1%
HONV	94.7%	94.8%	94.6%	94.8%

Table 4.2: Average hand posture detection accuracy for each depth map size per descriptor type

As shown in table 4.2, the various depth map sizes appear to have limited effect on prediction accuracy, with HONV outperforming HOG in all cases. Figure 4.9 shows the prediction results of the depth map sizes for each participant. Here it is shown that while the depth map sizes have limited effect on accuracy, HONV significantly improves the accuracy for the hardest case participant. P3 benefits from an average 4.5% increase in ac-



(a) HOG



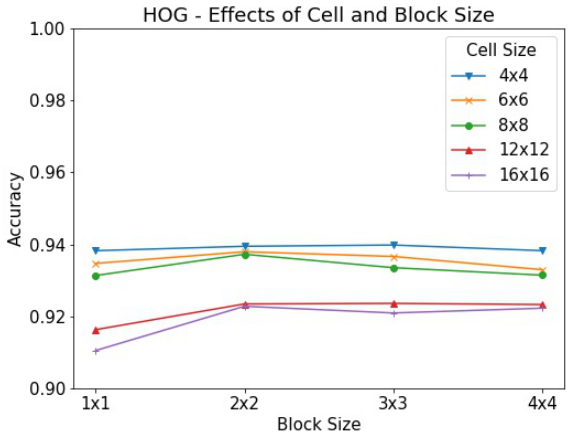
(b) HONV

Figure 4.9: Individual participant posture detection accuracy of different depth map sizes when using HOG and HONV descriptors

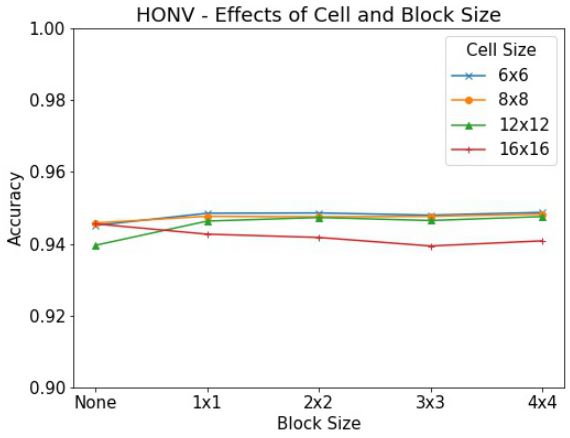
curacy using HONV. As depth map size appears to have negligible effects on performance I use a scaled size of 128×128 for the rest of the experiments.

Cell and Block Size: Dalal and Triggs (2005) found the optimal HOG cell size detecting humans in images to be 6×6 and the optimal block size to be 3×3. When detecting humans in RGB images, HOG create a representation of humans through edge shape. Using HOG with depth maps for hand posture detection,on the other hand, creates a geometric representation of the hand surface in addition to identifying edge shape. With different underlying representations between RGB images and depth maps, its not clear that the parameter values of Dalal and Triggs (2005) will translate to

the problem of hand posture detection. Further, Tang et al. (2012) used only a cell size of 8×8 in their work and do not employ block normalization. To understand what sizes work best for hand posture detection, I evaluate the performance of different cell size and block sizes for both descriptors. Additionally, I evaluate HONV with and with block normalization since Tang et al. (2012) did not implement this method from the work of Dalal and Triggs (2005).



(a) HOG



(b) HONV

Figure 4.10: Hand posture detection accuracy for HOG and HONV with different cell and block sizes

Figure 4.10 presents results of the evaluation of detection models using a range of cell and block sizes for each descriptor. Results show that a

smaller cell size improves detection performance in nearly all of the cases. Due to computational resources I omit a cell sizes of 4×4 for HONV. Since HONV uses 2D histograms, the resulting feature vectors require significantly more space than HOG. Similar to Dalal and Triggs (2005) block sizes of 2×2 and 3×3 work best with HOG. HONV, on the other hand, shows some benefit from normalization but is less affected by block size. As HONV has shown the best overall performance regardless of parameter values, I use the HONV descriptors for the rest of the experiments. A cell size of 8×8 is implemented since to balance performance and feature vector size for increased runtime performance. Further, I normalize cells individually using blocks of size 1×1.

Exercise Based Training and Oversampling

In the previous experiments, cross validation was performed by partitioning the data into one second windows and splitting the windows into training or testing sets. This approach is prone to some overfitting as frames from the same recording will be in both the training and testing data sets. To evaluate the student specific detection models in a more realistic way, in this section I employ a leave-one-exercise-out approach for cross validation. In this approach, cross validation is performed for each student by training the model with four of the five exercises and then validating the model using the left out exercise. This is repeated so that each exercise is left out once, and the results of each iteration area are then averaged together. For this experiment, HONV descriptors with a cells size of 8×8 pixels and a 1×1 blocks size are employed for feature extraction. The features are used with an SVM classifier configured with an RBF kernel and hyperparameter values, $C = 10$ and $\gamma = .01$.

A challenge in developing customized detection models for individual students is a lack of control over the number of samples collected per posture class. This potentially leads to an imbalanced data set where at least one class has a far greater number of samples than the other. Table 4.3 provides an overview of the category counts per participant. This table shows that each participant is prone to different posture class distributions with some having relatively few samples.

	Correct	Low Wrists	Flat Hands
P1	6011	162	1021
P2	3917	1336	47
P3	2262	3376	0
P4	6111	286	27

Table 4.3: Class counts per participant

Two common methods to balance data are majority undersampling and minority oversampling in the feature space. Undersampling is not a good idea in this case because it would require the data to be downsample to the size of the smallest class. In this case the data would not be large enough to train a robust model. Instead I employ oversampling to balance the data and evaluate two techniques: Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) and Adaptive Synthetic Sampling (ADASYN) (He et al., 2008). Rather than simply oversampling with replacement (i.e. copying existing feature vectors), SMOTE generates synthetic samples by calculating a new feature vector that lies between a minority sample, x_i, and a neighbor, x_{zi}, selected randomly from the k nearest neighbors of x_i. Features for the generated feature vector are calculated using $x_{new} = x_i + \lambda(x_{zi} - x_i)$ where λ is a value between between 0 and 1 selected randomly for each sample . There are four variations of SMOTE that affect the selection of minority samples to use for sample generation. Regular SMOTE simply employs a random selection from all possible minority samples (Chawla et al., 2002). The Borderline-1 and Borderline-2 SMOTE variants classify minority samples as *in danger* if less than half the neighboring samples are from the same class. The *in danger* samples are then selected to use for new sample generation (Han et al., 2005). SVM SMOTE takes the support vectors of a trained SVM into consideration to select the samples used for new sample generation (Nguyen et al., 2011). ADASYN (He et al., 2008), similar to SMOTE, uses interpolation to generate new samples but is biased to select samples that are harder to learn. In other words, more synthetic samples are generated for samples that are hard to learn, effectively adapting the decision boundary towards the hard to learn samples. Oversampling, using either SMOTE or ADASYN, generates a balanced data set to reduce learning bias when training posture

detection models.

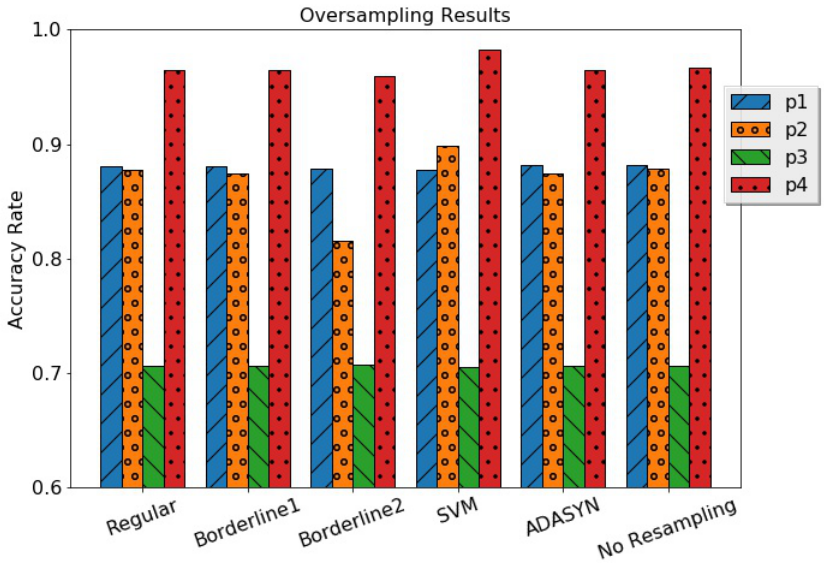


Figure 4.11: Hand posture detection accuracy with different oversampling methods.

Figure 4.11 shows the results of student specific hand posture models using using each of the SMOTE variants and ADASYN to balance the data sets. Most of the oversampling variants have little effect on performance of the inference models, but there are a few exceptions. The SVM SMOTE variant shows improved accuracy for P2 and P4 and Borderline 2 shows a decrease in accuracy for P2. P3 shows little change with each technique because the data was already well balanced between two classes. Further, a review of the participant-based confusion matrices from models trained with SVM SMOTE, Figure 4.13, compared with the confusion matrices for models trained with no oversampling, Figure 4.12, shows that SVM SMOTE improves prediction for certain minority classes. For example, there are improvements in the flat hands class for P1, as well as the low wrists class for P4. In cases, where the number of samples is substantially smaller than the majority class, oversampling does not provide an improvement.

The results of the exercise-based cross validation scheme, show a working posture detection model can be achieved using as little as four exercises for training. This is dependent upon the severity and occurrence frequency of a student's errors in posture. In cases where the third posture category

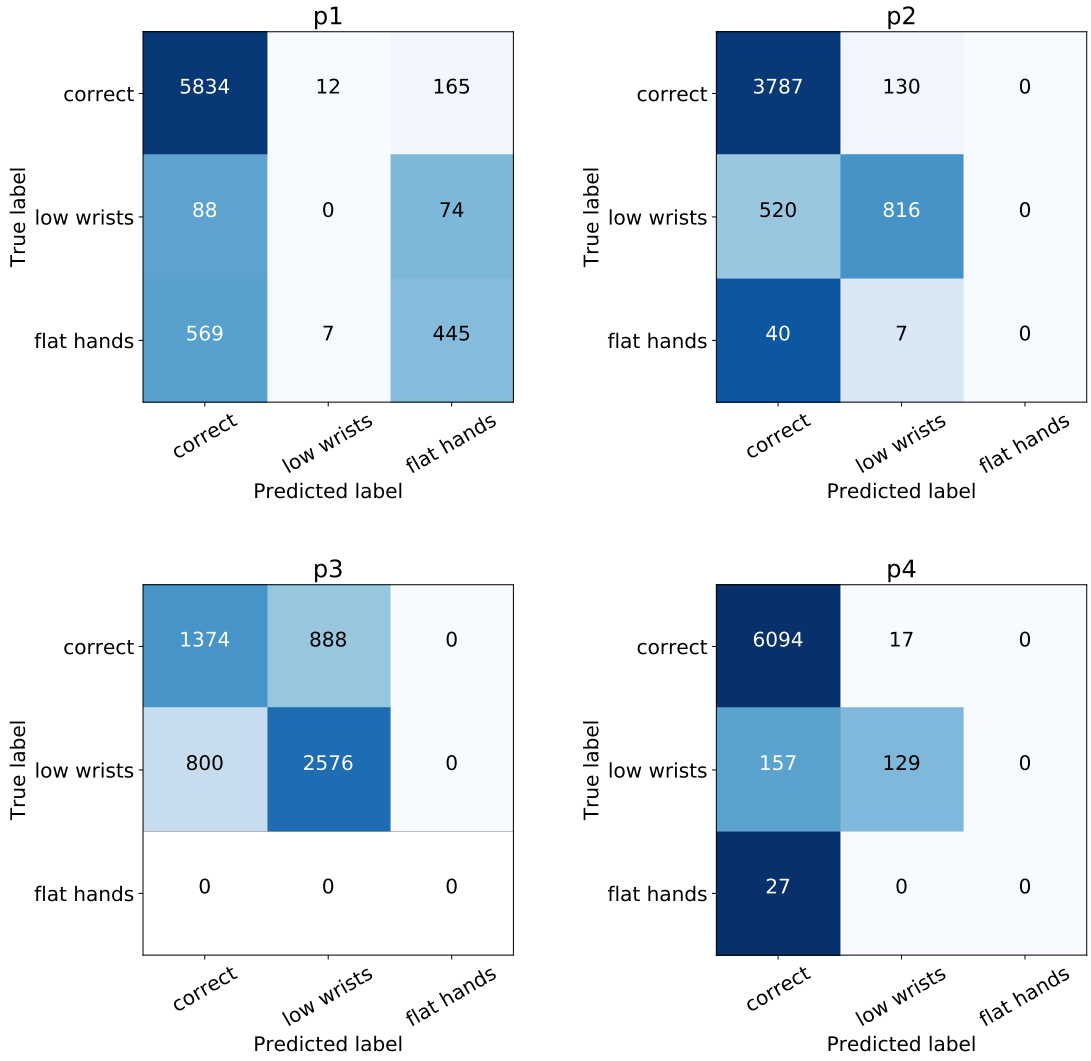


Figure 4.12: Confusion matrices for each student posture model trained without oversampling

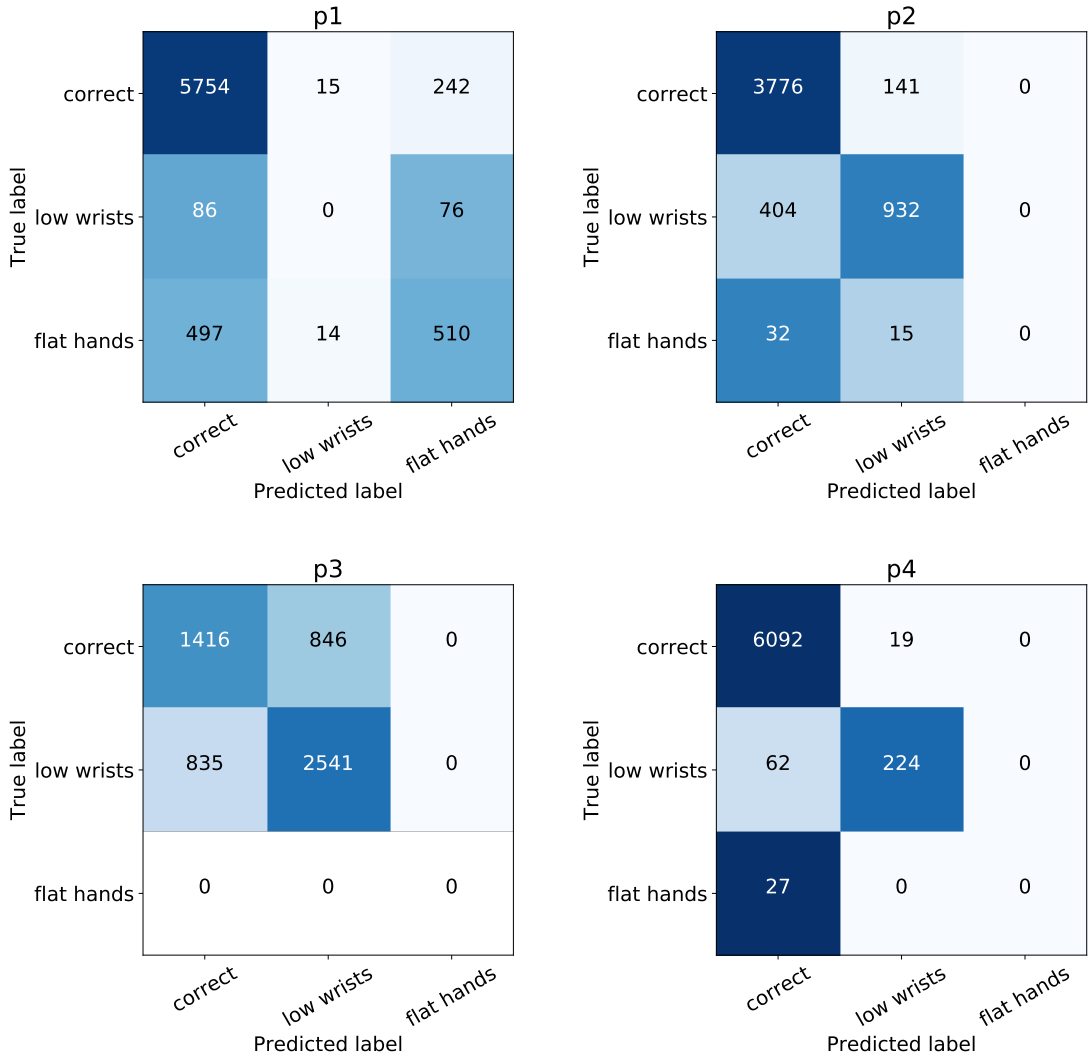


Figure 4.13: Confusion matrices for each student posture model trained using SVM SMOTE oversampling

is too small, the model could instead be trained as a binary classifier until enough samples of the category are recorded. For piano students that make only a few errors, a data balancing techniques, such as SMOTE or ADASYN, are needed. These factors should be taken into consideration when designing an interface for training the detection models.

4.6 Discussion

This chapter presents and evaluates an approach for the automatic assessment of pianist hand posture using data recorded with a depth camera. Implementing this system into a CAMIT interface requires converting assessment output, i.e. the detected hand posture class, into feedback that is presented to the student. In Section 4.6.1 I discuss my thoughts on the design of such an interface. Further, employing student specific detection models raises a number of implications for system design, I discuss the implications with possible solutions in Section 4.6.2.

4.6.1 Considerations for Interface Design

Based on a review of the CAMIT literature, I find there are three main techniques that may be employed to provide beginning students feedback about their performance: real time feedback with auditory cues (Ferguson, 2006; Ng et al., 2007), video playback of a practice session augmented with visual feedback (Ng et al., 2007), and assessment scores and visualizations on the quality of a performance (Blanco and Ramirez, 2019). To be effective, a CAMIT interface must be motivating, informative, and help the student improve. When choosing a feedback method, a number of aspects must be taken into consideration: the amount of information presented to the students, the student's ability to understand and process the information, and the robustness of the assessment system for providing correct information.

The simplest feedback method is to provide the student with a single score or visualization indicating the quality of their performance. To assist with self-evaluation, this method would allow a student to compare their performance assessment during a practice session to the their performance of a previous session or to that of an expert. Furthermore, this design

would allow a teacher to quickly track students' progress through sessions in which the teacher is not present. One of the benefits of this method is that it may be one the easiest for a student to understand, making it ideal for young or beginning students. Additionally, using a score-based method would support a tutoring system with gamification to motivate students. Technically, the scoring method would be the easiest to implement because it is the least vulnerable to posture classification errors as improvement is relative to past performances and minor errors in classification would not be noticed as explicitly. The main drawback is lack of context to indicate which mistakes were made and when they were made. Without the detailed information a student may not know exactly how to improve their performance, especially if a teacher is not available. Previous research of a visual feedback system for performance quality, however, shows this method to be potentially effective for improving performance (Blanco and Ramirez, 2019).

A more informative approach to presenting performance feedback is video playback of the performance, augmented with visual indication of posture errors. With this method, students would be able to view exactly when and how mistakes were made. Furthermore, as opposed to real-time feedback, the student would be able to analyze their performance while not focused on the other cognitively demanding aspects of practice, such as playing the correct notes. There are some challenges to using such as system though. Namely, the detection accuracy must be near perfect as detection errors may adversely affect a student's ability to self-evaluate. Furthermore, students (especially young ones) may find watching a recording of their performance to be boring and demotivating.

Providing real-time feedback, instead, may address motivation issues by integrating feedback directly into the practice session. One method for real-time feedback is to use auditory cues. Providing real-time auditory cues to signal an identified mistake is already familiar to students because this is similar the style of feedback they would receive during training sessions with a teacher. A system for beginners should only alert the student to an issue after a specific period of time playing with poor technique as continuous feedback may be too cognitively demanding. With this method, once the detection system recognizes that a student performed with incorrect posture for a number of seconds, it could trigger an auditory alert,

such as ”remember to keep your wrists up.” This method would help a student self evaluate by receiving auditory cues exactly at the moment they occur allowing a student to quickly adjust their technique. While real-time auditory feedback may be cognitively challenging, it is most similar to the feedback they are already receiving from their teacher.

Another technique for real-time feedback is the use of visual cues on a computer display. This is the technique used by market available CAMIT systems such as Yousician (2019). Such applications require students to constantly shift focus between the music instrument and the application to get feedback which could be missed if they don't shift focus quickly enough. Further, these systems typically also provide visual guidance for performance adding to the cognitive demands of trying to process multiple streams of visual input while also focusing on auditory output. As technology in XR advances, implementing immersive systems with RTVF may also prove to be effective at overcoming these challenges. For example, the hand posture detection system could be integrated with a XR environment in which the student would be presented with computer generated visual cues overlaying their hands providing direction, such as arrows pointing up or down, for posture improvement. While there has been some research developing these type of systems, it has yet to be seen how effective this method is for musical tutoring. More research is still needed to provide guidance on how best to design XR interfaces for real-time hand posture assessment and correction.

4.6.2 Future Work

This study lays the groundwork for an automatic assessment of hand posture to enhance piano pedagogy for beginning piano students but there are still two main challenges to address. First is accuracy of the information provided by the detection system and the robustness to variations in hand formation not related to posture. The data used in the experiments was taken from typical exercises of beginning students so there is only minimal variation in hand movement and deformations, such as the lateral spread of the fingers. Thus, the detection system as is, may not be robust to more advanced techniques required as students improve. Secondly, employing a

per user training scheme requires effort from the teacher and the student to train the model before use. If too much effort is required for training, the system becomes impractical. I leave these challenges for future work but discuss possible methods for addressing them here.

One potential solution is to build a larger data set with greater variation in hand shape and playing style to improve the generalization of the detection model. One of the biggest challenges with machine learning, however, is that building generalized models requires large scale data sets, for example, one of the largest data sets use in machine learning research, ImageNet (Deng et al., 2018), now has over 14 million images (ImageNet, 2010). This is especially true when working on new problems that have little or no existing data and that require domain experts, i.e. piano teachers, for annotation. Furthermore, student hand posture errors may not be limited to those presented in this article. While this could be addressed through one-class classification, in which training is performed using only correct posture, such a system would not be able provide a student information about how to correct errors. To address the challenges related to large scale data collection, I propose a per-user training system for posture detection in which the student, teacher, and interface work together to train the posture detection system. I have shown with this research that it is possible with limited amounts of data.

A per-user training scheme has the benefit that detection models are able to be customized to each student's skill level and overcome the challenges in obtaining enough data for generalization. Customization may be achieved by allowing teachers to define their own posture categories and choose the appropriate training exercises that match the students' skill levels and playing style. Giraldo et al. (2019) took a similar approach in their work on tone quality prediction to overcome challenges of subjectivity in tone perception. Per-user training is not without its drawbacks, however. Most notable is the fact that it takes time and effort from the teacher and student to train the models. If training is too arduous, such as labeling an entire recording, than the system will not be used. Additionally, teachers cannot be expected to be machine learning experts so a training system should be easy to understand. To address issues such as these, there is emerging work in human-in-the-loop machine learning, such interac-

tive machine learning (Amershi et al., 2014; Chen et al., 2018; Holzinger, 2016) and active learning (Settles, 2009), in which humans work directly with a training system to build and improve learning models. Active learning works by selecting samples to be labeled based on some criteria, such maximum uncertainty, then asking a human participant to label the selected samples. iML builds on this idea with a focus on designing interfaces in which humans work with the machine through iterative train-feedback-correct cycles to improve the learned model (Amershi et al., 2014). Integrating iML and active learning techniques into model training will help improve model robustness by making per-user training feasible and will help improve accuracy through iterative training cycles.

Assuming a model reaching near perfect classification of hand posture is developed, there are still times in which it may not be appropriate to analyze a student's hand posture, such as when the hand is in transition. While transitions are generally minimal for beginning students, providing posture feedback for for more advanced students should ignore these transient periods so as not to provide incorrect feedback. It may be possible to address this by adding an additional category to the posture detection model to identify hand positions that should be ignored. A more robust method may be to integrate gesture detection that tracks hand motion to first identify when hands are in an appropriate state for posture detection. Integrating these capabilities would improve when posture feedback is presented to the student.

4.7 Conclusion

In this chapter I present a system for detecting hand posture mistakes for piano students from a single frame of a depth camera recording. Using a per-pixel classification scheme for hand segmentation, I find that the DCF descriptors from Liang et al. (2014) result in the best segmentation accuracy. The higher accuracy, compared to DIF, is possibly due to DCF implementing denser sampling of context offsets closer to the pixel being classified. While positive results are achieved with the current sampling distribution, there are still misclassified pixels in areas where the hand

is in direct contact with the piano. To account for this, future work using a sampling distribution that is even more densely sampled near the classification pixel should provide more fine grained context for better classification. Once the hands have been segmented from the depth map, an SVM is employed to detect the posture of each hand. Evaluation of posture detection showed HONV depth map descriptors (Tang et al., 2012) provide the best performance. Further, HONV was improved by adding block normalization to the feature extraction process. To account for shape and size variations in hands as well as varying practice environments, I implement student specific hand posture detection models customized and trained for individual students. Using individual models, however, presents a problem as students will not always perform with an equal distribution of posture categories, resulting in unbalanced data for training. The experiments presented in this chapter demonstrate that this problem can be addressed using over-sampling in the feature space with SMOTE. Further, the results show the effectiveness of the proposed CV pipeline for student specific hand posture detection models. With this pipeline in mind, I have discussed thoughts on designing interfaces to provide feedback to piano students using the detection system as well as on designing interfaces using iML to improve the process of training individually customized detection models. The design of these interfaces is left for future research.

Chapter 5

Evaluating the Effectiveness of XREMIL

In the previous chapter, I presented a CAMIT system for the automatic assessment of pianist hand posture. Designing an interface requires an effective method for converting the assessment results into feedback for the student. XR affords new methods for implementing real-time visual feedback (RTVF) by overlaying musical instruments with visual cues but there is limited work that evaluates the effectiveness of such an feedback approach.

This chapter presents the results of a controlled user study evaluating the effectiveness of an extended reality enhanced musical instrument learning (XREMIL) environment with RTVF addressing RQ_2. The research methodology employed for this study requires a working XREMIL to evaluate. Before presenting the user study, the first part of the chapter discusses the design and development of a new XREMIL environment for learning the theremin that integrates a physical theremin with a virtual learning environment. Through this interaction design process, I examine affordances and guidelines for designing XREMIL addressing RQ_3. The work presented in this chapter has resulted in two publications[1,2].

[1] D. Johnson and G. Tzanetakis. VRMin: Using Mixed Reality to Augment the Theremin for Musical Tutoring. In *Proceedings of the 2017 Conference on New Interfaces for Musical Expression*, 2017.

[2] D. Johnson, D. Damian, and G. Tzanetakis. Evaluating the Effectiveness of Mixed Reality Music Instrument Learning with the Theremin. *Virtual Reality*, Provisionally Accepted 2019.

5.1 Introduction

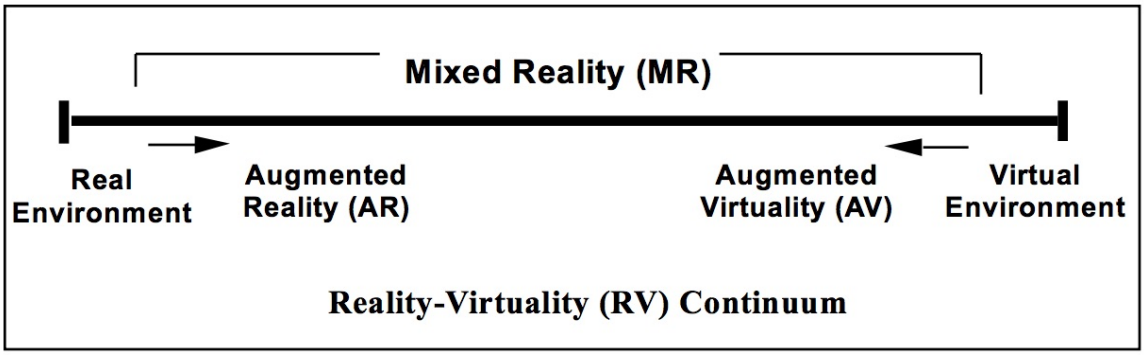


Figure 5.1: Milgram's Reality-Virtuality Continuum

The goal of CAMIT research is to study computational tools and techniques to improve the musical learning process using emerging technologies (Percival et al., 2007). The emergence of large CAMIT projects, such as the TELMI project (Ramìrez et al., 2019; Dalmazzo and Ramirez, 2019), as well as the success of computer based music education platforms, such as Yousician (2019), show that there is demand for computational systems to enhance the music instrument learning experience. Such systems facilitate improved methods for self-teaching and enhance teacher led training by augmenting daily practice with musical performance feedback. As discussed in Section 4.6.1, performance feedback can be presented to students with a variety of methods, including post performance feedback and real-time visual or auditory feedback, each with their own benefits and drawbacks. For this chapter, I focus on methods for presenting students with real-time visual feedback (RTVF) as they are practicing.

Challenges of employing RTVF with a standard computer display include the need for a student to constantly shift focus between the display and the music instrument as well as the need to cognitively map visual cues from the 2D interface to their corresponding location on the musical instrument. These challenges increase demand on the student's cognitive processing abilities. The increasing availability of consumer level XR devices affords new immersive training techniques that may help overcome these challenges and change the way music is learned by mapping RTVF directly onto musical instrument. Augmenting the instrument with virtual

objects for RTVF enables students to keep on the instrument while also viewing the feedback. To the best of my knowledge, there is no research on validating the effectiveness of RTVF for either standard displays or XR.

While many industries have found XR training to be effective, music education has yet to see a breakthrough in the use of these technologies. Music pedagogy presents unique challenges that require further research to gain an understanding of how immersive environments should best be employed to improve music learning. While performance and training errors are not as critical as in areas such as surgery, playing music requires skilled interactions through high levels of hand precision and coordination. Additionally, musicians need to feel the response of their actions through the physicality and haptics naturally afforded by real instruments. One of the biggest differences in music instrument learning is that proficiency depends not only on precise motor skills but also a well trained ear. Thus, music instrument learning tools should be designed so that that students are able to focus on the auditory aspect of their performance while also learning the proper motor skills. Because of the high precision and haptic feedback required for music instrument learning, I believe that extended reality enhanced musical instrument learning (XREMIL) environments should be designed towards the AR end of the RV Continuum in Figure 5.1 (Milgram et al., 1995). Such design affords environments where the XR virtual layers interact with real world objects. In other words, students should be able to practice with their physical instruments while receiving visual cues through XR technologies.

In this chapter, I build on this idea and study the effectiveness of such an approach for XREMIL by developing an immersive learning environment for the theremin, called *MR:emin*. *MR:emin* integrates a physical theremin with an immersive learning environment. In line with (Milgram et al., 1995), I consider this system to be MR because of the integration of a physical object, i.e. the theremin, with the virtual world; thus, contrasting with VR environments in which students would interact with virtual instruments. Using *MR:emin*, I perform a user study to enable an increased understanding of the effects of the additional visual layer of XR on the music learning process.

5.1.1 The Theremin

Developed in the 1920s by Russian physicist Lev Theremin, the theremin is one of the earliest known electronic music instruments and one of the few instruments that is performed without physical contact. It is controlled through two antennas that are able to sense the proximity of electromagnetic fields, such as a person's hands. The distance from one antenna controls the frequency of oscillators that modify the pitch, and the distance from the other antenna controls the amplitude of the output. The design creates a continuous pitch space, making the theremin one of the most expressive electronic music instruments for performers. This also makes it one of the most difficult instruments to learn and to play.

Using a theremin as the basis for the user study provides some benefits. First, very few people have played the theremin, making it easy to find and recruit participants with similar skill levels. The mechanisms for pitch and volume control are easy to learn but require high proficiency to play well. Furthermore, because the theremin has no physical signifiers or visual cues, students learning to play the theremin must focus on the auditory aspect of their performance rather than simply memorizing note locations. This helps ensure improvement in performance is a factor of both improved motor skills, as well as, improved listening skills. These factors contributed to the decision to build *MR:emin* for our study of XREMIL.

5.1.2 Contributions and Outline

While there has been some research on the design and development of XREMIL environments, to the my knowledge, there are no known studies on the effectiveness of such environments. With this work, I contribute to the fields of CAMIT and XR training by addressing RQ_2 and RQ_3. The major contributions are

C_1 an XREMIL environment controlled with a physical music instrument affording high fidelity musical interactions,

C_2 the results of a user study on the effects of learning in such an environment contributing to a the general body knowledge for CAMIT and XR training ,

C_3 and some general guidelines for designing XREMIL.

The rest of this chapter is structured as follows. In Section 5.2 I discuss the design process to develop a XREMIL environment to utilize for the proposed user study. This section includes a discussion, in Section 5.2.1 on design guidelines used inform the design process. Section 5.3 outlines the details and design of the user study using *MR:emin*. I then provide an analysis of the quantitative and qualitative results of the user study in section 5.4, followed by a discussion of the results in section 5.5. I then conclude the article and discuss potential future work.

5.2 System Design

This section discusses the design and implementation of a novel XREMIL environment for the theremin that will be used for the proposed user study. With limited previous work to build on, I explored the XREMIL design space by developing the theremin learning environment iteratively using the interaction design process (Preece et al., 2002). Before developing the system, I identified a set of design guidelines for XREMIL environments based on previous research and experience. Then, I designed a prototype system, called *VRMin* to gain an understanding of the technical and design needs for immersive music learning. Next, I informally evaluated the system using an Heuristic Evaluation (HE) to identify major usablity issues. Results of the informal evaluation informed the design of the final version of the theremin XREMIL environment for use in my proposed user study.

Design Considerations

There are a few design requirements to take into account during the development of the VE. First, the system must integrate a physical theremin into the VE providing users with an MR experience in which to perform and generate sound with the actual instrument they are learning. Second, because a thereminist's hands should be unobstructed to allow for control of musical output, all interaction and hand tracking should be performed using the theremin as opposed to requiring standard VR input controllers.

A Moog Theremini Moog (2019) is employed to meet the design needs regarding system interaction. The Theremini outputs Musical Instrument Digital Interface (MIDI) messages for both pitch and volume antennas affording control of the VE through interaction with the instrument. For each antenna, the Theremini outputs MIDI control messages with values $[0, 127]$ indicating proximity to the respective antenna. These values will be mapped to actions controlling virtual elements in the VE. For example, the motion of virtual hands will be controlled by mapping the MIDI values to position data for the hands.

Lastly, to support the task of learning notes on the theremin, the learning environment must provide some visualization of the pitch space to support the learning process. One of the major challenges of learning the theremin is a lack of affordances and constraints that provide guidance within the pitch and volume spaces. Furthermore, the theremin has continuous control rather than being composed of discrete notes. This makes it difficult for new thereminists to find the correct location of a given note. Augmenting the pitch space with visual signifiers will support learning by providing users with a clear destination to play a given note.

To facilitate VE design and development, the Unity game engine (Unity, 2019) is employed. Unity provides the necessary tools to implement an immersive 3D learning environment for a theremin that augments the theremin with virtual objects. The virtual objects will be used to visualize the pitch space and provide additional interface elements required by the system. Unity also supports a variety of XR HMDs needed for an immersive experience.

5.2.1 Design Guidelines for XREMIL

With little previous XREMIL research to build on, a set of guidelines will help inform design decisions when developing new environments. Since there are no known existing guidelines for the design of XREMIL, I look to general VE design guidelines. Specifically, Gabbard et al. (1999) proposed a framework of design guidelines developed specifically for VEs. VEME have different characteristics of from those of traditional VEs. Serafin et al. (2016) presented nine design principals specifically for VRMIs, a number

of which are also relevant to XREMIL. From a pedagogical standpoint, an optimal use of performance feedback is an important factor in the design of VEs. Research has been conducted in simulation-based medical education that shows how learning is affected by different types of performance feedback Hatala et al. (2014). The ideas from each of these fields and general HCI design (Nielsen and Molich, 1990; Nielsen, 1994) have informed our development of design guidelines for XREMIL which are also used as heuristics for HE, discussed in Section 5.2.3.

Before providing the guidelines, I must make the distinction between two types of feedback for learning systems, *performance feedback* and *interface feedback*. *Performance Feedback* relates to the feedback given to a user to based on an assessment of their musical performance. This type of feedback is analogous to the feedback a music teacher would give while watching a student perform. On the other hand, *interface feedback* is feedback presented to a user based on the results of a given interaction with interface elements. An example of interface feedback is a visual signifier indicating that an action, such as loading a new score, was successful.

Based on the research discussed above and my experiences in XR design, I established the following categories of preliminary design guidelines for XREMIL. The guidelines selected have a focus on design factors that lead to positive user experiences for XREMIL. Presented here the high level guidelines that I found to be the most important for the task of learning musical concepts.

1. **Guidelines for performance feedback** - these guidelines help design performance feedback that optimizes learning and minimizes the over reliance on feedback (Hatala et al., 2014).

    (a) RTVF shouldn't prevent a user from focusing on aural feedback.

    (b) Reduce cognitive load by limiting the amount of concurrent feedback (Hatala et al., 2014).

    (c) Provide terminal performance feedback upon completion of the practice task (Hatala et al., 2014).

2. **Guidelines for VE design** - these guidelines are used to guide design decisions about the VE for the practice space promoting an enriching

and comfortable experience. A properly designed environment also increases the usablity of the system.

(a) Visibility of system status (Nielsen, 1994).

(b) Choose metaphor(s) that naturally match the application task space (Gabbard, 1997).

(c) Match between system and the real world (Nielsen, 1994) .

(d) Create a Sense of Presence (Serafin et al., 2016).

(e) Consider Display Ergonomics (Serafin et al., 2016).

(f) Consider Controller Ergonomics.

(g) Represent the Player's Body (Serafin et al., 2016).

(h) Ensure that users' avatars provide a familiar, accurate, and relevant frame of reference (Gabbard, 1997).

(i) Allow users to alter point of view, or viewpoint (Gabbard, 1997).

3. **Guidelines for interaction in XREMIL** - while some interaction with the interface may be required during practice it should be limited to reduce students' cognitive load affording more mental capacity for learning new skills.

(a) Limit non-essential interaction during practice.

(b) Recognition rather than recall (Nielsen, 1994).

(c) Make use of existing skill (Serafin et al., 2016).

5.2.2 Prototype Design

In the first iteration of the design process, I developed *VRmin*, an MR system that integrates the Theremini (Moog, 2019) for sound generation and interaction with mobile VR technology for visual augmentation. Implementing the system using mobile VR increases accessibility as devices become more readily available. In this iteration, I use the Google Daydream platform (Google, 2019) and a Google Pixel XL mobile device. Using this setup, the environment provides visual cues to guide the performer within the

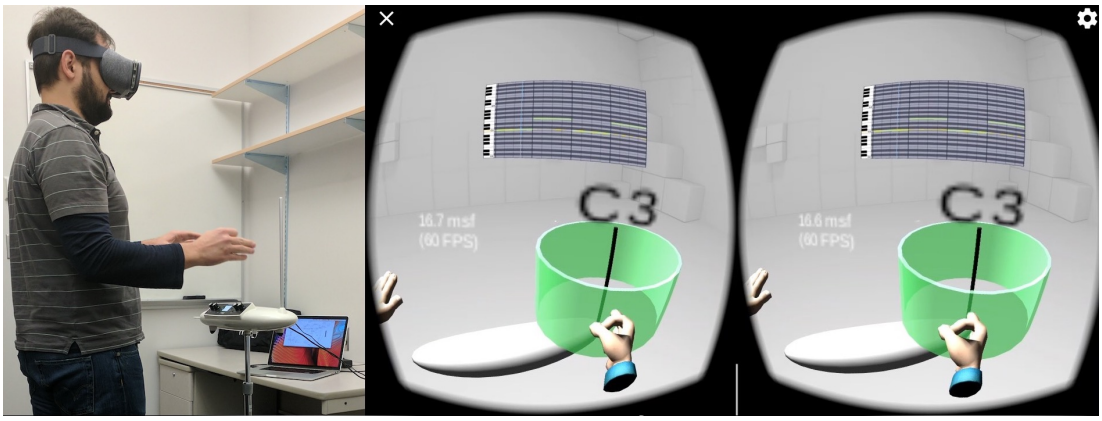


Figure 5.2: The author practicing the theremin using VRMin and a screen capture of the learning environment.

pitch space by augmenting the theremin with visual signifiers. Figure 5.2 shows the author practicing the theremin with the *VRmin* XREMIL.

Because the Daydream is not directly connected to a computer, data from the theremin must be transferred over a wireless network connection. To do this, the Theremini is connected to a computer running a Pure Data patch which acts as middleware for converting MIDI messages to Open Sound Control (OSC) for lightweight network communication with the mobile VR device. Additionally the middleware handles simple gesture detection to add functionality to the theremin. OSC messages containing the pitch and volume control data, as well as detected gestures, are sent from the middleware to the *VRmin* system which is configured as an OSC server. The OSC messages received by the server are used to control interface elements described in the next section.

User Interface

The *VRmin* environment, shown on the left in Figure 5.2, contains visual representations of the physical elements that comprise the performance space, in this case the Theremini and the user's hands. These are augmented with visual elements not present in the real world, a visual note signifier and a virtual piano roll. The visual elements are controlled through input from the middleware allowing the Theremini to act as the controller. The pitch and volume values control the locations of the hands relative to

the antenna and gesture messages are used to control interactions with the tutoring interface.

The VE is designed with virtual objects that represent their physical counterparts as closely as possible in respect to both their physical dimensions and positions. The theremin is represented with a simple 3D model with dimensions that replicate the physical theremin within the performance space. 3D hand models are used to represent the location of the user's hands relative to each antenna. Each hand is modeled in a posture similar to one a thereminist would typically use; the pose of right hand is similar to the "OK" sign and left hand uses a horizontally oriented flat hand. Using the "OK" sign also mitigates an issue the authors found in which a vertically oriented flat hand occludes the tip of the middle finger from the field of view. This posed a problem because of the precision required to correctly play a note with a theremin.

The location of notes within the the theremin pitch space can be represented with a circle since all points are equidistant from the antenna. For a 3D representation of notes I use hollow cylinders to indicate the location of a note relative to the antenna within the pitch space. Each note is also accompanied by text indicating the name of the note for additional feedback. Notes are transparent so that the user has constant feedback as to the location of the tip of the virtual hand even as their hand passes through the cylinder. To play a displayed note, a user moves their hand such that the position of the tip of the virtual hand is located at the center of the two concentric circles that comprise the hollow cylinder. To further enhance performance feedback, the cylinder's color changes to green when the tip of the middle finger is in the correct location[3].

The interface also contains a piano roll to display a score during the performance session. The piano is positioned just above the pitch antenna and is rotated along the Y axis by 30 degrees. This positioning provides the best view of the piano roll that is not obstructed by any portion of the theremin or note indicators. The user is able to easily view the piano roll by slightly tilting their head up. This placement also keeps a view of the right hand and virtual note in the user's periphery as they view the score. The piano roll includes a pitch indicator line to provide additional performance

[3]see a video demo at `http://web.uvic.ca/~davidjo/demos/VRMin-NIME2017-Demo.mp4`

feedback as the score is being viewed.

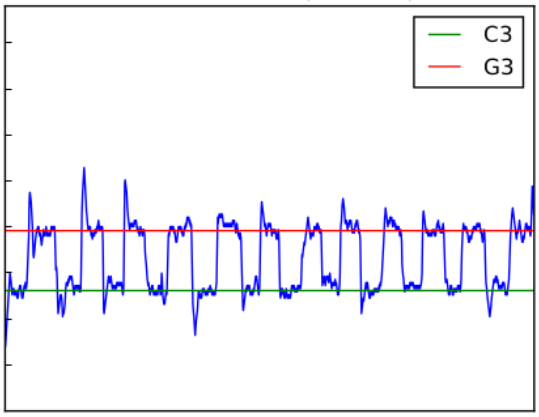
Since *VRmin* has the design constraint of avoiding a separate controller, I implemented simple gestures for interacting with the system using each antenna as a trigger. To activate a trigger, a user simply places a hand in the near position of the antenna for a period of time that is longer than the hand would typically be in that position during performance. For example, the volume antennae is used as a trigger to start and stop each practice session by holding a hand in the near position of the antennae for two seconds. Additionally, the user is able to change the score used for the practice session similarly using the pitch antenna trigger. A third trigger is implemented (but not currently not mapped to an action) by placing hands at the near position of both antenna at the same time.

Performance Analysis

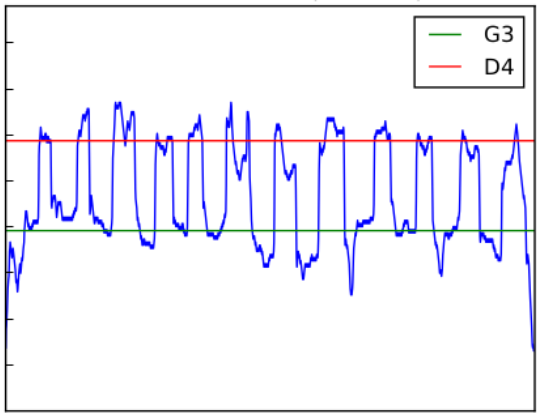
To improve the analysis of practice sessions, I implemented performance logging in *VRmin*. The logging system records the OSC messages and values sent to *VRmin*. This data could then be used to replay a practice session for a visual analysis of the performance (similar to reviewing video recordings). More importantly, teachers can gain insight into a student's progress through a quantitative analysis of the data by using plots similar to figure 5.3.

The plots in figure 5.3 show the results of two pitch matching practice sessions, one using *VRmin* and one without. In each session, the student attempts to match computer generated tones and hold the tones for five seconds each. In this case the student is practicing moving between intervals of a perfect fifth. After each session, the performer attempted to perform the same tones without *VRmin* or the corresponding pitch playing in the background.

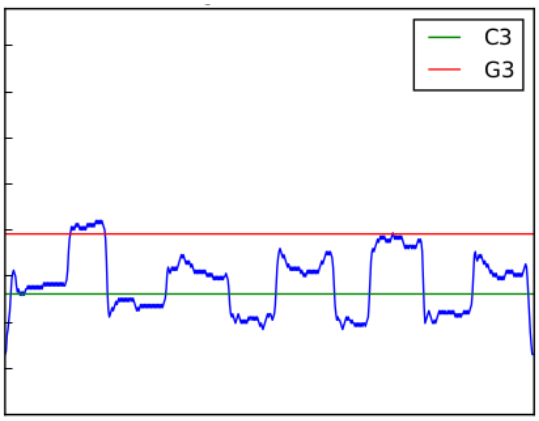
In figure 5.3 the red and green lines represent the pitches the thereminist was instructed to play as part of a pitch matching practice session. The first two charts show the results of the student during the pitch matching training sessions and the last two show data from the student attempting to play the notes on their own. While these are by no means conclusive, I do see that when using *VRmin* the user is much more precise during the



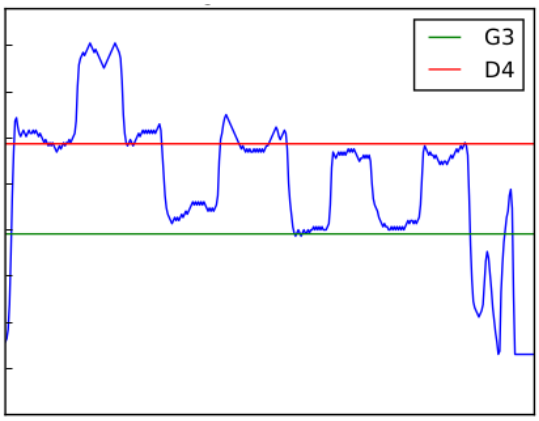
(a) Training session with VRmin



(b) Training session without VRmin



(c) Performance after training with VRmin



(d) Performance after training without VRmin

Figure 5.3: Performance analysis plots of practice sessions with and without VRMin

practice session as there is much less vibrato and it was easier to find the correct location. Post pitch matching, however, it appears the user trained without *VRmin* was better able to recall the locations of the notes after a bit of adjustment.

5.2.3 Heuristic Evaluation of Prototype

Self evaluation of an interface is a critical step in the interaction design process to quickly evaluate the design before the intrusive and expensive process of a user study. One commonly used method is Heuristic Evaluation (HE), a method in which an evaluator determines the good and bad aspects of an interface based on a set of guidelines, known as heuristics (Nielsen and Molich, 1990). Many of these heuristics common for traditional user interfaces (see (Nielsen, 1994; Weinschenk and Barker, 2000)) are not well suited to the design challenges presented for virtual interfaces and environments. As Gabbard et al. (1999) claim, "[these guidelines are] too general, ambiguous, and high level for effective and practical heuristic evaluation of VEs". The newly identified design guidelines provide a set of XR specific heuristics for HE.

Performing an HE requires a user task to guide the evaluation of the interface. For this evaluation, I employ an ear training task for beginning thereminists. During the practice session *VRmin* plays an aural tone of a specific pitch according to a practice score. The user's goal is to match the the pitch by moving their hand to the correct location relative to the pitch antenna. When immersed in the *VRmin* environment, the user is presented with visual signifiers to help guide them to the correct location. My findings from the implementation of an HE while performing the practice task are listed in Table 5.1.

Evaluation Discussion

By performing HE on *VRmin* using the outlined design guidelines as heuristics, I identified several potential design limitations. Some of the limitations can be addressed in a later iteration of the interface while others may requires a more formal usablity study to gain a better understand of their effects. Below I discuss the design limitations from Table 5.1 that I feel

ID	Comments
1a	*By trying to match the position of the hand to a visual element a user's attention is directed to the visual feedback rather than the auditory feedback from the theremin.*
1b	*Performance feedback includes both the visual pitch cues and the piano roll interface.*
1c	*Performance analysis graphs are available but are not integrated into the VRMin interface.*
2a	Visual cues with the note name are provided and a piano roll representation of the score provides status on the current and upcoming notes.
2b	All current interaction is performed using the only the theremin.
2c	The VE provides virtual representations that match the real world space.
2d	Presence is promoted by matching the real world to the VE and by integrating real objects with virtual objects.
2e	The Google Daydream is a comfortable display but *has a small field of view making it difficult to view visual objects in the periphery.*
2f	The environment is controlled through interactions with the theremin removing the need for an additional input controller.
2g	Virtual hands are displayed in the environment but *there is no avatar representing the students body.*
2h	The hands are in postures used by a number of renowned theremins but *not all thereminists use the same hand posture.*
2i	*The Daydream platform provides offers 3 Degrees of Freedom for headset movement tracking limiting the ability to alter viewpoints.*
3a	After starting a practice session there are no additional interactions required that are not directly tied to practice.
3b	*There are no visual cues signifying the control gestures available, require students to recall them.*
3c	*VRmin* uses natural interaction affording use of existing skills.

Table 5.1: Findings for each design guideline from the implementation of a Heuristic Evaluation (italics indicate a potential design issue).

are most important to either address through interface changes or gain an understanding of with a controlled user study.

Design guideline 1c suggests providing terminal feedback upon completion of the practice task. Further, Blanco and Ramirez (2019) show the effectiveness of terminal feedback for improvement in violin playing. *VRmin* does not provide terminal feedback to the user after the practice session. To address this limitation I suggest implementing an additional view in the XREMIL environment displaying data from the performance logging after a practice session.

Design guideline 2e suggests to consider the display ergonomics. Taking ergonomics into consideration, the Daydream platform is light and comfortable to wear but has a small field of view limiting peripheral vision. Playing an instrument requires peripheral vision to allow a wider view of the performance space; therefore, future iterations of the system should employ an HMD with a larger field of vision.

Design guideline 2i suggests that users should have the ability to alter their viewpoint within the VE. The Daydream HMD only implements three degrees of freedom (3-DoF) tracking for head movement. Therefore, the *VRmin* environment allows for only limited alterations in viewpoint. Implementing the environment using an HMD with six degrees of freedom (6-DoF) would allow a student to move around more freely in the environment for altering their viewpoint. With limited DoF users are constrained to a specific location and viewpoint in the VE which may not be comfortable or ideal for that particular user.

Finally, the most significant design challenge is designing the proper balance of performance feedback to minimize the chances that students become over reliant on a visual layer. This may be achieved by adhering to design guideline 1a. While performing the pitch matching exercise in *VRMin*, I felt that most cognitive abilities were focused on adapting hand position based on the visual cues rather than focusing on the auditory feedback of the theremin. The added cognitive demand of processing and matching the visual cues may lead to an over reliance on the learning environment to perform accurately. The visual feedback does, however, provide valuable information for students with limited ear training. By placing cues where hands should be located for given pitches, the student can be con-

fident that they are playing the correct pitch, increasing the efficiency of their practice. It has yet to be seen if the increased confidence in performance outweighs the increased cognitive demand from XR based RTVF and how this affects students' learning. As a user gains experience with the interface, the increased practice efficiency may compensate for the increased cognitive load in the beginning. Finding the proper balance will take a comprehensive user study that evaluates students' learning performance with and without *VRmin*.

5.2.4 *MR:emin* Description

MR:emin is the revised version of the XREMIL environment to be used in the user study for evaluating the effects of XR RTVF on musical instrument learning. The new environment builds on *VRmin*, addressing the design limitations learned during the HE. While still integrating a Moog Theremini (Moog, 2019), *MR:emin* is now implemented for the Windows Mixed Reality (WMR) platform with an immersive Samsung Odyssey HMD (Microsoft, 2019). I avoid using the included input controllers and instead offer control through interaction with the theremin by utilizing the MIDI data from the antennas. For theremin performance, the standard input controllers are too intrusive since a thereminist's hands should be unobstructed for proper control of the musical output. The *MR:emin* environment was developed using Unity with the Windows Mixed Reality Toolkit (MRTK) (Microsoft, 2019b).

The WMR platform provides a full 6DoF with inside-out tracking offering a more engaging room scale XR experience. Upgrading to a room scale experience provides users with an increased ability to alter their viewpoints in the environment, addressing design issues related to design guideline 2i. By allowing for altered viewpoints, users can move freely through environment to find a comfortable standing position that best suites them. Further, the upgraded HMD has a larger field of view for improve ergonomics and peripheral vision addressing issues related to design guideline 2e.

MIDI Control

The Moog Theremini features MIDI connectivity for data transfer between the instrument and a computer. MIDI is a common technical standard and communication protocol commonly used with digital musical devices. Using the MIDI communication protocol, *MR:emin* receives control data from the Theremini allowing the theremin to act as a controller, thus, affording interaction without standard hand controllers. Both the pitch and volume antennas generate MIDI control data, with values $[0, 127]$. The values represent the proximity of a sensed object to the respective antenna. A MIDI input device component is implemented within *MR:emin* to receive and process this data. The data received from the controller is mapped to the position of virtual hands within the VE. Because the output of each antenna contains only one range of values, the virtual hands move with one degree of freedom and cannot replicate the exact location or pose of the performer's hands. Using the theremin as the controller, however, allows a student to perform and generate musical output from the physical theremin while being fully immersed in the VE.

In addition to receiving MIDI input, *MR:emin* also implements a MIDI file reader for importing MIDI scores. A MIDI score is a digital representation of a music score that is composed of sequences of integers, i.e. MIDI notes. The music score imported from the MIDI file is used to control the content for student training exercises.

Virtual Environment

Similar to the *VRmin* iteration of the VE, the *MR:min* VE, shown in Figure 5.4, contains 3D models of the physical elements that comprise the performance space (seen in use by a user study participant in Figure 5.5c), namely the theremin and the student's hands. The performance space is then augmented by overlaying virtual objects, not present in the real world, on the 3D models. This includes a visual signifier to indicate the location of a specified note and a virtual element to display score information, such as the current note, the next note and the current note's duration. The new score information display replaces the piano roll interface implemented in *VRmin*, for a simplified environment reducing visual noise to conform to

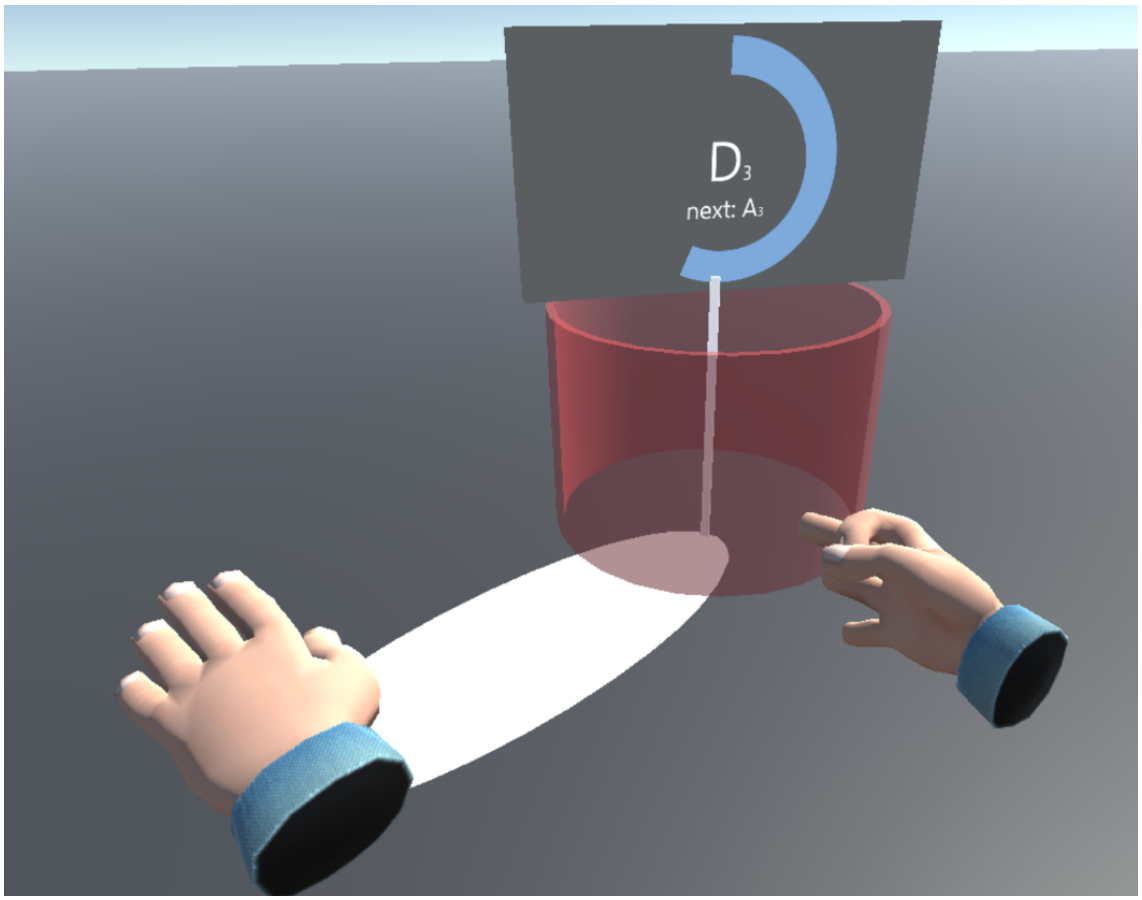


Figure 5.4: The *MR:emin* XREMIL Environment

design guidelines 1a and 1b.

In the VE, virtual objects are mapped as closely as possible to their real world counterparts with respect to both dimensions and positions. The theremin is represented using a 3D model with dimensions that replicate that of the physical theremin and is positioned within the VE based on physical theremin's real world location. Virtual hand models are used to represent the location of the user's hands relative to each antenna. As previously mentioned, hands are only able to move in the VE with one degree of freedom so are only able to communicate a relative position to the users. Each hand is modeled in a static posture similar to one a thereminist might typically use during performance. This design affords high fidelity between the VE and the real world interactions.

The *MR:emin* learning environment teaches students a sequence of notes on the theremin by augmenting the theremin with visual cues, providing

the students with real-time visual guidance and feedback for note locations. The location of a note in the theremin pitch space is represented as the edge of a virtual tube that is displayed around the pitch antenna. The virtual note is transparent so that the user has constant visual feedback on the location of the virtual hand even as their hand passes through the edge of the tube. To play the note currently displayed, a user moves their hand such that the position of the tip of the virtual hand is located just at the edge of the virtual tube. To provide additional feedback on the correctness of the action, the color of the virtual note changes to blue when the hand is in the exact note location. The sequence of notes displayed during training is controlled via an imported MIDI score. At each note transition in the score, the radius of the virtual note changes size to indicate the location of the new note. As the note size changes an animation takes place to make the visual transition smooth. The visual cues are used to train students to play sequences of notes on the theremin by guiding the student to the correct location of each note in a score and then providing visual feedback when they arrive at the exact location.

To help with ear training, *MR:emin* implements a tone matching feature allowing students to hear the correct tone alongside their performance as they follow the score. To implement this feature, the system includes a tone generator that is used to create synthesized tones at specified frequencies. The tone matching exercise is used as the basis for the user study as described in Section 5.3.

5.3 User Study

Music learning poses different challenges compared to other fields that have evidence in support of positive learning transfer through XR training. In addition to learning precise motor skills, students of music must also learn to listen to the sound they produce to self evaluate their performance. Furthermore, some instruments that lack signifiers of discrete notes, such as the violin, require precise ear training. A student learning piano, for example, can easily memorize where a $C_3^{\#}$ is located, even if they don't learn what it sounds like. Students of fretless instruments, on

the other hand, must learn to find a specific note by ear. Adding a visual layer to the music learning process makes this already difficult task more complex as students will need to process both visual feedback and auditory feedback at the same time. The purpose of this study is to gain an improved understanding of how the addition of visual cues with XREMIL affects the music learning process.

5.3.1 Research Questions

My experiences developing CAMIT have raised questions about how the added complexity of RTVF with XR affects the musical instrument learning process raising RQ_2. In terms of learning transfer, I hope to gain a better understanding of how skills learned while training on an instrument in MR with visual augmentation transfers to playing the instrument in the real world with no visual cues. Specifically with this study, I hope to contribute to a better understanding of auditory learning transfer from XREMIL to real world performance. It is possible that a student may learn to play a music instrument through the memorization of motor skills alone, so it is important to know if students are still able to focus their cognition on the auditory feedback of the instrument while processing the added visual cues. Additionally, I aim to learn if XREMIL provides a benefit over simply displaying the same feedback on standard 2D display. This leads to the following research questions.

RQ_{2_1} Do participants practice with better precision with RTVF in a XREMIL environment?

RQ_{2_2} Does RTVF in a XREMIL environment improve theremin learning more than learning using only audio?

RQ_{2_3} Does RTVF in a XREMIL environment improve theremin learning more than learning with RTVF with a 2D HD display?

In addition to objective evaluation on the learning transfer of *MR:emin*, the study is designed to obtain subjective data from the participants about their experience with the environment. The subjective assessment is meant

to gather insight on the design of XREMIL addressing RQ_3, leading to the following research questions.

RQ_{3_1} How does training in an environment with RTVF affect participants' ability to focus on auditory feedback?

RQ_{3_2} Which learning environment is more engaging to participants?

Hypotheses

The hypotheses below are evaluated through the user study. These were established based on informal evaluations of *MR:emin* prototypes, as well as, my own experience with the learning environment during the development process. The first prototype of *MR:emin* was presented as a research exhibition for a local technical exposition and during a demo session at the international conference on NIME (Johnson and Tzanetakis, 2017). While both sessions were informal, they provided invaluable insight into the design of the interface and the user study. Furthermore, I gained additional insight through a small pilot study performed with the current *MR:emin* environment. These insights contributed to the development of the following hypotheses.

H_1 Training in a virtual learning environment with visual cues will help students find the correct note faster and hold the correct position more precisely. Furthermore, students trained in the fully immersive *MR:emin* environment will perform with greater precision during training than those using the non-immersive environment.

H_2 The learning transfer of a single training session, will be greater in a learning environment in which there are no visual cues.

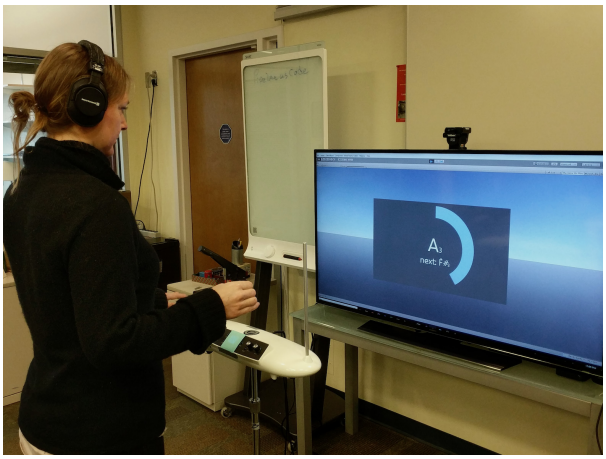
H_3 Of the two *MR:emin* environments, students trained in the immersive one will see a greater improvement in performance than those trained with the non-immersive environment.

H_4 Students will be more engaged and have a better user experience with the immersive learning environment compared to both the non-visual and non-immersive *MR:emin* environments.

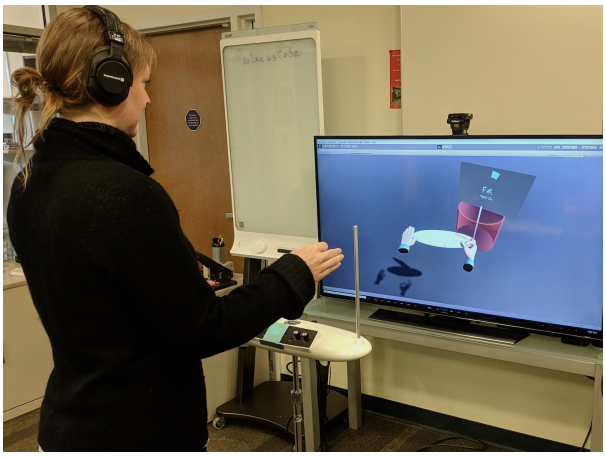
5.3.2 Study Design

To evaluate and compare the effectiveness of different learning environments, the user study employs three training interfaces: the standard immersive *MR:emin* environment, a version of the environment displayed on a 2D HD display, and an environment with no visual feedback. A between-groups user study, with ethics approval provided by the University of Victoria Human Research Ethics Board, was conducted to evaluate and compare the learning transfer of each environment. Participants were recruited from the general population of graduate students and faculty at the University of Victoria ($N = 30$) and were evenly and randomly distributed in either the control group or one of two treatment groups.

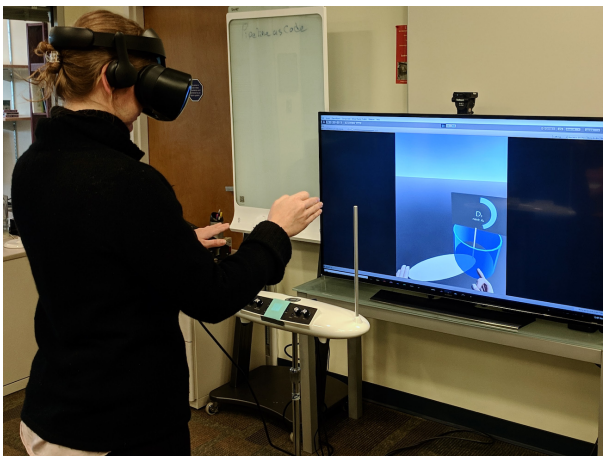
The effectiveness of each environment is evaluated by treating the training environment as the independent variable and participants' performance and experience as the dependent variables. The control group participants receive theremin training with no visual feedback (the **NoVis** group). Participants in this group must rely on auditory feedback alone to self-evaluate their performance during training. Participants of the treatment groups are trained with the *MR:emin* VE which includes real-time visual cues to provide guidance and feedback to assist participants in evaluating their performance during training. Participants in one treatment group receive training from the fully immersive *MR:emin* VE as described in section 5.2.4, with an HMD (the **Imm** group); participants in the other treatment group receive training from the non-immersive *MR:min* environment displayed on an HD display placed directly in front of them. (the **NoImm** group). Figure 5.5 shows a participant performing in each of the training environments. In figure 5.5a the participant is performing with the NoVis environment and has no visual guidance to assist with locating specific notes. The only visuals displayed are for conveying information about the practice score. In Figures 5.5b and 5.5c the participant performs with the NoImm and Imm environments, respectively. In Figure 5.5b the participant uses the HD display to view *MR:emin*; whereas, in Figure 5.5c the participant is fully immersed in *MR:emin* (the HD display is only used for observation by the research team). With this study design, I compare the effectiveness of the different training environments.



(a) NoVis Environment



(b) NoImm Environment



(c) Imm Environment

Figure 5.5: A participant performing in each of the three different training environments.

5.3.3 Performance Task and Evaluation Metrics

To assess the learning transfer of theremin skills, I employ a tone matching exercise for performance evaluation. During the exercise a participant hears a set of tones generated sequentially at specific pitches and is expected to play those same tones on the theremin. For this study, participants are tested on a sequence of three notes (D_3, $F_3^{\#}$, A_3) with each note lasting for a duration of eight seconds, giving the user time to find the note and keep their hand steady in the correct location for the duration. To ensure consistent evaluation, participants from each sample receive the same testing environment which has no visual guidance for note locations (this is the same as the NoVis environment as described in section 5.3.2).

For quantitative analysis of the learning environments, performance metrics are calculated that describe how well a participant matched the expected note sequence. Using performance logging capabilities of *MR:emin*, the participants' performance errors are measured by calculating the distance of the MIDI input signal from a signal representing the expected pitches (Figure 5.6 shows examples of these signals during training sessions). Two distance metrics, the euclidean distance and the cosine distance, are averaged to calculate the total performance error, D. The lower score for D the better the participant has performed. The euclidean distance measures how close the participant was to the correct notes while the cosine distance measures how well the participant is able to match the relative position for each note interval. To account for varying musical abilities among the participants, the final measure used for statistical analysis is the percentage of improvement, PI, from the pre-training assessment to the post-training assessment calculated by $PI = (PreScore - PostScore)/PreScore$.

5.3.4 Procedure

With the exception of the training environment, all participants followed the same procedure to evaluate their learning transfer for the tone matching task described in section 5.3.3. First participants were given a familiarization session to get used to the tone matching exercise and the corresponding training environment. Once familiar with the setup, a baseline of

the participant's abilities was assessed with a pre-test in which the participants performed the tone matching exercise, without any visual cues, for two repetitions of the three note sequence. Next, participants underwent a training session in their assigned training environment. The training consisted of three short sessions in which the participant performed the tone matching exercise for six repetitions each session. For participants in the treatment groups, training included the visual feedback of *MR:emin* (with or without immersion, depending on the sample they were assigned) to assist with finding the correct note location. After training, the participants were given a short break and asked to fill questionnaires about their experience: NASA Task Load Index (NASA TLX) questionnaire (Hart, 2006) and the User Experience Questionnaire (UEQ) (Schrepp et al., 2014). Participants trained with the immersive *MR:emin* VE were also asked to complete the Presence questionnaire (Witmer and Singer, 1998). After completing the questionnaires, participants were given the post-test, which was exactly the same as the pre-test, to calculate a measure of their improvement, section 5.3.3 discusses the performance metrics in more detail.

For subjective comparison of training environments, participants were given a short training session in a different training environment after completing the main training tasks. Participants originally trained in the NoVis or NoImm environments were given a chance to experience training with the Imm environment. Those trained with Imm performed a short training session with the NoImm environment. To finish the session, I interviewed the participants about their experiences with the different training environments.

5.4 Results

5.4.1 Quantitative Analysis of Objective Data

The metrics, D and PI, discussed in section 5.3.3, provide objective data to evaluate the effectiveness of each treatment sample compared to the control group. I use PI as the statistical value for analysis of learning transfer between samples. Statistical analysis is also performed on participants' performance during training, using D as the statistical value, to analyze

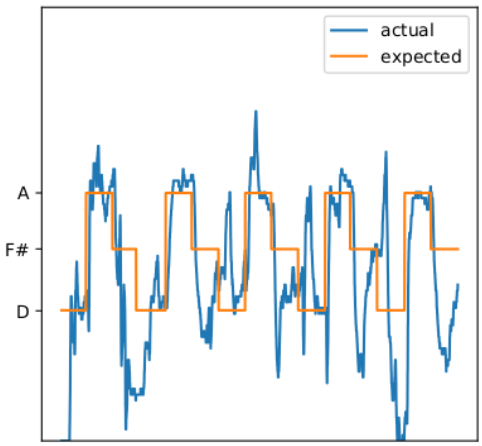
training environment efficiency. Non-parametric analysis is employed because I was unable to show that the data meet the assumptions for parametric analysis; namely that the data are normally distributed and each sample has equal variances according to both the Shapiro-Wilk's test and the Levene's test for equal variances (Siegel and Castellan Jr., 1988).

Training Performance

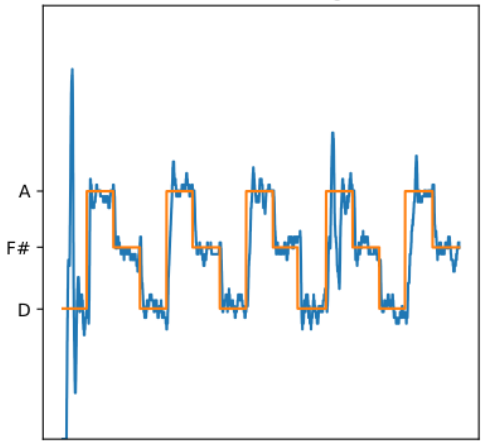
The graphs in Figure 5.6 show the training performance of three participants, one from each training sample, and are typical examples of the expected and actual performance data used to calculate D. The orange line indicates the tone the participant was expected to perform, and the blue line is their actual performance. A review of the graphs for all 30 participants leads me to believe that participants were much more precise while training with visual methods; while I only show graphs from one participant per sample the results are typical across all participants. The boxplots in Figure 5.7 summarize the training performance, in terms of D, for the control and each of the treatments. It is clear from the boxplots that visual environments improve precision during training. A Kruskal-Wallis test shows a statistically significant difference in training precision between the different training samples, $H = 20.23$ and $p < 0.001$. Single tail post-hoc pairwise comparison using Dunn's test with Bonferroni correction (Dunn, 1964) reveals that Imm has the smallest training D (mean rank = 6.5) with significant differences between NoVis (mean rank = 24.2), $p < 0.001$, and NoImm (mean rank = 15.8), $p = 0.0273$. NoImm D is also shown to be significantly smaller than NoVis, $p = 0.0493$.

Learning Transfer

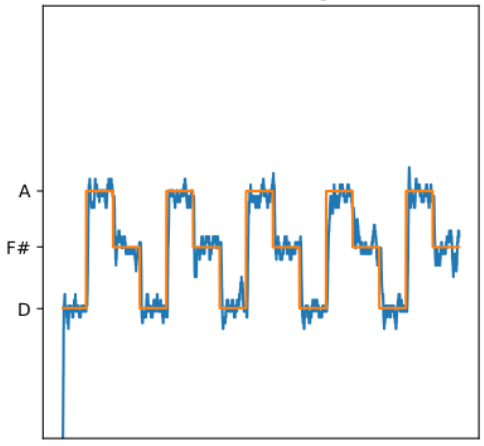
To analyze learning transfer across the training treatments the percentage improvement of D from pre-test to post-test, PI is employed. The percentage improvement is summarized with boxplots in Figure 5.8. A Kruskal-Wallis test indicates there is no significant difference in PI between the samples, $H = 3.003$ and $p = 0.2227$. A single tail post-hoc pairwise comparison using Dunn's test with Bonferroni correction on the other hand reveals that Imm (mean rank = 12.7) training resulted in a significantly smaller percent



(a) NoVis Training Session



(b) NoImm Training Session



(c) Imm Training Session

Figure 5.6: Performance data from the training sessions of participants from each study sample

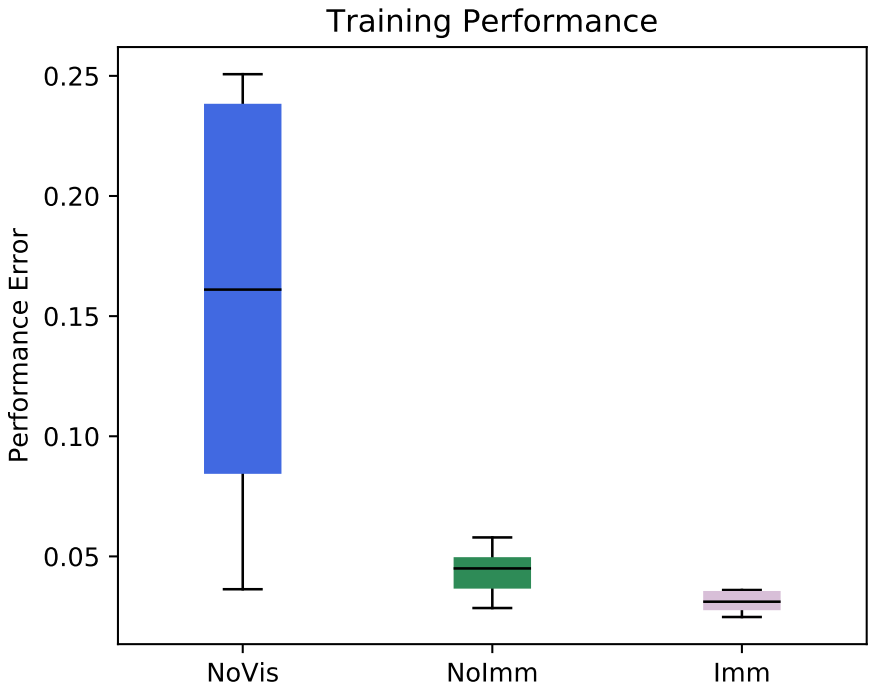


Figure 5.7: Boxplots for the performance metric, D, of each sample during training

improvement from NoVis (mean rank = 19.3), $p = 0.0468$. No significant interactions were found between NoVis and NoImm (mean rank = 14.5), $p = 0.1114$, or between Imm and NoImm, $p = 0.3238$.

5.4.2 Quantitative Analysis of Subjective Data

To better understand participants' subjective experiences during training I administered three questionnaires during the study: the NASA TLX for assessing task work loads (Hart, 2006), the UEQ for measuring overall training experience (Schrepp et al., 2014), and the Presence questionnaire for a subjective assessment of a participant's sense of presence within the virtual environment (Witmer and Singer, 1998). I provide statistical analysis of both the NASA TLX and the UEQ data below. I found no significant interactions in the data obtained from the Presence questionnaire.

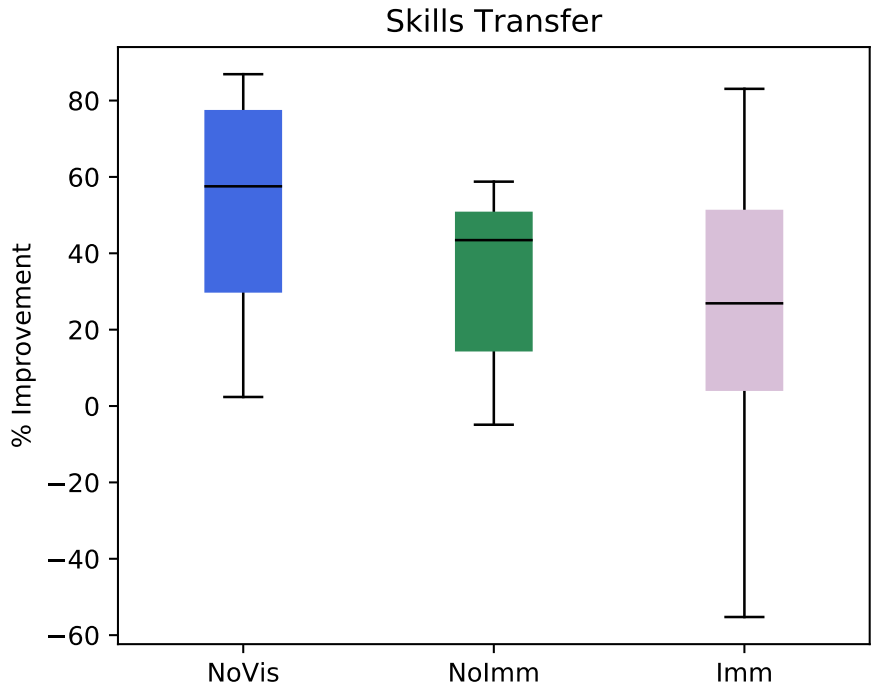


Figure 5.8: Boxplots for PI from pre-test to post-test for each sample

NASA TLX Questionnaire

The NASA TLX questionnaire allows participants to provide a subjective assessment of perceived cognitive workload during their interaction with an interface. I used the raw NASA TLX scores to measure overall workload and the six individual factors which include mental demand, physical demand, temporal demand, performance, effort, and frustration. The total workload index is summarized in the boxplots in Figure 5.9. While my hypothesis that the perceived workload would be higher within in the Imm sample due to the added complexity of the visual layer, a Kruskal-Wallis test shows that there is no significant difference between the samples, $p = 0.999$. Although there is no difference between the overall workload index, there are a couple interesting observations from individual subscales.

Figure 5.10 shows boxplots for each individual subscale of the NASA TLX. Interestingly the boxplot for mental demand indicates that NoVis may be perceived as more mentally demanding than either visual group. A Kruskal-Wallis H test shows significance for mental demand at $p = 0.065$.

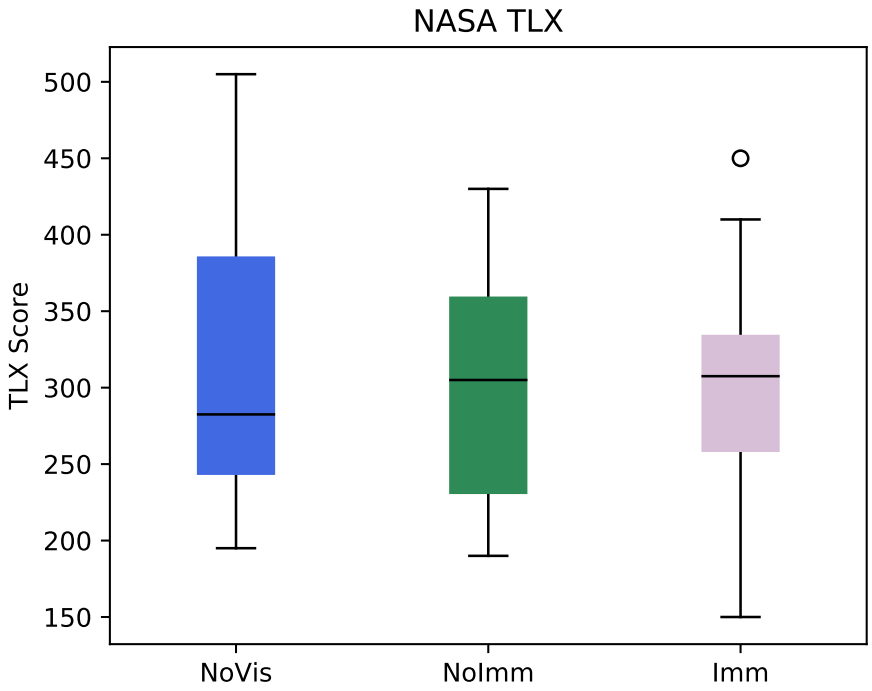
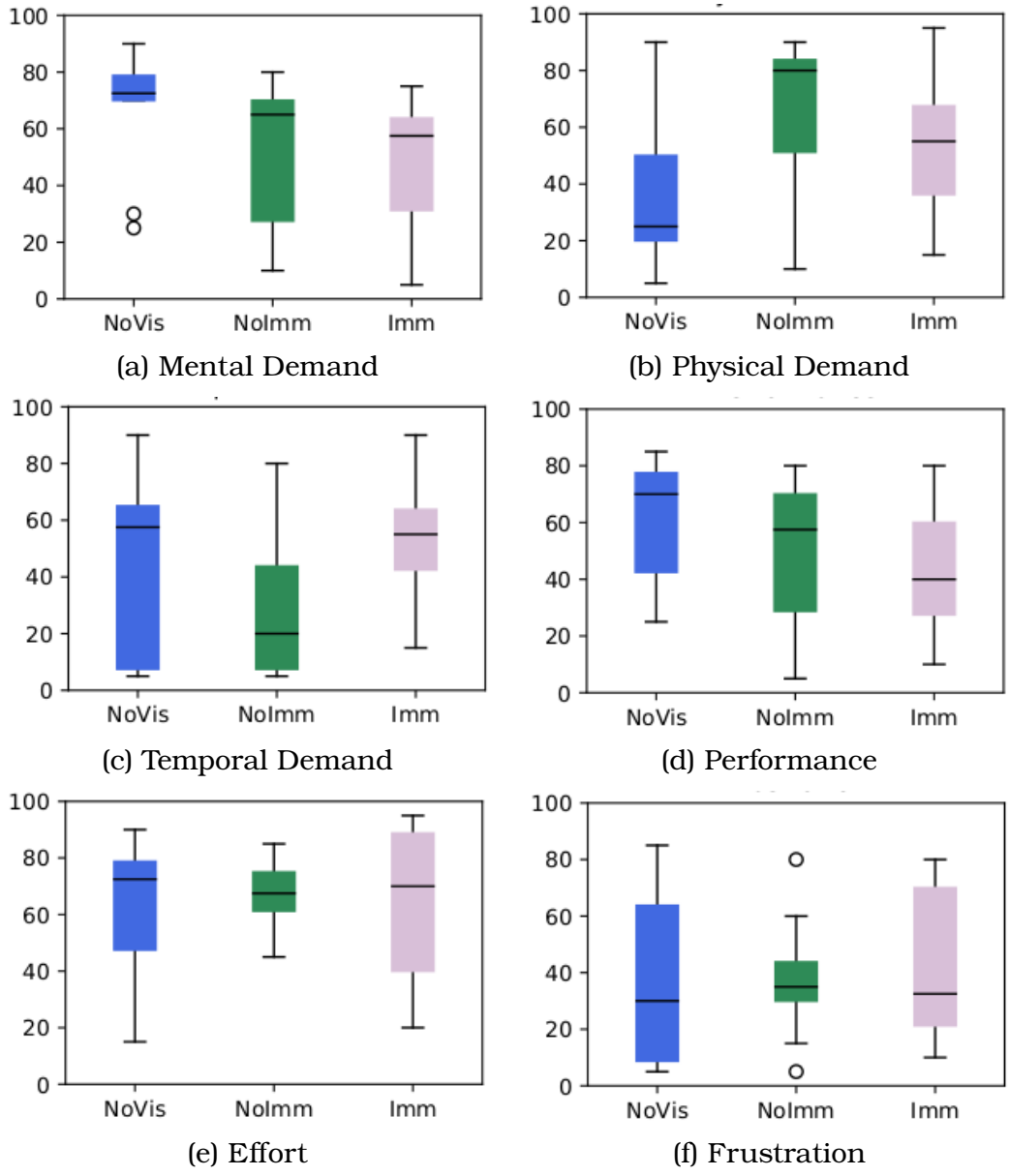


Figure 5.9: Boxplots for the total NASA TLX assessment score

A post-hoc analysis with Dunn's test with Bonferroni correction reveals NoVis (mean rank = 20.45) to have a significantly higher mental demand compared with Imm (mean rank = 11.45), $p = 0.032$. There is no significant difference between NoVis and NoImm (mean rank = 14.6), $p = 0.202$ and between NoImm and Imm, $p = 0.500$. Additionally, the boxplot for physical demand indicates NoVis may be perceived as the least physically demanding. While there is no significant difference between the groups, $p = 0.153$, a post-hoc analysis shows that physical demand for NoVis (mean rank = 11.65) is lower than NoImm (mean rank = 19.25) at a significance level of $p = 0.079$. Furthermore, NoVis is not significantly different than Imm (mean rank = 15.6), $p = 0.471$ and NoImm is not significantly different than Imm, $p = 0.500$. All other subscales show no significant interactions.

User Experience Questionnaire

The UEQ (Schrepp et al., 2014) was also administered after the training session was completed. The UEQ is a measure of the overall user experience



(a) Mental Demand

(b) Physical Demand

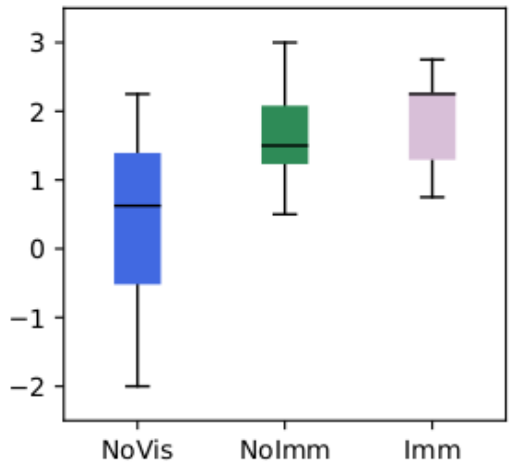
(c) Temporal Demand

(d) Performance

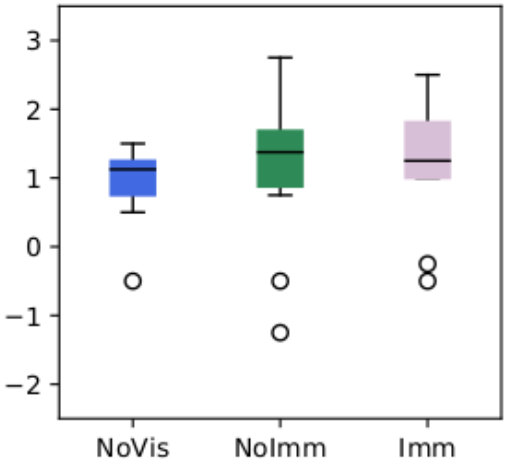
(e) Effort

(f) Frustration

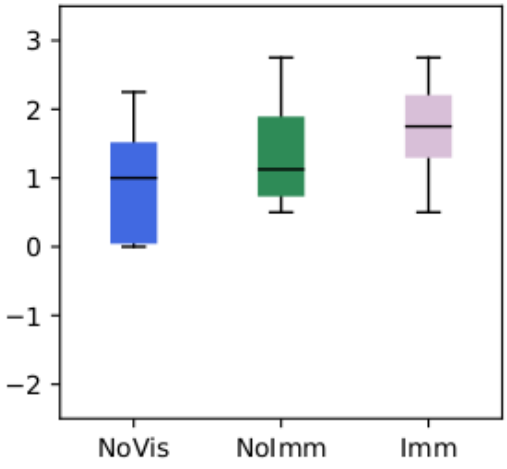
Figure 5.10: Boxplots for individual NASA TLX subscale scores.



(a) Perspicuity Scale



(b) Efficiency Scale



(c) Dependability Scale

Figure 5.11: Boxplots of individual UEQ subscale scores

and covers both usability and user experience aspects of the interaction. While there are six total quality scales, I only analyze the three usability scales: efficiency, perspicuity and dependability. The scales not included evaluate qualities not related to the goals of this study. The efficiency scale evaluates how well a user is able to perform task with the interface; the perspicuity scale evaluates how well a user is able to learn and understand an interface; and the dependability scale evaluates how well an interface meets the users' needs and expectations. The ratings of each scale score are summarized in the boxplots in Figure 5.11.

The only scale to show significant difference, determined by a Kruskal-Wallis test, was perspicuity, $H = 7.525$ and $p = 0.023$. A post-hoc analysis using Dunn's test with Bonferroni correction shows that participants ranked the perspicuity of NoVis (mean rank = 9.5) significantly less than Imm (mean rank $= 19.7$) groups, $p = 0.013$. While the perspicuity of NoImm was shown to be greater than NoVis with $p = 0.067$. There is no significant difference in the perspicuity between NoImm and Imm, $p = 0.500$. No other scales showed a significant difference in participant responses as determined by a Kruskal-Wallis test. Furthermore, post-hoc analysis showed no significant pairwise interactions within the scales.

5.4.3 Qualitative Analysis

After completing the training sessions and learning transfer assessment, participants performed a second training session in either the Imm or the NoImm environment. Participants that originally trained with Imm experienced a training session in NoImm and participants that originally trained in NoImm or NoVis completed a training session in Imm. Participants were then interviewed about their experiences after all sessions were complete. During the interviews, the participants were asked to discuss the training method that they felt would be most effective to them. Participants were also asked about any problems or challenges they encountered during the training sessions including the second training session. I ended the interview with open discussion to let the participants discuss any additional thoughts they might have.

Of the 30 participants, 21 preferred the Imm learning environment, six

indicated they would prefer NoVis, and just three preferred the NoImm environment. Most of the participants that preferred Imm, said that the visual cues helped guide them to the notes that they had a hard time finding on their own. This increased their confidence that they were performing the correct notes. Furthermore, a number of these participants found that the Imm was more precise and was able to get them closer to the note than other environments. A few participants also mentioned that they liked the immersion of the VE which helped them to concentrate on the task. Five of the six participants that preferred a non visual training had a strong musical background and were able to find the tones based on hearing alone. These participants mostly found the visuals to be distracting. These results indicate a strong preference for the Imm environment especially among participants with less music training.

In general, most participants felt it was difficult to focus on both the audio and visuals in the NoImm and Imm interfaces. Fourteen participants indicated that they paid little attention to the audio and only used the visuals to find the correct hand participation, eleven said they were able to focus on both modes of input using the visuals as guidance, and five had no discussion on this factor. One of the most distracting features, according to participants, was the virtual note changing color when the participant's hand was in the correct location. Due to the nature of the theremin, the space for the correct hand location is very narrow and holding that position for a period of time is very challenging. This resulted in the virtual note changing color frequently causing a flickering state that many participants found distracting. Participants' lack of focus on the auditory feedback, caused by an increased cognitive demand due to the new visual elements, may be one of the major contributing factors of decreased learning transfer.

Another challenge noted by participants while using the Imm environment was a lack of spatial awareness between the real world and the virtual world. One concern was the disconnect between their real world hand position and virtual hand position. Because *MR:emin* uses MIDI data from the theremin as control input for the virtual hands (instead of more complex camera-based or controller tracking) the virtual hand movement had only one degree of freedom. Most participants that mentioned this challenge found that once they accepted the difference, movements started to feel

more natural and had the illusion that the virtual hand was their hand. A more concerning factor was a lack of spatial awareness between the participants physical body and the virtual theremin. Although the theremin was calibrated to be located in precisely the same location in the VE and the real world, many participants felt a disconnect with the spatial awareness between their body and the theremin. This could be attributed to the participants' bodies not being represented in the virtual world. This spatial disconnect between the physical body and the virtual elements may be another factor in the decreased learning transfer.

5.5 Discussion

The goal of this chapter was to enhance the fields understanding of the benefits and drawbacks of XREMIL. To study these effects I presented an XREMIL environment, *MR:emin*, that augments a physical music instrument, the theremin, with visual cues for guidance and feedback. This design allows a student to practice with the instrument they are learning while benefiting from the affordances of a visual layer provided by XR technologies. While the visual cues may make practice easier, it is important to validate that the additional feedback does not negatively affect the learning transfer of auditory skills necessary for learning music. Using the *MR:emin* environment I designed and administered a user study for evaluating the effectiveness of immersive music learning and the factors, such as training performance, learning transfer, and user experience that lead to a positive learning experience.

5.5.1 Factors of *MR:emin* on Learning Transfer

Statistical analysis of the training data provided clear evidence that the *MR:emin* training environments, Imm and NoImm, provided users the most precise guidance and feedback, leading to more accurate performances during training, supporting hypothesis H_1. Precise training, however, doesn't necessarily lead to improved learning transfer. Imm training, in fact, lead to a smaller percentage of improvement than NoVis, with statistical significance at $\alpha < 0.5$, supporting hypothesis H_2. And while there was no

significant difference at $\alpha = 0.1$, learning transfer in the NoImm environment was close to being significantly less than the NoVis environment. I believe that the decreased learning transfer of the *MR:emin* environments was likely due to the the addition of visual elements to the training environment thats caused participants to focus their attention on the visual rather than audio feedback. Interviews with participants strengthened this view, as they often mentioned that their focus was on matching the virtual hand position to visual feedback. Some participants even mentioned not hearing or noticing the audio feedback at all. This could potentially be avoided by carefully designing interfaces that encourage students to use the audio feedback for evaluating the correctness of an action and using the visual cues mostly for guidance. For example, this could be achieved in *MR:emin* by removing the visual feedback of the virtual note that indicates when a hand is in the exact location by changing colors. Then the visual element is only used to provide guidance to the general note location, encouraging participants to listen to the auditory feedback to find the precise location. In this case, the visual cues would provide guidance while the audio provides feedback on correctness.

Statistical analysis of the training data provided no support for hypothesis H_3 since there was not a significant difference in learning transfer between the Imm and NoImm training environments. Imm, however, was the only environment in which some participants performed worse after training. The further decrease in learning transfer may be attributed to some participants' lack of spatial awareness between the VE and the real world, as expressed by a number of participants. This makes a strong case for the use of immersive augmented reality for music learning environments that are able to overlay visual cues directly on the physical instrument (rather than a virtual representation, as in our case), affording the student more awareness of their spatial surroundings while practicing. Immersive AR resources, however, are not as readily available to consumers at this time as compared with VR and MR devices. Therefore, designers of immersive music learning environments for VR and MR should build in some mechanism to provide students with a frame of reference to the physical environment.

Carefully designed interfaces may help to alleviate some of the adverse effects of factors that contribute to decreased learning transfer, namely the

shifting of focus to visual feedback and lack of spatial awareness. These factors, as well as the novelty of the immersive environments, may also diminish as students become familiar with the environment and improve their cognitive abilities with more practice. In other words, as students gain more experience within the immersive learning environments, they will gradually learn to process both the visual and auditory feedback, as well as, become more comfortable with matching the spatiality of the VE with the real world. Thus, I believe the enhanced training precision of the *MR:emin* environments may outweigh the initial decline, but this is left for future research.

5.5.2 User Experience of *MR:emin*

Learning transfer is not the only factor that contributes to a successful learning environment, user experience also plays a large role. Providing a positive experience to the user may enhance the learning process through increased student engagement and motivation (Dalgarno and Lee, 2010). The analysis of the UEQ responses showed that only the perspicuity of Imm was significantly different than NoVis indicating that the immersive learning environment made it easier for participants to understand how to perform the training task. The other subscales of the UEQ showed no significant difference for participants' experience between training environment but interviews with participants showed that the majority of participants preferred the Imm environment, and found it the most engaging, over both the NoImm and NoVis environments, supporting hypothesis H_4. One caveat to this is that participants with more advanced musical abilities preferred the NoVis environment. These were generally participants with a well trained ear that didn't need the visual cues to find the correct notes and thus found the visuals overly distracting with no added value to the training. Participants that preferred the Imm environment indicated that it improved their confidence that they were doing the right thing leading to increased satisfaction during practice. Improved engagement and satisfaction can lead to increased product adoption and use; so although learning transfer in Imm may be slightly worse than non visual environments participants are more likely to continue practicing making up for the deficit.

5.5.3 Threats to Validity

The main threat to the internal validity of the study is participant selection. While participation was open to the general university population, most participants were from the faculty of engineering and may be more accustomed to new technologies, as well as, more proficient with computers. Furthermore, participants that volunteered for the study were likely to be interested in music or XR and may be more motivated to do well than the general population. Another potential threat to validity deals with the participants' level of musical listening skills. I did not screen or test participants on their pitch hearing beforehand. I attempt to mitigate the effects of the heterogeneity among participants in pitch hearing by utilizing percentage improvement in performance as the statistical measure.

This user study, as with any empirical study, is subject to some threats to validity. Another threat in the study results is concerned with the external validity of the results. As this study is intended to provide insight into general immersive music learning environments, the results should be generalizable outside of *MR:emin*. I previously outlined that the uniqueness of the theremin provided a number of benefits for the study (see section 5.1). This uniqueness, however, challenges the external validity of the study in that traditional instruments have more natural haptic interactions. I attempted to mitigate this threat by focusing on the auditory factors of music learning which is an important aspect of any instrument.

5.6 Conclusion

In this chapter, I present an immersive music learning environment and a user study to better understand the factors of learning music in such an environment. The results of the study present evidence that music training in an immersive environment reduces performance error during training. I found, however, that music learning environments with visual feedback may slightly interfere with the auditory learning process. During interviews with participants of the user study I found that many trained in a visual environments focused most of their attention on the visual cues and neglected to focus on the auditory feedback. But the guidance provided by the visual

cues improved participants' confidence that they were performing correctly during the training sessions, potentially leading to improved engagement with the training environment. Furthermore, most of the participants had either no or limited previous engagement with XR technologies, suggesting that the novelty of the environment may have influenced participants' performance. Therefore, with more practice and carefully designed interfaces that place auditory feedback over visual feedback, XR could improve the music learning process. To validate this hypothesis, an extension of this work would be a similar user study in which users are trained over multiple sessions. Another extension would be to perform a user study with modifications to the visual cues that emphasize listening to the auditory feedback and reduce the focus on the visual layer.

The user study I present in this paper is designed around an immersive learning environment for the theremin but the results are meant to inform the design of the an immersive learning environment for the previously discussed CAMIT system to detect and correct hand posture mistakes during piano practice. The results of this study motivate a few general guidelines for the design of immersive music learning environments that will help guide our design decisions. First, *design interactions that encourage users to focus on audio over visual feedback*, this means limiting visual cues and feedback to the bare minimum, and reducing all visual noise from the VE. Along these lines I also recommend to *use visual cues for guidance rather than feedback*. Throughout the training sessions I observed that participants were overly focused on minimizing errors in their visual feedback rather than the audio feedback. While this was the goal of the visual interface, by trying to perform so precisely the users focused less on the audio and more on getting the visual state perfect. Furthermore, to improve users' spatial awareness, immersive music learning environments should *design environments towards the AR end of the RV continuum*. An MR approach, as opposed to full AR, was selected for this study due to the lack of general consumer accessibility to AR devices. Ideally as AR technologies improve and become more accessible, see-through AR devices will be used for immersive music learning environments. Finally, more experienced musicians may find the visual more distracting than helpful so *design visual cues with users' musical experience in mind*. Designing with these guidelines in

mind will contribute to immersive learning environments that enhance the music learning process by improving the effectiveness of student practice and by increasing student engagement.

Chapter 6

Virtual Environments for Musical Expression

In the previous chapters I discussed the application and effects of XR and related technologies to the field of CAMIT for enhanced musical learning experiences. Through this experience, I identified guidelines for effective XREMIL design. In this chapter, I explores the application of XR to NIME through the development of VEME to address RQ_4. To enable rapid prototyping and make VEME development more accessible, I present a toolkit, OSC-XR, to address challenges associated with their development. After describing details of the toolkit, I present three use cases in VEME to explore the development process. The development of the use cases informed the design of OSC-XR as well as showcases it's capabilities. This work resulted in a publication[1] as well as two open source toolkits to facilitate VEME development[2,3].

[1] D. Johnson, D. Damian and G. Tzanetakis. OSC-XR: A Toolkit for Extended Reality Immersive Music Interfaces. In *Proceedings of the 2019 Sound and Music Computing Conference*, To Appear 2019.

[2] D. Johnson. UnityOscLib: A Simple Open Sound Control (OSC) library for Unity. `https://github.com/fortjohnson/UnityOscLib`, 2019.

[3] D. Johnson. OSC-XR: A Toolkit for Extended Reality (XR) Immersive Music Interfaces. `https://github.com/fortjohnson/OSC-XR`, 2019.

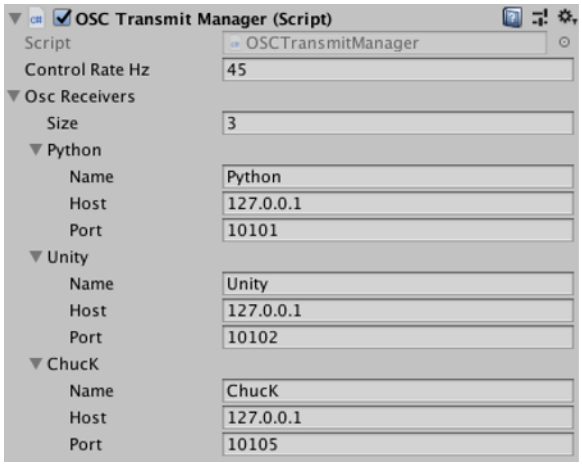
6.1 Introduction

Jaron Lanier's performance of *The Sound of One Hand* (Lanier, 1992), was remarkable in that he was able to simultaneously play multiple instruments to perform music that could not easily have been performed with traditional instruments. His performance showed the potential for MusE-XR, but since then there has been limited research exploring the musical interactions afforded by XR technologies. When Serafin et al. (2016) recently surveyed the state of art in virtual reality music instrumentss (VRMIs), the number of interfaces available was fairly small. The capabilities and relatively few design constraints of XR create the potential for a wide array of immersive interfaces based on any of the four categories of music interfaces proposed by Miranda and Wanderley (2006): *Augmented Musical Instruments*, *Instrument Like Controllers*, *Instrument Inspired Controllers*, and *Alternate Controllers*. With such broad possibilities, more research is needed to increase our understanding of the affordances of immersive environments and interaction techniques best suited for MusE-XR.

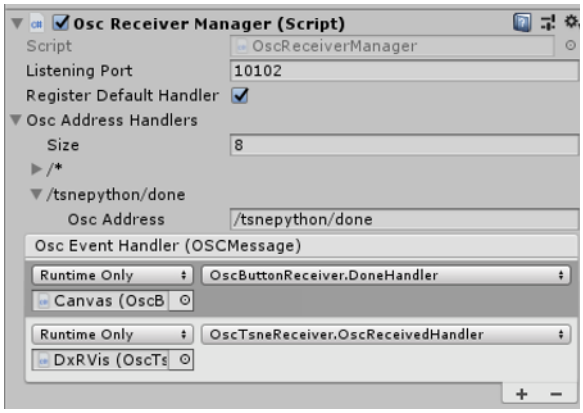
To support further research into immersive interfaces for music, I present OSC-XR, a toolkit for rapidly prototyping immersive musical environments in XR using Open Sound Control (OSC), a communication protocol widely used in audio software (Wessel and Wright, 2002). Influenced by multi-touch OSC controllers, OSC-XR provides developers with a wide range of readily available components in order to make designing immersive environment more accessible to researchers and sound designers. In this chapter, I discuss the infrastructure of OSC-XR, validate its generated data by comparing with a popular multi-touch OSC controller, and present three environments developed to demonstrate its capabilities for immersive interface design.

6.2 Unity OSC Library

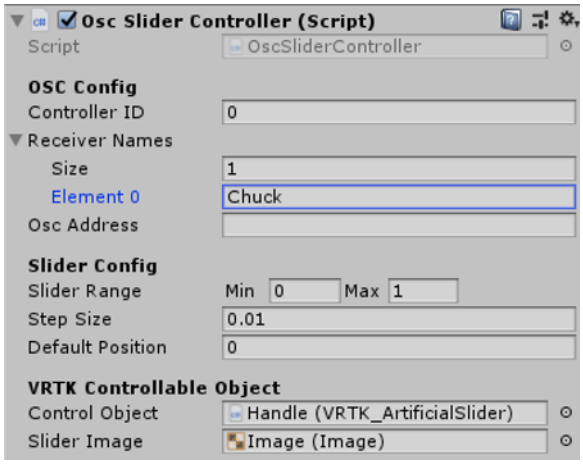
To implement OSC in Unity, many projects have used Jorge Garcia's UnityOSC library (Garcia, 2019). I have found this library to be somewhat difficult to integrate in new projects; further, the API lacks a Unity Inspector interface for simplified configuration. To simplify the OSC configuration



(a) The OSC-XR Transmitter Interface



(b) The OSC-XR Receiver Interface



(c) The OSC-XR Slider Interface

Figure 6.1: Example Unity Inspector Interfaces from the OSC-XR toolkit

process, I present a library, UnityOscLib, that builds on Garcia's core OSC classes and integrates them into the Unity development work flow. The new library simplifies configuration by implementing separate `MonoBehaviour` (a Unity base class from which all Unity scripts must be derived) classes for receiving and transmitting OSC messages. The configuration process has also been simplified by exposing OSC properties in the Unity Inspector as well as through Unity scripting. In this section, I introduce the library and complete details, including examples, can be found on the project's Github repository[4].

The `OscTransmitManager` is a Unity `MonoBehaviour` that handles all aspects of transmitting OSC messages. To send OSC messages from a Unity application, add the OSC transmit manager to one `GameObject` and configure connection information for one or more OSC receivers. OSC receiver configuration details are exposed through the Unity Inspector, as shown in Figure 6.1a, in addition to the scripting interface using the `AddReceiver` method. Once configured, the environment is ready to transmit OSC messages using `SendOscMessage` or `SendOscMessageAll`.

The OSC transmit manager also implements an optional control rate feature, to configure the frequency of OSC message transmission. Transmitting OSC messages may be triggered by specific events that only occur periodically, such as collision events, but they may also be triggered continuously, for example when an objects position is changing. This type of continuous data is generally calculated at a rate specified by Unity's `Update` or `FixedUpdate` messages. With XR these messages typically occur at around 90 FPS or faster as technologies improve. Audio applications may not be able to handle incoming messages at this rate. The UnityOscLib control rate feature is implemented to limit the rate OSC messages are transmitted. To use this feature, developers should register a method that transmits OSC messages with the`OnSendOsc` event of the `OscTransmitManager`. Any methods registered with`OnSendOsc` will be called at the control rate specified in the Unity Inspector.

The `OscReceiverManger` is a Unity `MonoBehaviour` class that manages the routing and handling of incoming OSC messages. To receive OSC messages in a Unity application, add the OSC receiver manager to one `GameObject` in

[4] `https://github.com/fortjohnson/UnityOscLib`

Name	Description Example OSC Message
OscSlider	A slider prefab with position mapped to a configurable range, see Figs. 6.1c and 6.3. `/slider/value 1 4.5`
OscPad	A drum prefab with pressed and released events including an optional velocity, see Figure 6.3. `/pad/pressed 1 1.5`
OscGyro	A virtual gyroscope prefab for sending angular velocities normalized to a range $[0, 1]$. `/gyro/velocities 1 .9 .7 .5`
OscTransform	A script for sending transform data via OSC. `/trans/local/pos 1 0.5 1.3 2.0`
OscTrigger	A script for sending Unity Trigger events; includes an ID and position information for the triggering object. `/trigger/enter 1 0.5 0.4 1.0 2`

Table 6.1: Examples of available OSC-XR controller prefabs and scripts

the scene and configure the receiver with the port to listen on, see Figure 6.1b. OSC address routing is implemented using Unity Events for configuration in the Unity Inspector as well as using delegate events for C# scripting. To route messages based on in the inspector, UnityOscLib exposes an interface in the inspector to add any number of OSC addresses and one or more handler methods for each address, see Figure 6.1b. Additionally, the receiver manager's `RegisterOscAddress` method is used to add OSC addresses and event handlers through Unity scripting. All OSC event handler methods used should accept a UnityOscLib `OscMessage` as an argument. This implementation provides flexible implementation for adding OSC handling during environment design or at runtime.

6.3 OSC-XR

The main contribution of this chapter is the OSC-XR toolkit for designing immersive XR environments for music control. It is developed using Unity and UnityOscLib to provide sound designers a simple interface for prototyping interactions in immersive environments. The OSC-XR toolkit contains two main components for building environments, 1) a set of scripts that

can be attached to any Unity `GameObject` to transmit the object's state via OSC and 2) a set of prebuilt music controllers, called controller prefabs, for transmitting control data via OSC, similar in concept to widgets in TouchOSC. With this infrastructure, developers with limited Unity experience can quickly design immersive music environments through the use of the controller prefabs. Furthermore, more experienced developers can easily extend custom`GameObjects` with OSC capabilities through the scripting interface. Finally, the robust Unity platform affords customization and extension of any OSC-XR components to those familiar with Unity and C#. The flexible design of OSC-XR, combined with the power of Unity, supports rapid prototyping to make designing immersive environments quicker more accessible.

OSC-XR was developed using Unity (Unity, 2019) and tested using the Samsung Odyssey Windows Mixed Reality Headset with SteamVR. By making use of the well known Virtual Reality Toolkit (VRTK), OSC-XR should work with any platforms supported by VRTK enabling multi-platform support. The remainder of this section discusses the OSC-XR infrastructure. The details provided hear are intended to give the reader high level understanding of how the toolkit is structured but the readers are encouraged to visit the project's Github repository[5] for complete details, including video examples.

6.3.1 OSC Controller Prefabs and Scripts

Adding OSC controller prefabs to a Unity scene is the quickest way to get started with OSC-XR. To implement a controller simply add the prefab from the `OSCXR/Prefabs` folder to the Unity game hierarchy. Once added to the scene, modify the object's transform as desired. At this point the object is ready to use in the environment. For additional configuration each controller exposes a set of properties in the Unity Inspector, see Figure 6.1c. Table 6.1 lists the descriptions of a few of the available OSC controller prefabs, including an example OSC message for each. Developers can further customize the controller prefabs using Unity tools. For example, the visual aspects of any of the controller prefabs can be modified by configuring

[5] `http://github.com/fortjohnson/OSC-XR`

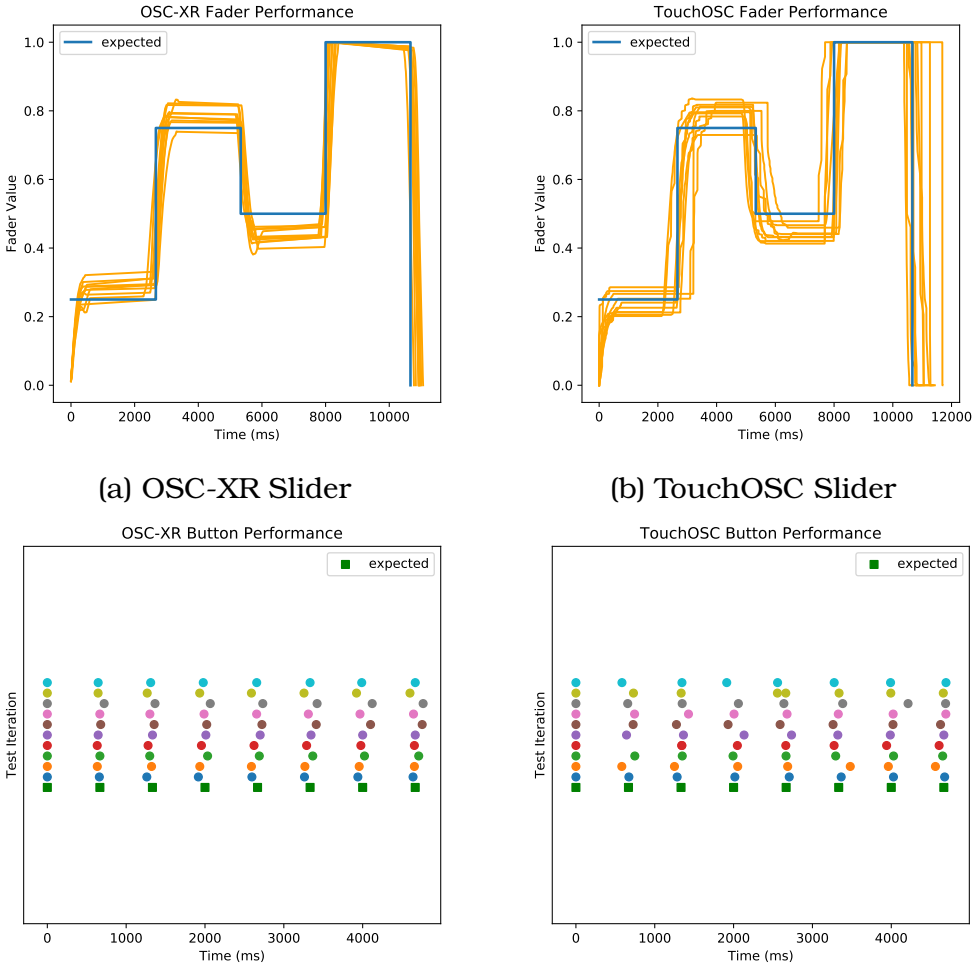
the Unity components that comprise each object, such as the meshes or materials.

The OSC-XR scripting interface allows developers to quickly add OSC capabilities to any `GameObject` by attaching any of the readily available controller scripts to the object. Each of the scripts models a predefined behaviour for triggering and sending OSC messages. By default adding an OSC controller script to a `GameObject` uses that object's state for creating and transmitting OSC messages. This can be overridden on most scripts by updating the `Control Object` property of the script with a different `GameObject`, in which case, the state of the configured `Control Object` will be used instead. This is useful when building a composite object where the tracked object is not the top level object. For example, the slider controller prefab implements this design in which case the state of prefab's handle is used for control data, as seen in Figure 6.1c. Table 6.1 lists the descriptions of a few of the available OSC-XR controller scripts, including an example OSC message for each.

Designers wishing to build their own OSC controller scripts should extend OSC-XR's `BaseOscController`. This class includes a number of base properties for OSC configuration, the controller ID and the OSC address, as well as methods for sending OSC messages. Furthermore, the class automatically registers the method, `ControlRateUpdate` to support transmitting OSC messages at the control rate specified in the OSC transmit manager. Any controller script that needs to send data at the configured control rate should override `ControlRateUpdate` with a method that generates and transmitting OSC data. Each custom script should extend these options as needed to achieve the behaviour being modeled.

6.3.2 Control Data Validation

To ensure that data generated by OSC-XR is consistent with users' expectations, I employed two simple user tasks for comparing OSC-XR with TouchOSC. An OSC receiver is implemented to log data generated by each task for an analysis of user performance. One task utilizes a slider controller (or fader widget in TouchOSC) to validate the control precision of the different applications. The second task utilizes a pad controller (or button widget



(a) OSC-XR Slider

(b) TouchOSC Slider

(c) OSC-XR Pad

(d) TouchOSC Pad

Figure 6.2: The results of control data validation for slider controllers and pad controllers.

in TouchOSC) to evaluate rhythmic control of the different interfaces. In addition to validating control data from OSC-XR, the results of the tests provide a baseline for evaluation of virtual object controllability in immersive environments.

Slider and Pad Evaluation

To perform the slider evaluation task, a user sets the position of the slider to specific values at regular time intervals. For this evaluation, the task requires setting the values of the sliders to 0.25, 0.75, 0.50, and 1.00 (in that

particular order). The user is required to transition the slider to each value on every fourth beat at a tempo of 90 beats per minute (BPM) indicated using a metronome. To perform a baseline analysis of the control data, a user performed the task ten times in both OSC-XR and TouchOSC. The output of the user's performance is then compared to signal representing the expected data, Figures 6.2a and 6.2b show the results of each run for both applications overlaid with the expected output. The data is compared quantitatively by calculating the euclidean distance between the actual and the expected output to calculate an error value. This value is averaged over all iterations for a final error metric. Table 6.2 lists the average errors of this task for both interfaces.

To perform the pad evaluation task, a user presses a pad controller for eight beats at a tempo of 90 BPM using a metronome to keep time. As in the previous task, baseline analysis of the data is captured with a user that performs the task ten times. Results of each iteration are shown in Figures 6.2c and 6.2d, for OSC-XR and TouchOSC respectively. Each iteration is represented as a row of dots where each dot in the row indicates a pad pressed event. For comparison the expected beat times are shown with the green squares in the bottom row. The error for each iteration is calculated as

$$\frac{\sum_{n=1}^{N} \mid t_{exp} - t_{act} \mid}{N} \tag{6.1}$$

where N is the number of beats per iteration, t_{exp} is the expected time of the beat, t_{act} is the actual time of the pressed event from the user. The errors are averaged over all iterations for the final error metric. The error results for both interfaces are listed in Table 6.2.

Discussion

Results of the slider evaluation provide a baseline comparison of OSC-XR with TouchOSC. Initial analysis of the data shows similar performance between both applications even though the interactions are slightly different. To move a slider in TouchOSC, a user slides their finger across the surface to the new location. Whereas, the OSC-XR slider requires an additional grab interaction to take control of the slider handle before moving it towards its destination. Overall, the OSC-XR slider error is slightly greater

	OSC-XR	TouchOSC
Slider	3.43	2.98
Pad (ms)	35.2	52.0

Table 6.2: Average errors for each evaluation task

than that of TouchOSC. I compensate for this in OSC-XR by adding a display prefab to the slider for additional feedback. While the interactions required for manipulating sliders are different, this evaluation shows that OSC-XR sliders may perform as well as multi-touch sliders and generate data that is consistent with an application sound designers may already familiar with.

OSC-XR also requires a different technique for interacting with pads due to the lack of haptic feedback. When pressing a pad in OSC-XR users are not provided the same haptic response naturally afforded through interaction with physical objects. Instead users must rely on wrist action and hand controller momentum to control rhythm. Initial evaluation of the pad controller indicates this may not adversely affect rhythmic performance. Results show that the user was able to perform slightly more accurately with OSC-XR. This may be a result of the user relying on wrist action for control rather than pressing a pad with a single finger. Although a larger study is needed to confirm any hypotheses, users may expect rhythmic control from OSC-XR that is consistent with TouchOSC.

6.4 OSC-XR Use Cases

In this section, I discuss three prototype use cases for immersive environments developed with OSC-XR. Prototyping the environments helped inform the design OSC-XR. Furthermore, the use cases demonstrate the capabilities of the toolkit in different scenarios providing readers ideas on how OSC-XR might be used for their own projects.

6.4.1 The Sonic Playground

The Sonic Playground is an immersive environment that explores a variety of OSC-XR controllers. The playground is composed of multiple zones

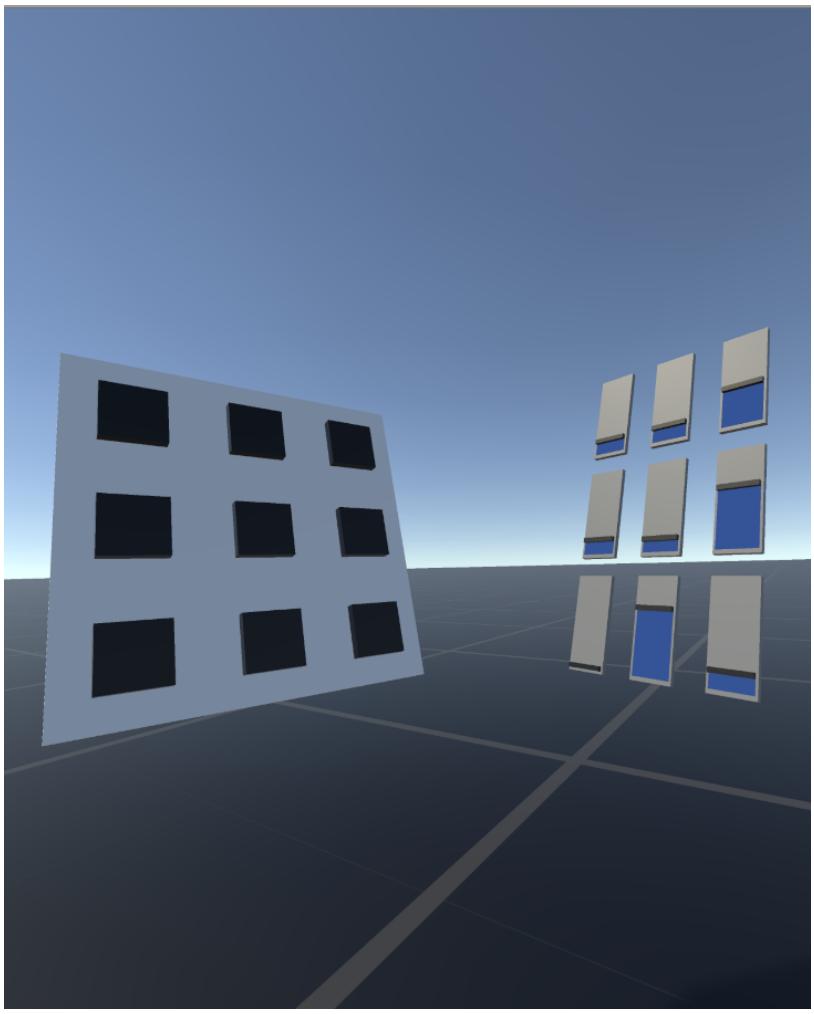


Figure 6.3: The Sampler Zone VEME which includes OSC-XR pads to trigger audio samples and corresponding OSC-XR sliders to control sample playback rate.

each with a different performance environment. Users are able to navigate between the zones using teleportation, providing the ability to quickly move between different performance environments. The Sonic Playground is designed to explore and demonstrate musical interaction with OSC-XR controllers that communicate with an external audio programming environment.

The Sampler Zone, seen in Figure 6.3, is an immersive sampler environment composed of a 3×3 matrix of pad controllers, to trigger sample playback, and a corresponding matrix of sliders, for additional control of the samples. Pads are configured to send the controller ID as well as pressed

and released with included velocity for mapping to sample volume. Each slider is configured to send a value ranging from 0.25 to 5.0 mapped in ChucK to the playback rate of the corresponding sample. Pad and slider events are all mapped to a corresponding sample using the controllers' IDs. This environment was developed as a proof-of-concept to demonstrate and explore the affordances of typical music controllers in immersive environments.

The first thing to notice in this environment is the size of the objects. Using input controllers for interaction requires the use of large objects as users lose the dexterity that is naturally afforded through interactions using the hand. Integrating hand tracking devices, such as the Leap Motion, may allow for designing dexterous interactions. Another challenge of performing in XR is the lack of haptic response to physical actions, such as tapping a pad. Even with these challenges, virtual pads in a musical environment afford their own interaction style with large expressive motions and gestures. The evaluation of the pad controller, discussed in Section 6.3.2, indicates that rhythmic control may not be severely affected through the lack of haptics and in this environment I learned the lack of haptics affords an expressive playing style.

The Sonic Objects zone is an environment for prototyping interactive sound environments. It is composed of various OSC-XR controllers that are readily available to communicate with an audio programming environment, such as ChucK. The environment affords the rapid prototyping of interactive sound design by combining OSC-XR's ability to easily add new controllers and interactions with the power of ChucK's development environment to quickly iterate on sound design.

One of the interesting affordances of immersive music environments I explore is the combination of real life physics based interactions with "impossible" interactions that ignore physics. For example, using physics I can toss objects around or stack and lean them on each other to create interesting soundscapes with generative audio patches. Sometimes, however, a user may want to have more control over when parameters of an audio patch stop as they reach a desired state. By ignoring the physics of an object I can lock it in space to immediately stop it from sending OSC messages. For example, an OscGyro object will always send angular ve-

locity data as its being moved, but a user may want to lock in the sound parameters before releasing the object. With this in mind, I decided to add an interaction to freeze the OscGyro anywhere in space. Once frozen the object will be suspended in space until the user grabs the object to move it again. Another interesting affordance I discovered through prototyping in this environment is the ability to easily add automation to controllers through Unity components, such as animation or particle systems. For example, the strongly timed behaviour of particle systems allows for particles to collide with an OSC Trigger controller for initiating musical events at rhythmic intervals. Furthermore, the movement of particles within the controller may be mapped to other audio parameters, such as frequency. These examples show how OSC-XR supports rapid prototyping for exploring and creating new musical interaction techniques in XR.

6.4.2 Virtual Hyperinstruments

In the NIME community it is common to augment a traditional instrument with sensors to extend its capabilities. Machover and Chung (1989) first presented work on this concept with their hyperinstruments in 1989. Typically hyperinstruments extend traditional instruments with direct augmentation of an instrument, such as a violin, with physical sensors (Overholt, 2005). Physical modification of an instrument can be invasive to it's design, therefore, non-invasive techniques have also been developed for augmentation without physical modification, through the use of cameras and depth sensors (Trail et al., 2012). These techniques use gesture detection and object tracking for added sound control but provide no visual signifiers to indicate the location of control objects. This is seen in the work of Trail et al. (2012) in which they augment a vibraphone with virtual faders that are controlled using mallet tips tracked by a Kinect. Because there are no computer generated signifiers, the fader locations are mapped to the vibraphone keys to signify control locations. Implementing XR in their system would have allowed the authors to add an additional visual layer to enhance visual feedback.

I previously explored the virtual hyperinstrument, in Chapter 5, by augmenting a physical theremin with virtual objects to visualize the pitch space

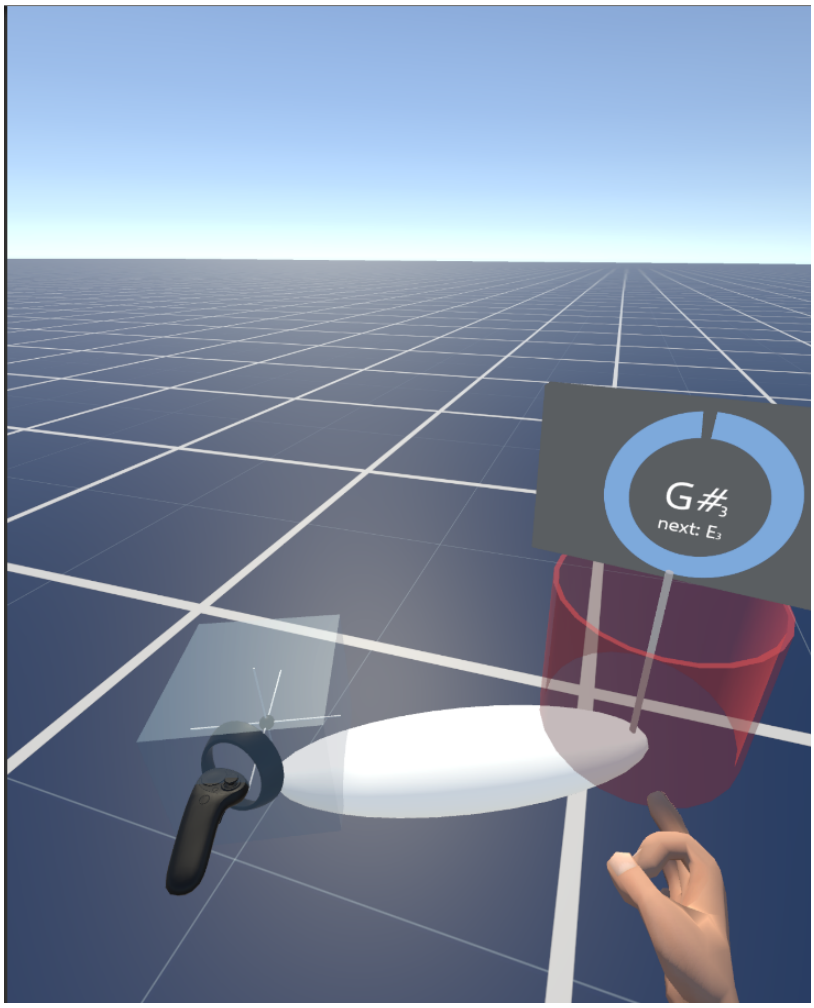


Figure 6.4: Hyperemin, a virtual theremin hyperinstrument with real-time ASP controlled with an OSC-XR 3D Grid.

for music tutoring. That work is extended with Hyperemin, a virtual augmented theremin. OSC-XR controllers are added to the Hyperemin environment to provide real-time control of ASP of the theremin audio. Audio from the theremin is routed to a ChucK patch for playback and audio processing. An OSC-XR 3D Grid controller is added to the environment to control the audio processing, allowing a performer to play the theremin while also controlling audio processing parameters. Currently, the interaction requires an immersive XR HMD and controllers, which may be intrusive to performance but the addition of a LeapMotion sensor, or use of a HoloLens with hand tracking, would address this. In addition to adding sensors directly to the instrument, one of the affordances of XR is the ability to place

objects anywhere in the space allowing users to create a customizable control interface not limited to pedals, small device displays or other physical input controllers.

The Hyperemin environment explores the capabilities of OSC-XR for augmenting physical instruments with virtual objects. As XR technology improves I expect that augmenting more traditional instruments will become more accessible. For example, with proper tracking technology it would be possible to attach an OSC-XR Gyro object to the head of a violin and a set of pads to the body adding additional control without physical modifications.

6.4.3 Immersive Vis Control

OSC-XR was designed with music interfaces in mind but its support for rapid prototyping make it ideal for designing other immersive environments requiring parametric control and distributed communication. In this use case, I explore the use of OSC-XR in an immersive visualization environment. With the emergence of XR technologies, there has been trend of research towards immersive environments for information visualization Marriott et al. (2018). With this comes the need to rapidly prototype interaction techniques to support the design of intuitive immersive interfaces.

To explore interaction needs of immersive analytics environments, I implemented a 3D visualization of the GTZAN music genre dataset (Tzanetakis and Cook, 2002). To visualize the high dimensional data in 3D, the 52 spectral and timbral features of each song in the dataset are transformed into 3D coordinates using t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton, 2008). To visualize the data, I integrate OSC-XR with an immersive visualization toolkit, DxR (Sicat et al., 2019). With DxR I was able to quickly develop an immersive scatterplot visualization using the t-SNE data. While DxR has an interface for controlling the visualization, it is limited to point and touch based interactions. Integrating OSC-XR into this environment allows us to quickly prototype new interfaces to control the visualization and augment it with with additional functionality.

Using OSC-XR, I designed a prototype interface to manipulate the visu-

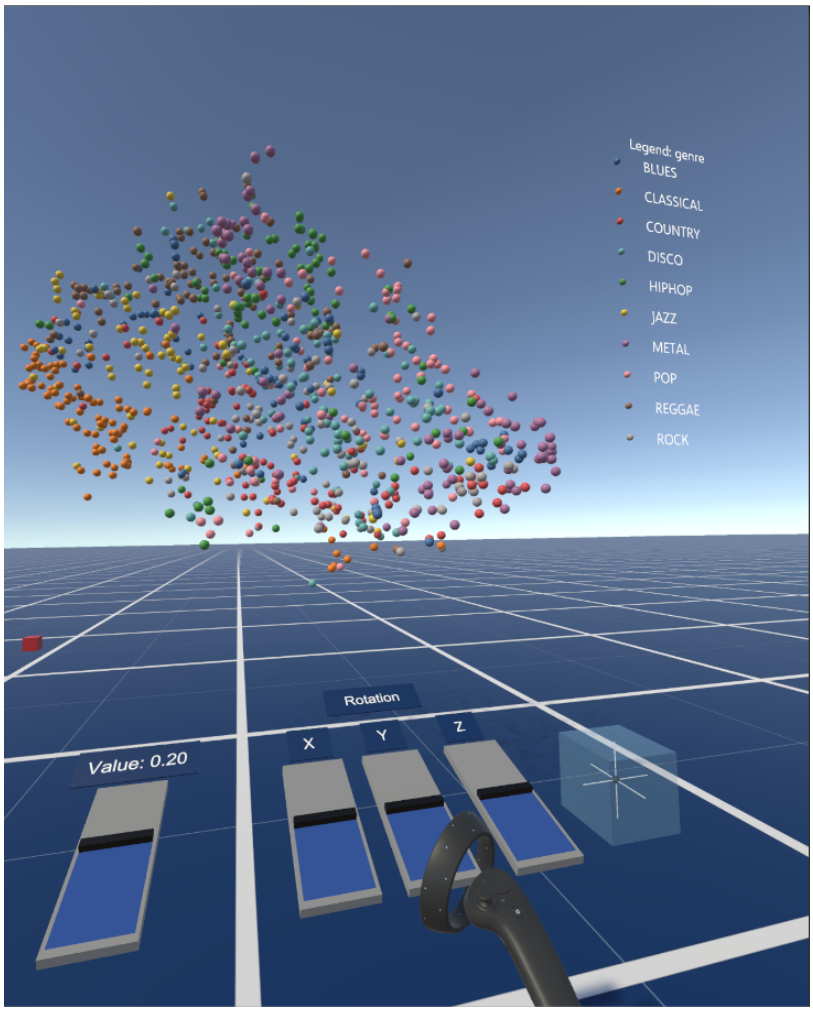


Figure 6.5: The t-SNE view control interface with OSC-XR sliders to control T-SNE view parameters such as rotation and scale.

alization using two control panels. The control panels are designed using sliders and other OSC-XR objects to control various aspects of the visualization. The interface is composed of two panels, the main panel for manipulating the view of the visualization, shown in Figure 6.5, and a secondary panel for controlling t-SNE parameters, which was not previously possible using DxR alone, shown in Figure 6.6. The main panel presents users with a set of sliders to directly manipulate view parameters such as zoom and rotation. Since this panel affects the visualization in real-time and would be utilized most frequently by a user for data analysis, it is oriented such that a user is facing the visualization while interacting with the controller. The t-SNE control panel is placed to the left of the user as the controllers

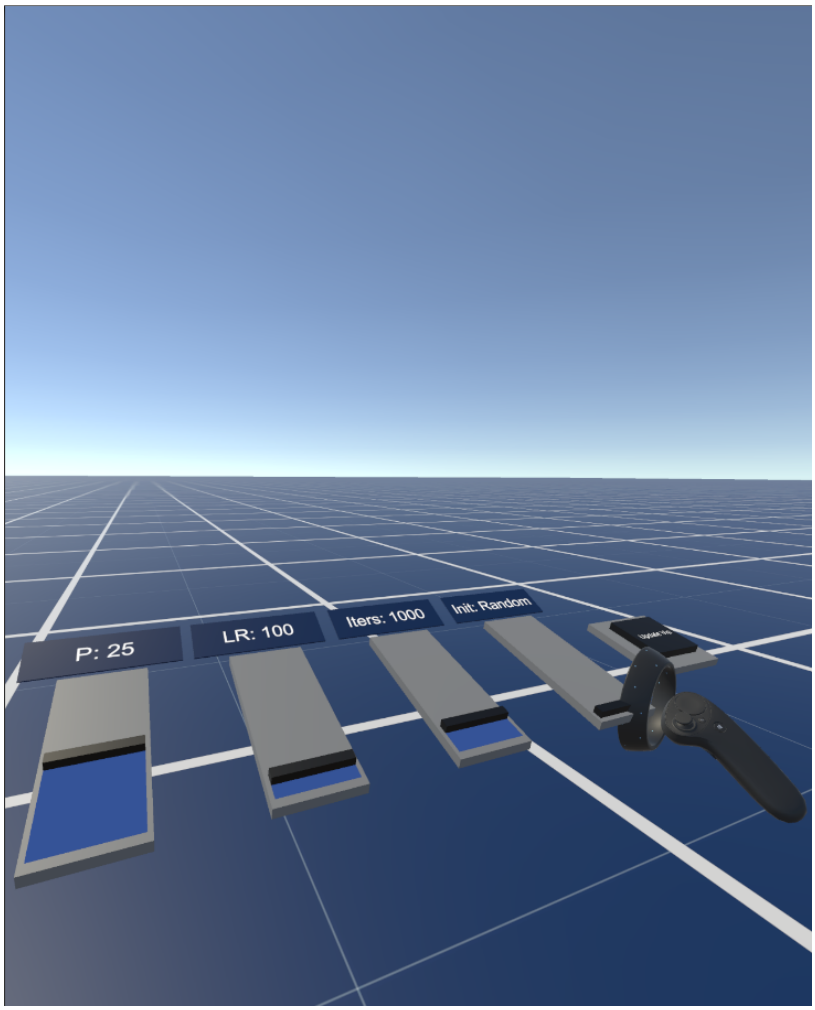


Figure 6.6: The t-SNE visualization parameter control interface with OSC-XR sliders to control T-SNE parameters and an OSC-XR pad to trigger visualization refresh with the new parameters.

do not directly manipulate the visualization in real-time. Using this panel users are able to adjust the t-SNE parameters and rerun the data transformation on a Python server. I also take advantage of OSC-XR to interact with visualization objects from a distance. Every mark in the visualization is configured as an OSC Pointer Trigger providing users the ability to interact with marks using the pointer from an input controller. Using this interaction technique a user is able to select any mark in the visualization to playback its associated audio file allowing users to explore the data aurally, as well as visually. Lastly, visualization marks can be filtered using the pointer to select any genre mark from the legend. Using OSC-XR con-

trollers I have been able to quickly prototype new methods for exploring and interacting with an immersive visualization.

While Unity, DxR, and OSC-XR are all used to build the immersive environment, other applications are needed to support it. t-SNE is implemented in Python and audio playback is implemented in ChucK. OSC communication affords us the ability to easily communicate between the distributed applications. In addition, OSC-XR also allows for communication within Unity by attaching OSC receiver methods to Unity `GameObjects` affording flexible and extensible event handling. By using OSC-XR, I are able to rapidly prototype an immersive environment with complex needs, such as toolkit integration and distributed communication.

6.5 Conclusion

In this chapter, I introduced OSC-XR, a toolkit for prototyping MusE-XR. By providing developers readily available controllers and scripts enabled with OSC, OSC-XR reduces the need to build control objects from scratch, making the development of immersive environments more accessible to researchers and developers. Combined with the power of Unity for building 3D environments, developers using OSC-XR are able to easily explore the affordances of immersive XR environments to find interactions for music control that would not be possible with other mediums.

The flexibility of OSC-XR creates many opportunities to further research on immersive music environments. First, I plan to implement features to spawn any controller prefab from within an immersive environment. This provides sound designers, with and without Unity development experience, the ability to build and customize immersive performance environments on the fly. Furthermore, to allow designers to take full advantage of the large amounts of data potentially created by such an environment OSC-XR would benefit from gesture learning capabilities, similar to those of the Wekinator Fiebrink et al. (2009). Adding these features to OSC-XR will expand the possibilities of immersive performance environments and make designing them more accessible.

Chapter 7

Conclusion

The previous chapters in this thesis presented novel systems and knowledge on the design space of MusE-XR through the application of XR for enhanced musical learning and performance. In this chapter, I summarize this research and discuss its implications in terms of the research goals and questions defined in Section 1.3. I close with thoughts on future research for advancing the state of MusE-XR.

7.1 Discussion

Composed of technologies that afford virtual simulation and augmentation of the real world, XR has the potential to provide technologically extended human experiences. Unfortunately, XR has yet to garner interest with the general population aside from its novelty. Jerald (2016) argues that this is because there is a lack of applications that promote its potential benefits, B_1, B_2, B_3, B_4, and B_5. In this dissertation I look to the field of Sound and Music Computing (SMC) to identify musical applications for XR that engage these benefits. The research presented in this dissertation promotes XR by enabling experiences

- that encourage mental exercise (B_4) by increasing engagement in musical learning and performance,
- that make lives easier (B_3) by making musical learning more accessible,
- and, that are not possible in the real world (B_1) by taking advantage of the affordances and capabilities of XR.

Moreover, by developing new XR musical experiences, I aim to promote musical engagement through enhanced musical experiences and accessibility.

To this end, I devised two research goals, see Section 1.1, to better understand the challenges and design considerations for applying XR to SMC in the fields of musical learning and performance. This led to five research questions for supporting my research goals. In the next sections, I will revisit the research presented in this thesis as it applies to my research goals.

7.1.1 XREMIL

The first goal of this thesis is to learn the technical challenges and design needs to develop practical CAMIT systems. To achieve this goal, I perform two projects aimed at addressing RQ_1, regarding the technical challenges of CAMIT implementation, and RQ_2, regarding the design needs of CAMIT environments.

To address RQ_1, regarding the development of a computer assisted musical instrument tutoring (CAMIT) system of analyzing piano playing technique, I present an applied research project to develop a CAMIT system that explores the implementation of a system for automatically assessing pianist hand posture using accessible sensor technologies, see Chapter 4. To ensure a practical system that is accessible outside of a laboratory setting, I use commodity depth cameras for capturing the pianists hands. Using a depth camera I acquire a real world data set of beginning piano student playing practice exercise. I then apply existing CV and ML techniques to segment hands from the depth maps and identify the hand posture using the depth data. The outcome of this experiment demonstrates positive results in the development of student specific detection models even with a limited amount of data.

There are, however, some remaining challenges. Namely, the lack of a generalized model for hand posture detection. The large variation of physical hand characteristics of piano students, such as size, shape and color, as well as playing styles make training a generalized model challenging. Creating a general model requires a large data set containing a wide range of hands and playing styles that would be expensive to acquire and annotate. Moreover, without a generalized model, developing student specific models requires input from piano teachers to train them. Since piano teachers are not ML experts, an interface usable by non experts should be made available for the training process. Even with these challenges, a practical CAMIT system for automatic assessment of pianist hand posture is possible with a intuitively designed interface.

To address RQ_2, regarding the effectiveness of extended reality enhanced musical instrument learning (XREMIL) environments, I present results of a basic research study to evaluate the effectiveness of a XREMIL with real-time visual feedback. The effectiveness of an XREMIL environment with RTVF for learning notes on the theremin is evaluated by performing a user study with thirty participants. Results of the study indicate that participants tend to focus on correcting musical errors using the visual feedback rather than the aural feedback. Focusing the visual rather than the aural may have had negative affects on the learning process. While the results may not have been statistically significant, overall the participants showed

a smaller improvement in abilities after training with an environment with visual cues compared with an environment without visual cues. Participants indicated, however, that the XREMIL environment increased their confidence and engagement in the learning process. This may lead to more effective practice sessions over the longer term as users become accustomed to the learning environment and cognitive demand lowers.

7.1.2 Design Considerations

The second main goal of this research is to provide an understanding of design considerations for the integration of XR with SMC. To achieve this goal, I perform research aimed at addressing RQ_3, RQ_4, and RQ_5, regarding the design factors for developing MusE-XR. RQ_5 was a byproduct of RQ_4 that aimed to make it easier to explore the design space of VEME through rapid prototyping. I address this question, by developing an open source toolkit, OSC-XR, to make VEME design more accessible to sound designers interested in the XR space but with limited experience with XR development (see Chapter 6). Through my experiences in the design of VEME using OSC-XR and the evaluation of XREMIL, I have observed the following affordances and identified some preliminary design guidelines of MusE-XR, answering RQ_3 and RQ_4.

Affordances

Building on the original affordances of XR, described in Section 1.2, I have observed the following set of affordances for XR as applied to musical applications.

A_{M_1} *Spatially oriented feedback and guidance.*

By taking advantage of A_{XR_1} and A_{XR_2}, visual cues for performance feedback and guidance can be spatially oriented in the VE.

A_{M_2} *Augmentation of physical instruments and objects with virtual overlays.*

Extending the previous affordance and taking advantage of A_{XR_3}, musical instruments (physical or digital) can be extended with new capabilities to enhance musical learning and performance experiences;

additionally, everyday physical objects can be extended with musical capabilities for new forms of musical expression.

A_{M_3} *Realistic musical interaction.*

By taking advantage of A_{XR_4}, XR enables enhanced experiences from musical interactions that users are already familiar with.

A_{M_4} *Nonrealistic musical interactions*

By taking advantage of A_{XR_5}, XR enables musical interactions that are impossible in the real world.

A_{M_5} *Customizable musical spaces.*

By taking advantage of A_{XR_1}, A_{XR_2}, and A_{XR_2}, XR enables users to easily create and customize musical spaces to suit the needs of a specific performance or learning requirement.

A_{M_6} *Environments with multiple interaction zones.*

By extending A_{M_6} and taking advantage of A_{XR_1} and A_{XR_2}, XR enables the design of virtual environments with any number of virtual performance spaces each customized for a specific musical interaction; additionally, by taking advantage of A_{XR_5} users can instantly navigate between each performance environment.

Design Guidelines

Through the experience of designing MusE-XR throughout this thesis, I have identified preliminary guidelines for designing effective and engaging musical experiences. Here I present important guidelines identified during my research and their implications toward MusE-XR design. (See Section 5.2.3 for more design guidelines including those not directly experienced in my research.)

Because MusE-XR design is an emerging field with limited previous research, it should be noted that these are *preliminary* design guidelines based only on the limited existing literature and my experience from research conducted in this dissertation. Therefore, the proposed design guidelines are subject to a few limitations. Namely, the guidelines are based on

only two research studies, a single user study on an XREMIL environment for a single instrument with a small number of participants and an experiment in VEME design with only one designer (myself). The guidelines are subject to evolve with continued research on MusE-XR as more designers gain experience in the research space.

G_1 *Visual cues should not prevent a user from focusing on aural feedback.*

This may be considered one of the most significant takeaways from this research. Learning music is a cognitively challenging process and adding a visual layer to the learning process only increases the cognitive demands. The results of my user study on XREMIL demonstrate that learning environments which encourage students to concentrate on visual feedback may have negative effects on the learning process. The same goes for musical performance, too much visual noise may distract the performers' aural focus. Therefore, designers should limit the amount of concurrent visual feedback to reduce cognitive load during performance. Additionally, the visual cues presented should be carefully designed so that they do not distract the performer or require high levels of concentration during musical performance and training.

G_2 *Musical experiences should be oriented toward the AR end of the RV Continuum unless immersion is part of the experience.*

Affordances of MusE-XR enable experiences that integrate the real world with the virtual. Designers of VEME and XREMIL experiences should consider this during environment design and attempt to make the environment as close to full AR as possible. In regard to XREMIL, AR experiences allow students to interact directly with the instrument they are learning, affording feedback via the natural haptic and tactile responses of the instrument. Additionally, compared with fully immersive VR environments employing AR for VEME allows performers to more easily connect with other musicians as well as their audience during performance. Furthermore, AR affords designer opportunities to design experiences that use the real world to their advantage, such as in the cases of XR hyperinstruments or adding musical capabilities to non-musical physical objects. Sometimes, however, immersion

can be part of the experience the designer is going for. In this case, immersive VR experiences are warranted.

G_3 *Use spatially oriented visual cues to overlay objects, including instruments, for guidance and feedback.*

XR affordances enable designers capabilities to virtually overlay visual objects on the real world. As is common in HCI, visual signifiers improve usablity through enhanced discoverability of features. With XR it is now easy to add this information directly to physical objects. Additionally, visual signifiers afford enhanced methods for user feedback; and with XR visual feedback can be overlaid directly on the objects it relates to. Spatially oriented visual cues in XREMIL environments afford new methods to provide guidance on correct musical performance during a lesson while allowing the student keep their focus on the music instrument. Additionally, spatially oriented visual cues allow for feedback to be placed directly where the errors occur. In regard to VEME, spatially oriented cues can provide users with information about how a musical object is to be used and feedback about the state of an object that is hard to discern from the aural output alone.

7.2 Future Work

MusE-XR is a developing field with the potential for compelling XR experiences that engage the general population. Throughout this thesis I present novel systems and knowledge on the MusE-XR design space to facilitate research towards these experiences. In this section, I discuss some directions to further the field and enable the design of innovative MusE-XR.

Through the development of a new system for automated assessment of playing technique as well as through an evaluation of the effects of RTVF on the learning process, I demonstrate new methods for enhanced musical learning. An evident extension of this work is to combine the assessment system with the design insights gained through the user study to develop a new XREMIL for piano tutoring. The development of such a system would require further research regarding both an interface for training assessment system and the environment for presenting RTVF. Regarding the as-

sessment system, research is needed to further improve model training. Currently, the system requires training student specific detection models which may be challenging for students and teachers not trained in ML. It may be possible to develop a generalized model with a large enough dataset, but individualized models afford customization to learn different error postures specific to each student that a generalized may not be able to handle. Teachers, however, should not be required to be ML experts to train the systems. Addressing this concern requires further research on the development of interfaces and techniques for simplified ML training, such as with Interactive Machine Learning (Amershi et al., 2014; Holzinger, 2016; Simard et al., 2017; Chen et al., 2018), Active Learning (Settles, 2009) and semi automatic approaches (Oberweger et al., 2016). Research applying HCI methodologies to with these techniques has potential to democratize ML. By developing new training interfaces, such research opens the door apply ML to new problems that generalized ML models may not be able to handle. Further, it will help address the cold start problem of ML especially when data annotation requires an expert.

Another extension to this work would be to experiment with other sensor technologies and improved hand segmentation approaches to obtain assessment data. Since the work discussed in Chapter 4 was completed, hand tracking and segmentation approaches have improved and been integrated within the SDKs of tracking technologies, such as the Leap Motion (2019). The latest Leap Motion SDK was also designed for XR experience making it an interesting sensor for further research on hand posture detection. Integrating the Leap Motion into the posture detection pipeline affords the use of a well-tested hand and finger tracking model, but still requires evaluation for an implementation in which hands are interacting with the piano. Additionally, Dalmazzo and Ramirez (2019) demonstrate that a Myo armband, a sensor that uses EMG and IMU data for motion tracking, works for the classification of bowing gestures in violin performance. Employing such as device for hand posture assessment may replace the need for a depth camera and mitigate some of the complications associated with CV. Using this technology, however, would limit the amount of hand location information available to a tutoring system.

Further research is also needed to develop an effective interface for pro-

viding students with the feedback from the performance assessment. I envision an XREMIL environment that integrates AR with a hand posture assessment system to display RTVF with visual cues providing direction to correct posture mistakes. Results of the user study and research from Chapter 5 provide valuable insights for designing such an interface, but more research is need to improve and understand the effects of XREMIL on musical learning transfer. Specifically, a longitudinal study over 5 or 6 training sessions with a more diverse user population is needed to better understand if the added cognitive demand of RTVF decreases over time as students become accustomed to the environment; and if so, does learning transfer improve? Further, additional research is needed to better understand what type of visual cues work best; is it effective to provide visual indicators instructing the student on how to correct their mistakes or is simply indicating that a mistake occurred sufficient? Research towards addressing these questions and others is important to advance and facilitate the development of XREMIL and other XR training environments.

Answering these questions will enable the design of new XREMIL for more traditional instruments. The lessons learned from the user study presented Chapter 5 are most applicable to fretless stringed instruments, such as the violin, cello and double bass. Because the fingerboards of these instruments do not have frets indicating note stops, musicians must learn to play and find notes by ear, similar to the theremin. Using XR to enhance the learning process, a designer may decide to simply augment the fingerboard of the instrument with visual cues indicating the location of given notes, but lessons learned through my user study suggest this may not be the best design for learning. By directly guiding the student to the exact location, the student may become overly reliant on the visualization and may not learn to properly listen to the tones around a given note. Instead, designers may want to provide visual guidance after the student makes too many incorrect attempts at the note. Alternatively, instead of directly guiding the student to the correct location, the designer may want to provide some form of positive RTVF only when the student performs the correct note location. Thus, encouraging experimentation to find the note while increasing student confidence by providing validation that the note is being performed correctly. These are only a few examples of considerations a

design may want to make when building XREMIL for fretless instruments, but they show the importance of the design choices needed to enhance the music learning process.

The emergence of XREMIL combined with new CAMIT systems for the automatic assessment of musical performance opens the door to new opportunities for the gamification of musical instrument learning. Gamification applied to education is concept of using game-based elements and mechanics to promote learning through increased engagement and motivation (Buckley and Doyle, 2016). Music games, such as Guitar Hero (Activision) and Rock Band (Harmonix, a), have been around for over a decade, and more recently music games have also entered XR with Rock Band VR (Harmonix, b). Successful as these games are, they mainly focus on entertainment rather than music learning. Integrating systems for the automatic assessment of musical performance with the previously identified affordances of MusE-XR provides researchers and designers with new possibilities to create immersive musical games that focus on musical learning through gamification. Automatic assessment technologies afford mechanisms for scoring users' musical performance, and the visual component of XR enables the design of visually engaging environments that builds on success of music games. Gamification, however, has shown mixed results in learning transfer and student motivation (Buckley and Doyle, 2016; Hanus and Fox, 2015) so researchers should be careful to evaluate gamified XREMIL.

This thesis has also covered research to facilitate the design and development of musical performance environments, or virtual environments for musical expression (VEME). OSC-XR paves the way for further research on a wide range of musical experiences with VEME, as demonstrated in the uses cases from Chapter 6. OSC-XR makes it easier for designers to build VEME, but it requires that they have some knowledge of Unity development. To support sound and instrument designers with limited or no development experience, OSC-XR should be extended so that designers are able to build musical performance spaces from directly within the VE. For example, by providing users in the VE with a palette of prebuilt virtual controller objects to choose from and place anywhere in the environment. It would also be interesting to allow users to create their own musical performance gestures from within in the VE. The amount of data made available with OSC-XR

enables gesture recognition and other machine learning techniques affording customized gestural mappings for musical control. Integrating OSC-XR with a system such as the Wekinator (Fiebrink et al., 2009) would enable users to develop there own models for mapping OSC-XR data to sound engine control. Using the Wekinator, however, would require a user to constantly switch back and forth between the VEME and the computer display to fine tune their models. Therefore, a new system that integrates directly in the VE is needed. OSC-XR is a novel toolkit to enable further research on VEME design by making the development workflow more accessible. Integrating the ideas discussed will take this a step further by enabling the creation of new immersive music control systems for users with no development training.

Although Lanier saw the potential for musical experiences in extended reality (MusE-XR) almost 30 years ago, there has since been limited research expanding the idea. This thesis provides an introductory study on its design space, and more research and experience are needed to develop compelling environments for engaging musical experiences. To that end, I hope the work presented in this thesis enables the advancement of MusE-XR by facilitating future research.

Appendix A

Publications

A.1 Publications from this Research

This section presents publications directly related to research performed for this thesis.

[1] D. Johnson, D. Damian, and G. Tzanetakis. Evaluating the Effectiveness of Mixed Reality Music Instrument Learning with the Theremin. *Virtual Reality*, July 2019.

[2] D. Johnson, D. Damian, and G. Tzanetakis. OSC-XR: A Toolkit for Extended Reality Immersive Music Interfaces. In *Proceedings of the 2019 Sound and Music Computing Conference*, May 2019.

[3] D. Johnson, D. Damian, and G. Tzanetakis. Detecting Hand Posture in Piano Playing Using Depth Data. *Computer Music Journal*, To Appear 2019.

[4] D. Johnson, I. Dufour, G. Tzanetakis, and D. Damian. Detecting Pianist Hand Posture Mistakes for Virtual Piano Tutoring. In *Proceedings of the International Computer Music Conference*, pages 168-171, 2016.

[5] D. Johnson, and G. Tzanetakis. VRmin: Using Mixed Reality to Augment the Theremin for Musical Tutoring. In *Proceedings of the 2017 Conference on New Interfaces for Musical Expression*, pages 151-156, 2017.

A.2 Publications not from this Research

[1] B. Manaris, D. Johnson, and M. Rourk. Diving into Infinity: A Motion-Based, Immersive Interface for MC Escher's Works. In *Proceedings of the 21st International Symposium on Electronic Art*, 2015.

[2] D. Johnson and G. Tzanetakis. Guitar model recognition from single instrument audio recordings. In *Proceedings of the 2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 370-375, 2015.

Appendix B

Publically Available Software

[1] UnityOscLib: A Simple Open Sound Control (OSC) library for Unity. https://github.com/fortjohnson/UnityOscLib, 2019.

[2] OSC-XR: A Toolkit for Extended Reality (XR) Immersive Music Interfaces. https://github.com/fortjohnson/OSC-XR, 2019.

[3] MRemin: A Virtually Augmented Theremin for Enhanced Learning. https://github.com/fortjohnson/MRemin, 2018.

Publically Available Software Not Part of this Research

[4] MATOS: Multi Agent Tangible Object Software. https://github.com/fortjohnson/MATOS, 2019.

Bibliography

ABRSM. Making Music: Teaching, learning, and playing in the UK. https://ca.abrsm.org/en/making-music/, 2014. Online; Accessed: 2019-04-06.

Activision. Guitar hero. https://www.guitarhero.com/ca/en/. Online; Accessed: 2019-07-17.

S. V. Adamovich, G. G. Fluet, E. Tunik, and A. S. Merians. Sensorimotor training in virtual reality: a review. *NeuroRehabilitation*, 25(1):29–44, 2009.

S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4): 105–120, 2014.

J. L. Aróstegui. Exploring the global decline of music education. *Arts Education Policy Review*, 117(2):96–103, apr 2016.

J. Atherton and G. Wang. Chunity: Integrated Audiovisual Programming in Unity. In *Proceedings of the 2018 Conference on New Interfaces for Musical Expression*, 2018.

M. Beaudouin-Lafon. Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 446–453, New York, NY, USA, 2000. ACM. ISBN 1-58113-216-6. doi: 10.1145/332040.332473. URL http://doi.acm.org/10.1145/332040.332473.

Berklee. Online piano and keyboard certificates and courses, 2019. URL https://online.berklee.edu/piano-and-keyboard.

D. F. Berthaut, M. Desainte-Catherine, and M. Hachet. Interacting with 3d reactive widgets for musical performance. *Journal of New Music Research*, 40(3):253–263, 2011.

F. Berthaut, C. Arslan, and L. Grisoni. Revgest: Augmenting gestural musical instruments with revealed virtual objects. In *International Conference on New Interfaces for Musical Expression*, 2017.

A. D. Blanco and R. Ramirez. Evaluation of a Sound Quality Visual Feedback System for Bow Learning Technique in Violin Beginners: An EEG Study. *Frontiers in psychology*, 10:165, 2019. ISSN 1664-1078.

S. Borsci, G. Lawson, B. Jha, M. Burges, and D. Salanitri. Effectiveness of a multidevice 3D virtual environment application to train car service maintenance procedures. *Virtual Reality*, 20(1):41–55, mar 2016.

N. Böttcher, S. Gelineck, L. Martinussen, and S. Serafin. Virtual reality instruments capable of changing physical dimensions in real-time. *Proceeding of Enactive*, 2005.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565.

P. Brinkmann, C. Mccormick, P. Kirn, M. Roth, and R. Lawler. Embedding Pure Data with libpd. In *Proceeding of the Fourth International Pure Data Convention*, pages 291–301, 2011.

P. Buckley and E. Doyle. Gamification and student motivation. *Interactive Learning Environments*, 24(6):1162–1175, 2016. doi: 10.1080/10494820.2014.964263.

I. Bukvic and S. Lee. Glasstra: Exploring the Use of an Inconspicuous Head Mounted Display in a Live Technology-Mediated Music Performance. In *Proceedings of the 2017 Conference on New Interfaces for Musical Expression*, pages 313–318, 2017.

E. M. Burnam. *A Dozen a Day Preparatory Book*. Hal Leonard Corporation, 2005.

J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, Nov 1986.

M. Carfagni, R. Furferi, L. Governi, M. Servi, F. Uccheddu, and Y. Volpe. On the performance of the intel sr300 depth camera: Metrological and critical characterization. *IEEE Sensors Journal*, 17(14):4508–4519, July 2017. ISSN 1530-437X. doi: 10.1109/JSEN.2017.2703829.

P. Carlson, A. Peters, S. B. Gilbert, J. M. Vance, and A. Luse. Virtual training: Learning transfer of assembly tasks. *IEEE Trans Vis Comput Graph*, 21(6):770–782, June 2015.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002. ISSN 1076-9757.

N.-C. Chen, J. Suh, J. Verwey, G. Ramos, S. Drucker, and P. Simard. Anchorviz: Facilitating classifier error discovery through interactive semantic data exploration. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces*, pages 269–280, March 2018.

J. Chow, H. Feng, R. Amor, and B. C. Wünsche. Music education using augmented reality with a head mounted display. In *Proceedings of the Fourteenth Australasian User Interface Conference*, AUIC '13, pages 73–79, Darlinghurst, Australia, Australia, 2013. Australian Computer Society, Inc.

D. A. Cook, S. J. Hamstra, R. Brydges, B. Zendejas, J. H. Szostek, A. T. Wang, P. J. Erwin, and R. Hatala. Comparative effectiveness of instructional design features in simulation-based education: Systematic review and meta-analysis. *Medical Teacher*, 35(1):e867–e898, 2013. doi: 10.3109/0142159X.2012.714886.

P. Cook. Principles for designing computer music controllers. In *Proceedings of the 2001 Conference on New Interfaces for Musical Expression*, NIME '01, pages 1–4, Singapore, Singapore, 2001. National University of Singapore. URL `http://dl.acm.org/citation.cfm?id=1085152.1085154`.

P. R. Cook. *Real Sound Synthesis for Interactive Applications*. A. K. Peters, Ltd., Natick, MA, USA, 2002. ISBN 1568811683.

P. R. Cook. Re-designing principles for computer music controllers: a case study of squeezevox maggie. In *Proceedings of the 2009 Conference on New Interfaces for Musical Expression*, volume 9, pages 218–221, 2009.

J. Copeland and J. Long. Restoring the first recording of computer music. http://blogs.bl.uk/sound-and-vision/2016/09/restoring-the-first-recording-of-computer-music.html, 2016. Online; Accessed: 2016-12-03.

S. M. Cormier and J. D. Hagman. *Transfer of learning: Contemporary research and applications*. Academic Press, 1987.

M. Csikszentmihalyi, S. Abuhamdeh, and J. Nakamura. *Flow*, pages 227–238. Springer Netherlands, Dordrecht, 2014a. ISBN 978-94-017-9088-8. doi: 10.1007/978-94-017-9088-8_15. URL https://doi.org/10.1007/978-94-017-9088-8_15.

M. Csikszentmihalyi, R. Graef, and S. M. Gianinno. *Measuring Intrinsic Motivation in Everyday Life*, pages 113–125. Springer Netherlands, Dordrecht, 2014b. ISBN 978-94-017-9088-8. doi: 10.1007/978-94-017-9088-8_8. URL https://doi.org/10.1007/978-94-017-9088-8_8.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

B. Dalgarno and M. J. Lee. What are the learning affordances of 3-D virtual environments? *Br Jour Educ Technol*, 41(1):10–32, jan 2010.

D. C. Dalmazzo and R. Ramirez. Bowing Gestures Classification in Violin Performance: A Machine Learning Approach. *Frontiers in Psychology*, 10: 344, mar 2019.

R. B. Dannenberg, M. Sanchez, A. Joseph, P. Capell, R. Joseph, and R. Saul. A computer based multimedia tutor for beginning piano students. *Interface*, 19(2-3):155–173, 1990.

R. B. Dannenberg, M. Sanchez, A. Joseph, R. Joseph, R. Saul, and P. Capell. Results from the piano tutor project. In *Proceedings of the Fourth Biennial Arts and Technology Symposium*, pages 143–150, 1993.

S. Das, S. Glickman, F. Y. Hsiao, and B. Lee. Music Everywhere - Augmented Reality Piano Improvisation Learning System. *Proceedings of the 2017 International Conference on New Interfaces for Musical Expression*, pages 511 – 512, 2017.

J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, volume 00, pages 248–255, June 2018.

B. Di Donato, J. Bullock, and A. Tanaka. Myo mapper: a myo armband to osc mapper. In *Proceedings of the Conference on New Interfaces for Musical Expression*, 2018.

D. Diakopoulos and A. Kapur. Argos: An Open Source Application for Building Multi-Touch Musical Interfaces. In *Proceedings of the 2010 International Computer Music Conference*, pages 88–91, 2010.

O. J. Dunn. Multiple comparisons using rank sums. *Technometrics*, 6(3): 241–252, 1964.

A. Elliott, B. Peiris, and C. Parnin. Virtual Reality in Software Engineering: Affordances, Applications, and Challenges. In *Proceedings of the International Conference on Software Engineering*, volume 2, pages 547–550, 2015.

K. A. Ericsson, R. T. Krampe, and C. Tesch-Römer. The role of deliberate practice in the acquisition of expert performance. *Psychological Review*, 100(3):363–406, 1993. ISSN 1939-1471. doi: 10.1037/0033-295X.100.3.363. URL `http://doi.apa.org/getdoi.cfm?doi=10.1037/0033-295X.100.3.363`.

P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587597.

S. Ferguson. Learning musical instrument skills through interactive sonification. In *Proc. of the 2006 Conference on New Interfaces for Musical Expression*, pages 384–389, Paris, France, 2006. IRCAM; Centre Pompidou. ISBN 2-84426-314-3.

R. Fiebrink, D. Trueman, and P. R. Cook. A meta-instrument for interactive, on-the-fly machine learning. In *Proceedings of the 2009 Conference on New Interfaces for Musical Expression*, pages 280–285, 2009.

J. Fillwalk. Chromachord: A virtual musical instrument. In *2015 IEEE Symposium on 3D User Interfaces*, pages 201–202. IEEE, 2015.

Y. Fukuya, Y. Takegawa, and H. Yanagi. A piano learning support system considering motivation. In *Proceedings of the 2013 International Computer Music Conference*, pages 62–68, 2013.

J. L. Gabbard. *A taxonomy of usability characteristics in virtual environments*. PhD thesis, Virginia Polytechnic Institute and State University, 1997.

J. L. Gabbard, D. Hix, and J. E. Swan. User-centered design and evaluation of virtual environments. *IEEE Computer Graphics and Applications*, 19(6): 51–59, Nov 1999.

J. Garcia. UnityOSC. `https://github.com/jorgegarcia/UnityOSC`, 2019. Online; Accessed: 2019-04-16.

J. Gibson. *The ecological approach to visual perception:*. Houghton Mifflin, 1979. ISBN 9780395270493.

S. Giraldo, G. Waddell, I. Nou, A. Ortega, O. Mayor, A. Perez, A. Williamon, and R. Ramirez. Automatic Assessment of Tone Quality in Violin Music Performance. *Frontiers in Psychology*, 10:334, mar 2019.

M. Gonzalez-Franco, R. Pizarro, J. Cermeron, K. Li, J. Thorn, W. Hutabarat, A. Tiwari, and P. Bermell-Garcia. Immersive mixed reality for manufacturing training. *Frontiers Robot AI*, 4:3, 2017.

Google. Google daydream mobile virtual reality. `https://www.moogmusic.com/products/etherwave-theremins/theremini`, 2019. Online; Accessed: 2019-05-06.

A. Hadjakos. Pianist motion capture with the kinect depth camera. In *Proc. of the Int. Conference on Sound and Music Computing, Copenhagen, Denmark*, 2012.

A. Hadjakos, F. Lefebvre-Albaret, and I. Toulouse. Three methods for pianist hand assignment. In *6th Sound and Music Computing Conference*, pages 321–326, 2009.

R. Hamilton. UDKOSC: An immersive musical environment. In *Proceedings of the 2011 International Computer Music Conference*, number August, pages 717–720, 2011.

R. Hamilton, J. P. Caceres, C. Nanou, and C. Platz. Multi-modal musical environments for mixed-reality performance. *Journal on Multimodal User Interfaces*, 4(3-4):147–156, dec 2011.

H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In D.-S. Huang, X.-P. Zhang, and G.-B. Huang, editors, *Advances in Intelligent Computing*, pages 878–887, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31902-3.

M. D. Hanus and J. Fox. Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Computers & Education*, 80:152 – 161, 2015. ISSN 0360-1315. doi: https://doi.org/10.1016/j.compedu.2014.08.019.

Harmonix. Rock band. `https://www.rockband4.com/`, a. Online; Accessed: 2019-07-17.

Harmonix. Rock band vr. https://www.rockbandvr.com/, b. Online; Accessed: 2019-07-17.

S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, 2006.

R. Hatala, D. A. Cook, B. Zendejas, S. J. Hamstra, and R. Brydges. Feedback for simulation-based procedural skills training: a meta-analysis and critical narrative synthesis. *Advances in Health Sciences Education*, 19(2):251–272, 2014.

H. He, Y. Bai, E. A. Garcia, and S. Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, June 2008. doi: 10.1109/IJCNN.2008.4633969.

Hexler. touchOSC: Modular OSC and MIDI control surface for iPhone / iPod Touch / iPad, 2019. URL https://hexler.net/software/touchosc.

A. Holzinger. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131, June 2016. ISSN 2198-4026.

F. Huang, Y. Zhou, Y. Yu, Z. Wang, and S. Du. Piano ar: A markerless augmented reality based piano teaching system. In *2011 International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 47–52, Aug 2011.

A. Hunt, M. M. Wanderley, and M. Paradis. The Importance of Parameter Mapping in Electronic Instrument Design. *Journal of New Music Research*, 32(4):429–440, dec 2003.

ImageNet. Imagenet summary and statistics. http://image-net.org/about-stats, 2010. Online; Accessed: 2019-05-01.

J. Jaime, I. Barbancho, C. Urdiales, L. J. Tardón, and A. M. Barbancho. A new multiformat rhythm game for music tutoring. *Multimedia Tools and Applications*, 75(8):4349–4362, apr 2016.

J. Jerald. *The VR Book: Human-Centered Design for Virtual Reality*. Association for Computing Machinery and Morgan & Claypool, New York, NY, USA, 2016.

N. W. John, N. I. Phillips, L. a. Cenydd, S. R. Pop, D. Coope, I. Kamaly-Asl, C. de Souza, and S. J. Watt. The use of stereoscopy in a neurosurgery training virtual environment. *Presence: Teleoper Virtual Environ*, 25(4): 289–298, 2016.

N. W. John, S. R. Pop, T. W. Day, P. D. Ritsos, and C. J. Headleand. The implementation and validation of a virtual environment for training powered wheelchair manoeuvres. *IEEE Trans Vis Comput Graph*, 24(5):1867–1878, 2018.

D. Johnson and G. Tzanetakis. Guitar model recognition from single instrument audio recordings. In *Proceedings of the 2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 370–375, Aug 2015.

D. Johnson and G. Tzanetakis. VRMin: Using Mixed Reality to Augment the Theremin for Musical Tutoring. In *Proceedings of the 2017 Conference on New Interfaces for Musical Expression*, 2017.

D. Johnson, B. Manaris, Y. Vassilandonakis, and S. Stoudenmier. Kuatro: A motion-based framework for interactive music installations. In *Proceedings of the International Computer Music Conference*, 2014.

B. Kang, K.-H. Tan, H.-S. Tai, D. Tretter, and T. Q. Nguyen. Hand segmentation for hand-object interaction from depth map. *arXiv preprint arXiv:1603.02345*, 2016.

C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*, chapter Real Time Hand Pose Estimation Using Depth Sensors, pages 119–137. Springer London, London, 2013.

T. Kitamura and M. Miura. Constructing a support system for self-learning playing the piano at the beginning stage. In *Proceedings of International*

Conference on Music Perception and Cognition, pages 258–262. Citeseer, 2006.

M. J. Kostka. Practice Expectations and Attitudes: A Survey of College-Level Music Teachers and Students. *Journal of Research in Music Education*, 50(2):145–154, jul 2002. ISSN 0022-4294. doi: 10.2307/3345818.

M. J. Kostka. Teach Them How to Practice. *Music Educators Journal*, 90 (5):23–26, 2004.

J. J. Kozak, P. A. Hancock, E. J. Arthur, and S. T. Chrysler. Transfer of training from virtual reality. *Ergon*, 36(7):777–784, 1993.

J. Kratus. Centennial series: Music education at the tipping point. *Music Educators Journal*, 94(2):46–48, 2007.

Y. Kweon, S. Kim, B. Yoon, T. Jo, and C. Park. Implementation of educational drum contents using mixed reality and virtual reality. In C. Stephanidis, editor, *HCI International 2018 – Posters' Extended Abstracts*, pages 296–303, Cham, 2018. Springer International Publishing.

K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE International Conference on Robotics and Automation*, pages 1817–1824, May 2011. doi: 10.1109/ICRA.2011.5980382.

J. Lanier. Virtual instrumentation, 1992. URL `http://www.jaronlanier.com/instruments.html`.

J. Lanier. *Dawn of the New Everything: Encounters with Reality and Virtual Reality*. Henry Holt and Co., 2017.

Leap Motion. Leap Motion. `https://www.leapmotion.com/`, 2019. Online; Accessed: 2019-04-27.

H. Leeuw. The electrumpet , a hybrid electro-acoustic instrument. In *Proc. of the 2009 Conference on NIME*, 2009.

K. S. Lehmann, J. P. Ritz, H. Maass, H. K. Çakmak, U. G. Kuehnapfel, C. T. Germer, G. Bretthauer, and H. J. Buhr. A prospective randomized

study to test the transfer of basic psychomotor skills from virtual reality to physical reality in a comparable training setting. *Ann Surg*, 241(3): 442, 2005.

D. Lewis. Listen to the first computer-made tune on alan turing's synthesizer. https://www.smithsonianmag.com/smart-news/listen-first-recording-alan-turing-playing-tune-synthesizer-180960586/, Sep 2016. Online; Accessed: 2019-04-16.

M. Li, P. Savvidou, B. Willis, and M. Skubic. Using the kinect to detect potentially harmful hand postures in pianists. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual Int. Conference of the IEEE*, pages 762–765, Aug 2014.

H. Liang, J. Yuan, and D. Thalmann. Parsing the hand in depth images. *IEEE Transactions on Multimedia*, 16(5):1241–1253, Aug 2014. ISSN 1520-9210. doi: 10.1109/TMM.2014.2306177.

H. Liang, J. Wang, Q. Sun, Y.-J. Liu, J. Yuan, J. Luo, and Y. He. Barehanded music: Real-time hand interaction for virtual piano. In *Proc. of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 87–94, NY, USA, 2016. ACM. ISBN 978-1-4503-4043-4.

F. Liarokapis. Augmented Reality Scenarios for Guitar Learning. In L. M. Lever and M. McDerby, editors, *EG UK Theory and Practice of Computer Graphics*. The Eurographics Association, 2005.

C.-C. Lin and D. S.-M. Liu. An intelligent virtual piano tutor. In *Proceedings of the 2006 ACM International Conference on Virtual Reality Continuum and Its Applications*, VRCIA '06, pages 353–356, New York, NY, USA, 2006. ACM. ISBN 1-59593-324-7. doi: 10.1145/1128923.1128986. URL http://doi.acm.org/10.1145/1128923.1128986.

H. Lu, B. Zhang, Y. Wang, and W. K. Leow. iDVT: An Interactive Digital Violin Tutoring System Based on Audio-Visual Fusion. In *Proceedings of the 16th ACM International Conference on Multimedia*, page 1005, New York, New York, USA, 2008. ACM Press.

T. Machover. Hyperinstruments: A Progress Report. Technical report, Massachusets Institute of Technology, 01 1992.

T. Machover and J. T. Chung. Hyperinstruments: Musically intelligent and interactive performance and creativity systems. In *Proceedings of the 1989 International Computer Music Conference*, 1989.

J. MacRitchie and A. P. McPherson. Integrating optical finger motion tracking with surface touch events. *Frontiers in Psychology*, 6:702, jun 2015.

T. Mäki-Patola, J. Laitinen, A. Kanerva, and T. Takala. Experiments with virtual reality instruments. In *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, pages 11–16, Singapore, Singapore, 2005. National University of Singapore.

K. Marriott, F. Schreiber, T. Dwyer, K. Klein, N. H. Riche, T. Itoh, W. Stuerzlinger, and B. H. Thomas, editors. *Immersive Analytics: An Introduction*, pages 1–23. Springer International Publishing, Cham, 2018.

Microsoft. Windows Mixed Reality, 2019. URL `https://www.microsoft.com/en-ca/windows/windows-mixed-reality`.

Microsoft. Azure Kinect Development Kit. https://azure.microsoft.com/en-ca/services/kinect-dk/, 2019a. Online; Accessed: 2019-04-27.

Microsoft. Windows Mixed Reality Toolkit. `https://github.com/Microsoft/MixedRealityToolkit-Unity`, 2019b. Online; Accessed: 2019-05-08.

H. C. Miles, S. R. Pop, S. J. Watt, G. P. Lawrence, and N. W. John. A review of virtual environments for training in ball sports. *Computers & Graphics*, 36(6):714 – 726, 2012. ISSN 0097-8493. 2011 Joint Symposium on Computational Aesthetics (CAe), Non-Photorealistic Animation and Rendering (NPAR), and Sketch-Based Interfaces and Modeling (SBIM).

P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Photonics for industrial applications*, pages 282–292. International Society for Optics and Photonics, 1995.

E. R. Miranda and M. Wanderley. *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard (Computer Music and Digital Audio Series)*. A-R Editions, Inc., Madison, WI, USA, 2006. ISBN 089579585X.

Moog. Moog theremini. `https://www.moogmusic.com/products/etherwave-theremins/theremini`, 2019. Online; Accessed: 2019-05-06.

A. G. Moore, M. J. Howell, A. W. Stiles, N. S. Herrera, and R. P. McMahan. Wedge: A musical interface for building and playing composition-appropriate immersive environments. In *2015 IEEE Symposium on 3D User Interfaces*, 2015.

J. Mora, W. s. Lee, G. Comeau, S. Shirmohammadi, and A. E. Saddik. Assisted piano pedagogy through 3d visualization of piano playing. In *2006 IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, pages 157–160, 2006.

F. Morreale, A. De Angeli, and S. O'Modhrain. Musical interface design: An experience-oriented framework. In *NIME*, pages 467–472, 2014.

A. G. E. Mulder, S. S. Fels, and K. Mase. Design of virtual 3d instruments for musical interaction. In *Proceedings of the 1999 Conference on Graphics Interface*, pages 76–83, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

M. Murcia-Lopez and A. Steed. A comparison of virtual and physical training transfer of bimanual assembly tasks. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1574–1583, April 2018.

K. Ng, P. Nesi, and V. S. Marta. i-Maestro: Technology-Enhanced Learning and Teaching for Music. In *Proceedings of the 2008 Conference on New Interfaces for Musical Expression*, 2008.

K. C. Ng, T. Weyde, T. Koerselman, B. Ong, K. Neubarth, and O. Larkin. 3D Augmented Mirror: A Multimodal Interface for String Instrument Learning and Teaching with Gesture Support. In *Proceedings of the Ninth International Conference on Multimodal Interfaces*, page 339, New York, New York, USA, 2007. ACM Press.

H. M. Nguyen, E. W. Cooper, and K. Kamei. Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 3(1):4–21, 2011.

J. Nielsen. 10 usability heuristics for user interface design. `https://www.nngroup.com/articles/ten-usability-heuristics/`, 1994. Accessed: 2017-01-28.

J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 249–256, New York, NY, USA, 1990. ACM. ISBN 0-201-50932-6.

D. A. Norman. *The Design of Everyday Things*. Basic Books, Inc., New York, NY, USA, 2002. ISBN 9780465067107.

M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit. Efficiently creating 3d training data for fine hand pose estimation. In *Proceeeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Oculus. Oculus Touch Controllers. `https://developer.oculus.com/documentation/pcsdk/latest/concepts/dg-input-touch-overview/#input-hands-and-controller-basics`, 2019. Online; Accessed: 2019-04-27.

A. Oka and M. Hashimoto. Marker-less piano fingering recognition using sequential depth images. In *Frontiers of Computer Vision, (FCV), 2013 19th Korea-Japan Joint Workshop on*, pages 1–4, Jan 2013.

M. Oren, P. Carlson, S. Gilbert, and J. M. Vance. Puzzle assembly training: Real world vs. virtual environment. In *2012 IEEE Virtual Reality Workshops (VRW)*, pages 27–30, March 2012.

Y. Orlarey, D. Fober, and S. Letz. Faust: an efficient functional approach to dsp programming. *New Computational Paradigms for Computer Music*, 290:14, 2009.

C. Oshima, K. Nishimoto, and N. Hagita. A piano duo support system for parents to lead children to practice musical performances. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 3(2): 9–es, may 2007. ISSN 15516857. doi: 10.1145/1230812.1230815. URL http://portal.acm.org/citation.cfm?doid=1230812.1230815.

D. Overholt. The overtone violin. In *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, pages 34–37. National University of Singapore, 2005.

J. A. Paradiso and N. Gershenfeld. Musical applications of electric field sensing. *Computer music journal*, 21(2):69–89, 1997.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. of Machine Learning Research*, 12: 2825–2830, 2011.

G. Percival, Y. Wang, and G. Tzanetakis. Effective use of multimedia for computer-assisted musical instrument tutoring. In *Proceedings of the International Workshop on Educational Multimedia and Multimedia Education*, pages 67–76, NY, USA, 2007. ACM.

A. Perez-Carrillo. Violin Timbre Navigator: Real-Time Visual Feedback of Violin Bowing Based on Audio Analysis and Machine Learning. In *Lecture Notes in Computer Science*, volume 11296 LNCS, pages 182–193. Springer, Cham, jan 2019. ISBN 9783030057152. URL http://link.springer.com/10.1007/978-3-030-05716-9{_}15.

C. Pettey. 3 reasons why vr and ar are slow to take off. https://www.gartner.com/smarterwithgartner/3-reasons-why-vr-and-ar-are-slow-to-take-off/, Sep 2018. Online; Accessed: 2019-05-12.

J. Preece, Y. Rogers, and H. Sharp. *Interaction Design*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2002. ISBN 0471492787.

R. Ramìrez, A. Perez, and G. Volpe. Technology Enhanced Learning of Musical Instrument Performance. http://telmi.upf.edu/, 2019. Online; Accessed: 2019-03-13.

S. Raptis, A. Chalamandaris, A. Baxevanis, A. Askenfelt, E. Schoonderwaldt, K. F. Hansen, D. Fober, S. Letz, and Y. Orlarey. Imutus - an effective practicing environment for music tuition. In *Proceedings of the International Computer Music Conference*, 2005.

J. Rasmussen. *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*. Elsevier Science Inc., New York, NY, USA, 1986. ISBN 0444009876.

K. Riley, E. E. Coons, and D. Marcarian. The use of multimodal feedback in retraining complex technical skills of piano performance. *Medical Problems of Performing Artists*, 20(2):82–88, 2005.

C. Roberts. Control: Software for end-user interface programming and interactive performance. In *Proceedings of the 2011 International Computer Music Conference*, pages 425–428, 2011.

S. Rogers. How vr, ar and mr are making a positive impact on enterprise. https://www.forbes.com/sites/solrogers/2019/05/09/how-vr-ar-and-mr-are-making-a-positive-impact-on-enterprise/#63df61b85253, May 2019. Online; Accessed: 2019-05-10.

Y. Rogers, H. Sharp, and J. Preece. *Interaction Design: Beyond Human - Computer Interaction*. Wiley Publishing, 4th edition, 2015. ISBN 9781119020752.

F. D. Rose, E. A. Attree, B. M. Brooks, D. M. Parslow, and P. R. Penn. Training in virtual environments: transfer to real world tasks and equivalence to real task training. *Ergon*, 43(4):494–511, 2000.

L. Russolo. Art of noise: A futurist manifesto. 1913.

A. Salgian and D. Vickerman. Computer-based tutoring for conducting students. In *Proceedings of the International Computer Music Conference*, 2016.

E. Schoonderwaldt, A. Askenfelt, and K. F. Hansen. Design and implementation of automatic evaluation of recorder performance in imutus. In *Proceedings of the International Computer Music Conference*, pages 97–103, 2005.

M. Schrepp, A. Hinderks, and J. Thomaschewski. Applying the user experience questionnaire (ueq) in different evaluation scenarios. In *International Conference of Design, User Experience, and Usability*, pages 383–392. Springer, 2014.

S. Serafin, C. Erkut, J. Kojs, N. C. Nilsson, and R. Nordahl. Virtual reality musical instruments: State of the art, design principles, and future directions. *Computer Music Journal*, 41(2), 2016.

B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

B. Shneiderman. *The New ABCs of Research: Achieving Breakthrough Collaborations*. Oxford University Press, 2016.

J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304, June 2011. doi: 10.1109/CVPR.2011.5995316.

R. Sicat, J. Li, J. Choi, M. Cordeil, W. Jeong, B. Bach, and H. Pfister. Dxr: A toolkit for building immersive data visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):715–725, Jan 2019.

S. Siegel and N. J. Castellan Jr. *Nonparametric statistics for the behavioral sciences, 2nd ed.* Mcgraw-Hill Book Company, New York, NY, England, 1988. ISBN 0-07-057357-3 (Hardcover).

P. Simard, S. Amershi, M. Chickering, A. Edelman Pelton, S. Ghorashi, C. Meek, G. Ramos, J. Suh, J. Verwey, M. Wang, and J. Wernsing. Machine teaching: A new paradigm for building machine learning systems. Technical report, July 2017.

Skoove. Skoove, 2019. URL `https://www.skoove.com/`.

M. Slater and M. V. Sanchez-Vives. Enhancing Our Lives with Immersive Virtual Reality. *Frontiers in Robotics and AI*, 3(December):1–47, 2016.

J. O. Smith. *Physical audio signal processing: For virtual musical instruments and audio effects*. W3K Publishing, 2010.

L. Spinello and K. O. Arras. People detection in rgb-d data. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3838–3843, Sept 2011. doi: 10.1109/IROS.2011.6095074.

I. E. Sutherland. The ultimate display. In *Proceedings of the IFIP Congress*, pages 506–508, 1965.

I. E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I), pages 757–764, New York, NY, USA, 1968. ACM. doi: 10.1145/1476589.1476686. URL `http://doi.acm.org/10.1145/1476589.1476686`.

S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao. Histogram of oriented normal vectors for object recognition with a depth sensor. In *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision*, pages 525–538. Springer, Berlin, Heidelberg, 2012.

A. S. S. Thomsen, D. Bach-Holm, H. KjÃęrbo, K. HÃÿjgaard-Olsen, Y. Subhi, G. M. Saleh, Y. S. Park, M. la Cour, and L. Konge. Operating room performance improves after proficiency-based virtual reality cataract surgery training. *Ophthalmol*, 124(4):524 – 531, 2017. ISSN 0161-6420.

M. Tits, J. Tilmanne, N. d'Alessandro, and M. M. Wanderley. Feature extraction and expertise analysis of pianists' motion-captured finger gestures. In *Proc. of the 2015 Int. Computer Music Conference*, pages 102–105, Denton, 2015.

J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graph.*, 33(5):169:1–169:10, Sept. 2014. ISSN 0730-0301.

S. Trail, M. Dean, G. Odowichuck, T. F. Tavares, P. F. Driessen, W. A. Schloss, and G. Tzanetakis. Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect. In *Proceedings of the 2012 Conference on New Interfaces for Musical Expression*, 2012.

G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002. ISSN 1063-6676. doi: 10.1109/TSA.2002.800560.

Udemy. Piano courses, 2019. URL https://www.udemy.com/topic/piano/.

Unity. Unity. https://www.unity3d.com, 2019. Online; Accessed: 2019-05-08.

Unreal. Unreal Engine. https://www.unrealengine.com/en-US/, 2019a. Online; Accessed: 2019-05-13.

Unreal. New Audio Engine: Early Access Quick-Start Guide. https://forums.unrealengine.com/development-discussion/audio/116874-new-audio-engine-early-access-quick-start-guide, 2019b. Online; Accessed: 2019-05-13.

L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 2008.

D. Waller, E. Hunt, and D. Knapp. The transfer of spatial knowledge in virtual environment training. *Presence: Teleoper Virtual Environ*, 7(2): 129–143, 1998.

M. M. Wanderley and N. Orio. Evaluation of input devices for musical expression: Borrowing tools from hci. *Computer Music Journal*, 26(3): 62–76, 2002. ISSN 01489267, 15315169. URL http://www.jstor.org/stable/3681979.

M. M. Wanderley, J. Malloch, J. Garcia, W. E. Mackay, M. Beaudouin-Lafon, and S. Huot. Human Computer Interaction meets Computer Music: The MIDWAY Project, May 2016. URL https://hal.inria.fr/hal-01370588. CHI'16 - Music and HCI Workshop, Extended Abstracts on Human Factors in Computing Systems. San Jose, United States.

S. Weinschenk and D. T. Barker. *Designing Effective Speech Interfaces*. John Wiley & Sons, Inc., New York, NY, USA, 2000. ISBN 0-471-37545-4.

S. Werrlich, P.-A. Nguyen, and G. Notni. Evaluating the training transfer of head-mounted display based training for assembly tasks. In *Proceedings of the 11th PErvasive Technologies Related to Assistive Environments Conference*, PETRA '18, pages 297–302, New York, NY, USA, 2018. ACM.

D. Wessel and M. Wright. Problems and Prospects for Intimate Musical Control of Computers. *Computer Music Journal*, 26(3):11–22, 2002. doi: 10.1162/014892602320582945.

D. Wessel, M. Wright, and J. Schott. Intimate musical control of computers with a variety of controllers and gesture mapping metaphors. In *Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–3. National University of Singapore, 2002.

B. G. Witmer and M. J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoper Virtual Environ*, 7(3):225–240, 1998.

M. Wright. Open Sound Control: An enabling technology for musical networking. *Organised Sound*, 10(3):193–200, 2005.

M. Wright and A. Freed. Open SoundControl: A New Protocol for Communicating with Sound Synthesizers. In *Proceedings of the International Computer Music Conference*, 1997.

X. Xiao and H. Ishii. Inspect, Embody, Invent. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pages 5397–5408, New York, New York, USA, 2016. ACM Press. ISBN 9781450333627. doi: 10.1145/2858036.2858577. URL `http://dl.acm.org/citation.cfm?doid=2858036.2858577`.

X. Xiao, B. Tome, and H. Ishii. Andante: Walking figures on the piano keyboard to visualize musical motion. In *Proceedings of the 2014 Conference on New Interfaces for Musical Expression*, pages 629–632, 2014.

D. Young. The hyperbow controller: Real-time dynamics measurement of violin performance. In *Proceedings of the 2002 Conference on New Interfaces for Musical Expression*, 2002.

Yousician. Yousician, 2019. URL `https://yousician.com/`.

D. Zielasko, D. Rausch, Y. C. Law, T. C. Knott, S. Pick, S. Porsche, J. Herber, J. Hummel, and T. W. Kuhlen. Cirque des bouteilles: The art of blowing on bottles. In *IEEE Symposium on 3D User Interfaces (3DUI)*, pages 209–210, March 2015.

Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773 – 780, 2006. ISSN 0167-8655.