

**L'ARTE DI INTERAZIONE MUSICALE:
NEW MUSICAL POSSIBILITIES THROUGH
MULTIMODAL TECHNIQUES**

JORDAN NATAN HOCHENBAUM

A DISSERTATION
SUBMITTED TO THE
VICTORIA UNIVERSITY OF WELLINGTON AND
MASSEY UNIVERSITY IN FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN SONIC ARTS

NEW ZEALAND SCHOOL OF MUSIC

2013

Supervisory Committee

Dr. Ajay Kapur (New Zealand School of Music)

Supervisor

Dr. Dugal McKinnon (New Zealand School of Music)

Supervisor

© JORDAN N. HOCHENBAUM, 2013

NEW ZEALAND SCHOOL OF MUSIC

Abstract

Multimodal communication is an essential aspect of human perception, facilitating the ability to reason, deduce, and understand meaning. Utilizing multimodal senses, humans are able to relate to the world in many different contexts. This dissertation looks at surrounding issues of multimodal communication as it pertains to human-computer interaction. If humans rely on multimodality to interact with the world, how can multimodality benefit the ways in which humans interface with computers? Can multimodality be used to help the machine understand more about the person operating it and what associations derive from this type of communication?

This research places multimodality within the domain of musical performance, a creative field rich with nuanced physical and emotive aspects. This dissertation asks, what kinds of new sonic collaborations between musicians and computers are possible through the use of multimodal techniques? Are there specific performance areas where multimodal analysis and machine learning can benefit training musicians? In similar ways can multimodal interaction or analysis support new forms of creative processes?

Applying multimodal techniques to music-computer interaction is a burgeoning effort. As such the scope of the research is to lay a foundation of multimodal techniques for the future. In doing so the first work presented is a software system for capturing synchronous multimodal data streams from nearly any musical instrument, interface, or sensor system.

This dissertation also presents a variety of multimodal analysis scenarios for machine learning. This includes automatic performer recognition for both string and drum instrument players, to demonstrate the significance of multimodal musical analysis. Training the computer to recognize who is playing an instrument suggests important information is contained not only within the acoustic output of a performance, but also in the physical domain. Machine learning is also used to perform automatic drum-stroke identification; training the computer to recognize which hand a drummer uses to strike a drum. There are many applications for drum-stroke identification including more detailed

automatic transcription, interactive training (e.g. computer-assisted rudiment practice), and enabling efficient analysis of drum performance for metrics tracking.

Furthermore, this research also presents the use of multimodal techniques in the context of everyday practice. A practicing musician played a sensor-augmented instrument and recorded his practice over an extended period of time, realizing a corpus of metrics and visualizations from his performance. Additional multimodal metrics are discussed in the research, and demonstrate new types of performance statistics obtainable from a multimodal approach.

The primary contributions of this work include (1) a new software tool enabling musicians, researchers, and educators to easily capture multimodal information from nearly any musical instrument or sensor system; (2) investigating multimodal machine learning for automatic performer recognition of both string players and percussionists; (3) multimodal machine learning for automatic drum-stroke identification; (4a) applying multimodal techniques to musical pedagogy and training scenarios; (4b) investigating novel multimodal metrics; (5) lastly this research investigates the possibilities, affordances, and design considerations of multimodal musicianship both in the acoustic domain, as well as in other musical interface scenarios. This work provides a foundation from which engaging musical-computer interactions can occur in the future, benefitting from the unique nuances of multimodal techniques.

Contents

Chapter 1 Introduction	1
1.1 NOISE AND INSPIRATION.....	1
1.2 ON HUMAN INTERACTION.....	2
1.3 A DEFINITION OF MULTIMODALITY	4
1.4 OVERVIEW	8
1.5 SUMMARY OF CONTRIBUTIONS.....	10
Related Work	12
Chapter 2 Background and Motivation	13
2.1 A BRIEF HISTORY OF MULTIMODALITY AND HCI	13
2.1.1 Detecting Affective States	15
2.1.2 Selected Examples of Multimodal Musical Systems.....	16
2.1.3 Summary.....	19
2.2 A HISTORY OF RELATED PHYSICAL COMPUTING.....	19
2.2.1 New Interfaces & Controllers: Building On and Diverging From Existing Metaphors	20
2.2.2 Hyperinstruments	23
2.3 TOWARDS MACHINE MUSICIANSHIP.....	26
2.3.1 Rhythm Detection	26
2.3.2 Pitch Detection	28
2.4 MUSIC AND MACHINE LEARNING	29
2.4.1 Supervised Learning and Modeling Complex Relationships in Music	30
2.5 SUMMARY.....	32
Research and Implementation	34
Chapter 3 The Toolbox	35
3.1 INSTRUMENTS, INTERFACES, AND SENSOR SYSTEMS	35
3.1.1 Esitar.....	36
3.1.2 Ezither	37
3.1.3 Esuling.....	38
3.1.4 XXL	39
3.2 NUANCE: A SOFTWARE TOOL FOR CAPTURING SYNCHRONOUS DATA STREAMS FROM MULTIMODAL MUSICAL SYSTEMS	40

3.2.1	Introduction to Nuance.....	40
3.2.2	Background and Motivation	42
3.2.3	Architecture and Implementation.....	44
3.2.4	Workflow	50
3.2.5	Summary	51
Chapter 4 Performer Recognition		53
4.1	BACKGROUND AND MOTIVATION.....	53
4.2	PROCESS	55
4.3	SITAR PERFORMER RECOGNITION.....	55
4.3.1	Data Collection	56
4.3.2	Feature Extraction.....	58
4.3.3	Windowing.....	59
4.3.4	Classification.....	59
4.3.5	Results and Discussion	60
4.4	DRUM PERFORMER RECOGNITION.....	64
4.4.1	Data Collection	65
4.4.2	Feature Extraction.....	67
4.4.3	Understanding Data Through Multimodal Visual Feature Clustering.....	68
4.4.4	Classification.....	70
4.4.5	Results and Discussion	71
4.5	DISCUSSION.....	76
Chapter 5 Drum-Stroke Computing.....		79
5.1	BACKGROUND AND MOTIVATION.....	79
5.2	DATA COLLECTION	82
5.3	ANALYSIS FRAMEWORK	82
5.3.1	Surrogate Data Training	83
5.3.2	Onset Detection.....	83
5.3.3	Feature Extraction	85
5.4	DRUM HAND RECOGNITION	86
5.4.1	Classification.....	86
5.4.2	Results: About the Tests.....	86
5.4.3	Results: Test One – All Data (Individual vs. Combined Scores).....	86
5.4.4	Results: Test Two – Data Split.....	89

5.4.5 Results: Test Three – Leave One (performer) Out	90
5.5 DRUM PERFORMANCE METRICS	91
5.5.1 Cross-modal Onset Difference Time (ODT).....	92
5.6 DISCUSSION.....	97
Chapter 6 Multimodal Onset Detection	99
6.1 ON MUSIC AND ONSETS	99
6.2 AUDIO VS. SENSOR ONSET DETECTION: STRENGTHS AND WEAKNESSES	102
6.3 SYSTEM DESIGN AND IMPLEMENTATION	104
6.3.1 Onset Detection Function.....	105
6.3.2 Fusion Algorithm.....	106
6.3.3 Data Collection	107
6.4 ONSET DETECTION AND FUSION RESULTS.....	107
6.4.1 Discussion: Audio-Only Onset Detection Results	108
6.4.2 Discussion: Sensor-Only Onset Detection Results	109
6.4.3 Discussion: Multimodal Onset Fusion Results	110
6.4.4 Discussion: Precision, Recall, and F_1 -Measure.....	111
6.5 MUSICAL CONTEXTS AND CONCLUSIONS	114
Chapter 7 Rethinking How We Learn: Performance Metrics and Multimodality in the Practice Room	117
7.1 BACKGROUND AND MOTIVATION	117
7.2 OVERVIEW OF METRICS EXPERIMENTS	120
7.3 DATA COLLECTION	120
7.3.1 Ezither Data	120
7.4 TEMPO METRICS AND STATISTICS	121
7.4.1 Tempo Estimation Algorithm	121
7.4.2 Tempo: Performance Timing	122
7.4.3 Tempo: Evolution of Timing over a Performance.....	123
7.5 BOW ARTICULATION TECHNIQUE METRICS AND STATISTICS	128
7.5.1 Definition of Bow Articulations.....	128
7.5.2 Bow Articulation: Tempo Accuracy	128
7.5.3 Bow Articulation: Onset Difference Time (ODT).....	135
7.5.4 Bow Articulation: Articulation Attack Slope.....	137
7.6 LONG-TERM METRICS ANALYSIS.....	139
7.6.1 Long-Term Tempo Metrics: Average.....	140

7.6.2 Long-Term Tempo Metrics: Standard Deviation.....	140
7.6.3 Long-Term Tempo Metrics: Range	144
7.6.4 Long-Term Bow Articulation Metrics: Tempo Accuracy	146
7.6.5 Long-Term Bow Articulation Metrics: Onset Difference Time	148
7.6.6 Long-Term Bow Articulation Metrics: Articulation Attack Slope.....	150
7.7 SUMMARY	152
Chapter 8 Conclusion.....	155
8.1 SUMMARY	155
8.2 PRIMARY CONTRIBUTIONS	156
8.2.1 Enabling Multimodal Musical Analysis with Nuance	157
8.2.2 Teaching the Computer to Know Who You Are.....	157
8.2.3 Negotiating Novel Understandings and Interactions in Drum Performance	158
8.2.4 Advancing Machine Musicianship through Multimodal Fusion	158
8.2.5 Refining The Way Musicians Learn: Multimodal Performance Metrics and Musical Pedagogy.....	159
8.3 PRINCIPLES AND CONSIDERATIONS ON THE DESIGN OF MULTIMODAL MUSICAL INSTRUMENTS AND SENSOR SYSTEMS	160
8.3.1 What is the musical context?.....	160
8.3.2 Exploitation	161
8.3.3 Transparency	162
8.3.4 Applying multimodality <i>to</i> a musical task vs. applying multimodality <i>into</i> a musical task.....	162
8.3.5 On Continuous Controls and ‘Leaky Faucets’.....	163
8.4 MAPPING MULTIMODAL MUSICAL SYSTEMS	164
8.4.1 One-to-one, Complimentary modalities, and multi-dimensional control	165
8.4.2 Many-to-one as a space for multimodal integration.....	166
8.4.3 Defining a set of parameterizations	167
8.5 FUTURE WORK	167
8.6 CONCLUSION	168
Appendix	169
Appendix A Live Performances and Applications	171
A.1 MINIM PERFORMANCE AT THE NEW ZEALAND SCHOOL OF MUSIC SONIC ARTS EXHIBITION CONCERT, OCTOBER 9TH, 2010	171

A.1.1	Excitation, Impulse, and Probability Machines	172
A.1.2	Composing by Improvisation	173
A.1.3	Performer Interaction	174
A.2	III: PERFORMANCE AT THE NEW ZEALAND SCHOOL OF MUSIC SONIC ARTS EXHIBITION CONCERT, OCTOBER 9 TH , 2011	174
A.2.1	Hyperinstruments and Gesture	175
A.2.2	Composition, Improvisation, and Iteration.....	177
A.2.3	Performer-Interaction.....	179
A.3	TRANSFORMATIONS: INTEGRATING MULTIMODAL MUSIC, DANCE, VISUALS, AND WEARABLE TECHNOLOGY, MAY 11 – 17, 2012.....	181
A.3.1	Designing the System.....	181
A.3.2	Discussion.....	185
A.4	SMARTFIDUCIAL.....	186
A.4.1	Background and Motivation.....	187
A.4.2	Implementation.....	188
A.4.3	Hardware.....	188
A.4.4	Software.....	192
8.6.2	Discussion: Spatial Relationships and Tangible Interfaces	193
A.4.5	Discussion: New Affordances for Tabletop Interaction	194
A.4.6	Final Thoughts on Augmented Fiducial Objects.....	195
A.5	SUMMARY.....	196
Appendix B Sensors		199
B.1	TRANSDUCERS	199
B.2	PIEZOELECTRIC SENSORS	200
B.3	FORCE-SENSING RESISTORS.....	201
B.4	ACCELEROMETERS.....	202
Appendix C Communication Systems and Protocols.....		205
C.1	MIDI.....	205
C.2	OPEN SOUND CONTROL.....	206
C.3	TUIO	207
Appendix D Machine Learning		209
D.1	SUPERVISED LEARNING	209
D.2	K-FOLD CROSS-VALIDATION	211
D.3	ALGORITHMS	212

D.3.1 Decision Trees	212
D.3.2 Naive Bayes	213
D.3.3 k-Nearest Neighbor (kNN).....	214
D.3.4 Artificial Neural Networks (ANNs)	215
Appendix E Refereed Academic Journals and Publications	217
Bibliography.....	219

List of Figures

Figure 1: Example of the McGurk effect integrating /ga/ (visual) and /ba/ (auditory), results in the perceived /da/	3
Figure 2: Unimodal vs. Multimodal Musical Interfaces.....	6
Figure 3: Overview diagram of Complementary Modalities vs. Multimodal Fusion.....	7
Figure 4: Overview of Research.....	9
Figure 5: EMG biometric and gyro-based position controller (arm bands, headbands and base) used in (Tanaka and Knapp 2002)	17
Figure 6: Max Mathews and the Radio Baton (left) and the Buchla Lightning III (right)	21
Figure 7: Collaborative music making on the Reactable (left), and Bricktable “Roots” (right)	22
Figure 8: Final Hyperbow violin design by Diana Young.....	24
Figure 9: Esitar sensor systems, close up of thumb sensor (left), and usb, standard audio jack, knobs, buttons, and switches (right).....	36
Figure 10: Pictures of the Ezither hyperinstrument and bow	37
Figure 11: Picture of the Esuling controller showing the two FSRs and buttons.....	38
Figure 12: XXL sensor system (left) and screenshot of XXLSerial MIDI/OSC translator (right).....	39
Figure 13: Requirement comparison of other software and frameworks considered as of May 2012.....	43
Figure 14: Overview of Nuance input synchronization and output scheme	46
Figure 15: Audio Recorder object.....	47
Figure 16: Serial Message Format	47
Figure 17: Example Arduino serial out messages for two analog sensors.....	47
Figure 18: Example .xml configuration.....	48
Figure 19: Sensor (serial), OSC and MIDI Recorders	49
Figure 20: Nuance session main editor panel screenshot	50

Figure 21: Overview of the performer recognition system (only sitar shown in figure).....	55
Figure 22: Overview of data capturing and feature extraction.....	58
Figure 23: Audio vs. sensor vs. multimodal accuracy achieved for improv data set after training with Exercise and <i>Yaman</i> data sets	64
Figure 24: Nuance software (Left) and custom sensor system (Right).....	65
Figure 25: Overview of drum rudiments and paradiddles performed by all performers for drum performer recognition in 4.4 and drum stroke computing in Chapter 5.....	66
Figure 26: Overview of features extracted at each event in the data set	67
Figure 27: Feature scatter-plots of audio features <i>regularity</i> vs. <i>roughness</i> on the left and <i>regularity</i> vs. <i>spectral centroid</i> on the right	69
Figure 28: Feature scatter-plot of audio feature <i>spectral rolloff</i> vs. sensor feature <i>average (mean) release phase deceleration</i>	70
Figure 29: Confusion matrix for all data sets and sensor features only using the MLP classifier.....	73
Figure 30: Performer recognition accuracy for all classifiers using all features and all data sets D1-D4	74
Figure 31: Accuracy for audio-only features vs. sensor features vs. multimodal features by averaging all classifiers	75
Figure 32: Overview of drum hand recognition system.....	82
Figure 33: Overview of onset detection algorithm	84
Figure 34: Average drum-hand recognition accuracy (%) across all performers for each classifier	88
Figure 35: Average classification of all classifiers for each performer	88
Figure 36: Classification results for the two best performing players when trained on all <i>other</i> data sets	91
Figure 37: Onset difference times for the 60-sec. of D1 (performer one top, performer two bottom)	92
Figure 38: Bar graph visualizing average onset difference time metrics (Table 13 - rush, lag, mean, standard deviation, and range) for all ten performers, in seconds.....	93

Figure 39: Snare drum waveform (left) and envelope representation (right) of the note <i>onset</i> (circle), <i>attack</i> (bold) and <i>transient</i> (dashed). Figure adapted from (Bello et al. 2005)	100
Figure 40: Strengths and Weaknesses of Audio and Sensor Onset Detection ...	104
Figure 41: General Overview of Multimodal Onset Fusion.....	105
Figure 42: Onset Curve (envelope) and peak-picked onsets (circles) for a short window of audio	105
Figure 43: Onset fusion algorithm pseudo-code	106
Figure 44: Audio onsets detected over an excerpt of mostly tremolo playing, TP (black vertical lines), FN (grey rectangles).....	108
Figure 45: Sensor (accelerometer) onsets detected over an excerpt of mostly tremolo playing, TP (black vertical lines), FP (grey rectangles).....	110
Figure 46: Multimodal fusion onsets detected over an excerpt of mostly tremolo playing, TP (black vertical lines), FN (grey rectangles).....	111
Figure 47: Comparison of precision, recall, and F_1 -Measure for audio, sensor, and fusion onsets	113
Figure 48: Comparison of TP, FP, FN and #Bows for Audio, Sensor, and Fusion Onsets	115
Figure 49: Melody Repeated in Data Set 3 (D3).....	121
Figure 50: Tempo estimated for three recordings from Ezither recording #7, data set D2	122
Figure 51: Tempo evolution of Ezither recording #7, data set D2, (a) andante, (b) moderato, (c) allegro	124
Figure 52: Box and whisker plot for Ezither recording #4, data set D1 showing bowing statistics of three bow strokes (detaché, martelé, and spiccato) when playing at the target tempo of 120bpm.....	129
Figure 53: Tempo estimate and statistics (mean, standard deviation, and range) for detaché, martelé, and spicatto bow strokes for the Ezither recording #4 data set D1 (target tempo = 120 bpm).....	131
Figure 54: Onset difference time (ODT) statistics for recording #9 data set D1	137

Figure 55: Note attack slope for Ezither recording #9 data set D1, Detaché entire recording (top), 2.8 second window from 10sec – 12.8sec (bottom)	138
Figure 56: Mean and standard deviation of bow stroke attack slopes for Ezither recording #9 data set D1 detaché, martelé, and spiccato	139
Figure 57: Average (mean) tempo of each D2 tempo recording (andante – bottom, moderato (middle), allegro (top), from the entire data corpus 1-16	140
Figure 58: <i>Standard deviation</i> for every session data set D2 tempo (solid) and linear trend lines (dashed)	142
Figure 59: <i>Range</i> for every session data set D2 tempo (solid) and linear trend lines (dashed)	145
Figure 60: Standard deviation (A – top) and range (B – bottom) of bow articulation tempo across all D1 data sets collected.....	147
Figure 61: Session-to-session change in Ezither articulation Onset Difference Time.....	149
Figure 62: Ezither average articulation attack slope difference over time for (AAS difference – solid, trend lines dashed)	150
Figure 63: Ezither articulation attack slope standard deviation (top) and range (bottom) practice session-to-session difference over time for (AAS difference – solid, trend lines dashed).....	152
Figure 64: Curtis Bahn and the EDilruba (Sensor Esraj).....	176
Figure 65: Performance of <i>III</i> , group (left), and close-up of modified 12-string acoustic guitar and bow (right)	179
Figure 66: Overview of dance technology, XXL accelerometers on hands, and Microsoft Kinect for real-time projection mapping (masking) onto the dancer	185
Figure 67: SmartFiducial System Overview Diagram.....	188
Figure 68: SmartFiducial hardware design and layout	189
Figure 69: SmartFiducial TUIO Protocol Specification.....	191
Figure 70: Overview of the SmartFiducial Serial Protocol	191
Figure 71: SmartFiducial Prototype (buttons 1 & 2 not pictured).....	192
Figure 72: Two SmartFiducials being used with Turbine	193

Figure 73: Overview of common forms of energy that transducers convert200

Figure 74: Common FSR shapes and configurations (force a and b, position c)
.....201

Figure 75: Illustration of a typical supervised learning flow, adapted from
(Fiebrink 2011).....210

Figure 76: Sample decision tree constructed from the feature set of
instruments (for genre classification) listed in Table 25213

Figure 77: Illustration of kNN classifier where there are two classes (class 1
and class 2), and a rounded rectangle marked ‘?’ in the center is the
test or prediction point214

Figure 78: Simple artificial neural network with one hidden layer.....216

List of Tables

Table 1: Bol Patterns and Alankar exercises (data set 1)	57
Table 2: Accuracy achieved using audio only (15-second window)	60
Table 3: Accuracy achieved using sensors only (15-second window)	61
Table 4: Accuracy achieved using individual sensor features on all data sets, T=Thumb F=Fret (15-second window).....	62
Table 5: Accuracy achieved using multimodal data (15-second window)	62
Table 6: Identification accuracy of sensors vs. audio vs. multimodal fusion using a combined corpus from all data sets (at various window periods)	63
Table 7: Performer recognition accuracy using audio features (and ODT feature) only.....	72
Table 8: Performer recognition accuracy using sensor features only	72
Table 9: Performer recognition accuracy using all features (audio & sensors) combined	74
Table 10: Accelerometer onset detection accuracy for performers 1 and 2.....	85
Table 11: L/R Drum hand recognition accuracy for all performers and data.....	87
Table 12: Classification accuracy using separate rudiments for training and testing	90
Table 13: Average onset difference statistics for both performers.....	93
Table 14: Rush/Lag distribution of performer ODTs	96
Table 15: Distribution of onsets detected from audio-only as either True Positive, False Positive, or False Negative	109
Table 16: Distribution of onsets detected from sensor-only (accelerometer) as either True Positive, False Positive, or False Negative	110
Table 17: Distribution of onsets detected from multimodal onset fusion as True Positive, False Positive, or False Negative.....	111
Table 18: Comparison of precision, recall, and F_1 -Measure for audio, sensor, and fusion onsets.....	113

Table 19: Comparison of TP, FP, FN and #Bows for Audio, Sensor, and Fusion Onsets	115
Table 20: Tempo evolution statistics (min, max, mean, standard deviation, and range) of Ezither recording #7, data set D2, andante (80 bpm), moderato (110 bpm), and allegro (140 bpm)	123
Table 21: Five number summary for each bow stroke in Ezither recording #4, data set D1 (target tempo = 120 bpm).....	131
Table 22: One-way ANOVA multiple comparisons (Tukey HSD) for each bow stroke, Ezither recording #4 data set D1 (dependent variable = tempo).....	134
Table 23: Average Range of tempos from D2 for all data collected.....	145
Table 24: Tempo, range, and standard deviation averages over all practice sessions	146
Table 25: Sample feature set of instruments for genre classification problem using a decision tree classifier.....	212

Acknowledgements

This dissertation would not have been possible without the encouragement, direction, and inspiration of many people. First and foremost, I'd like to thank my primary advisor, Dr. Ajay Kapur. Ajay not only introduced me to the world of academia, and encouraged me to pursue postgraduate education, but he showed me that it is possible to simultaneously live in both academics and the arts. Among the many things I've learned from Ajay, he showed me it was possible to combine my passion for music and technology. His encouragement in developing my personality as a musician, engineer, and computer scientist, has been elemental over the course of my research. It goes without saying that Ajay has been and continues to be an inspirational source in my life, and I can't thank him enough.

Secondly, I owe endless thanks to my close colleague and friend, Owen Vallis. Working with Owen has been nothing short of amazing. His endless supply of brilliant ideas, creativity, and propensity to willingly dive head first into the deep-end and find his way out has always been a driving force, pushing me to achieve more. From peer coding to playing live music together as FlipMu, we're always on the same page, and Owen has been a hugely influential person in my life these past few years.

I'd also like to give many thanks to my second advisor Dr. Dugal McKinnon, and the New Zealand School of Music. Your encouragement has enabled me to achieve the work in this dissertation, your support has enabled me to focus on my research, and pursue the corners of my mind. Dugal, I am lucky to have received your perspective; from meetings over coffee, to your thought provoking posts on the Sonic Arts Facebook and email groups, I owe you many thanks. Additionally, I'd like to thank Victoria University and the Victoria Postgraduate Students Association, for supporting me with the Victoria PhD Scholarship, and the Postgraduate Research Excellence Award.

I also owe numerous thanks to the many individuals I've been lucky enough to have collaborated with over the last few years. To Dr. Matthew Wright, you have been highly influential, from your work on Open Sound Control, which has changed the way I interact with the computer musically, to collaborating with

you on sitar performer recognition in this dissertation. Your notes and guidance in what was one of my first research endeavors were crucial to my research approach, and was a springboard to the remainder of this dissertation. To Blake Johnston and Jason Erskine, it's been a pleasure working with you two and delving further into the world of hyperinstruments. You both brought your own voices to our practice sessions, our tech meetings, and uncovered incredible facets of composing and performing music with multimodal hyperinstruments. Blake, special thanks for bravely venturing into the world of Nuance, spending so much time working with the software, and letting me know when it worked, and when it didn't!

Lastly, it goes without saying that I owe the world and more to my family. Dad, Mom, Rami, Leyat, Natalie, and Nina—thank you for always supporting me, not only over the course of my PhD, but from day one. You have all inspired me in so many ways, and have made me who I am.

Chapter 1

Introduction

Motivation & Overview

“Futurist musicians should substitute for the limited variety of timbres that the orchestra possesses today the infinite variety of timbres in noises, reproduced with the appropriate mechanisms.”

—Luigi Russolo (Russolo 1986)

1.1 Noise and Inspiration

In the highly regarded manifesto, *L'Arte dei Rumori*¹ (Russolo 1986), Italian Futurist Luigi Russolo exalts in his 1913 letter to Futurist composer Francesco Balilla, the idea that novel mechanisms must be created in order to facilitate a new means of sonic expression. Russolo believed that humans had grown accustomed to the sounds of the matured industrial landscape, and that this mechanized urban environment presented an infinite spectrum of unheard sonorities and sounds—far surpassing the reproducibility of traditional instrumentation. Thus was born the *Art of Noises*, a manifesto in which Russolo first systematically describes a broad history of music; influenced by man's growing desire for an increasingly complex nature in sound tonalities, rhythm, and musical relationships. Russolo then discusses his belief that the future of music (at least as an attempt to convey truly “new” sonorities, rhythms, and emotion) was within the “noise-sound” of machines and nature. So much so that the only way to achieve these new sounds would be to create new instruments, mimicking these noise-sounds and learning to play and compose for them with great virtuosity.

¹ “*The Art of Noises*” translated from Italian to English.

This research does not attempt to fulfill *L'Arte dei Rumori's* goal of investigating the noise-sound, however, it derives the following from Russolo's fundamental beliefs; (1) there exists a growing desire to investigate novel relationships in sound phenomena, and (2) that new tools and methodologies are necessary to usher forth a new era of expressivity in musical performance and interaction. Specifically, this research investigates the use of novel *multimodal* techniques (a definition of multimodality is provided in section 1.3), and the possibilities when applied to the pedagogical aspects of a musician's practice, the learning environment in which a musician grows, and the ways in which a machine (computer) can affectively communicate and understand music and performance. In addition to the creation of new tools and methodologies, this research looks to principles emerging in other fields such as design, affective computing, and human-computer interaction (HCI), to investigate the implications and potential artistic freedoms gained from the research. Holistically, this dissertation explores novel multimodal technologies that enable new sonic engagements between musician and sound; an attempt to not only understand the intricacies of music and the nuance of a musician's technique, but to enrich the emotive qualities of musical interaction and experiences—*L'Arte di Interazione Musicale (The Art of Musical Interaction)*.

1.2 On Human Interaction

Everyday human interaction relies on our ability to deduce emotion and intent by simultaneously processing multiple channels of information from various sensory modalities (e.g. hearing, sight, touch, smell, taste). In even the simplest day-to-day interactions, our decisions and actions result from the evaluation of our beliefs in non-verbal (e.g. facial expressions, body gestures) and verbal (e.g. vocal tone/inflection, etc.) cues. A famous example that exploits this human multimodal integration is the McGurk effect (McGurk and MacDonald 1976). First published in 1976, the McGurk effect suggested the multimodal nature of speech perception by demonstrating an experiment where participants were

shown a video of an individual speaking one phoneme², while the audio was dubbed with another phoneme. Participants experienced a third intermediate phoneme being spoken, and the experiment proved the interdependency between hearing and vision in speech perception. Even when aware of the effect, the participant's perception often remained unchanged, further demonstrating the potency of human multimodal integration.

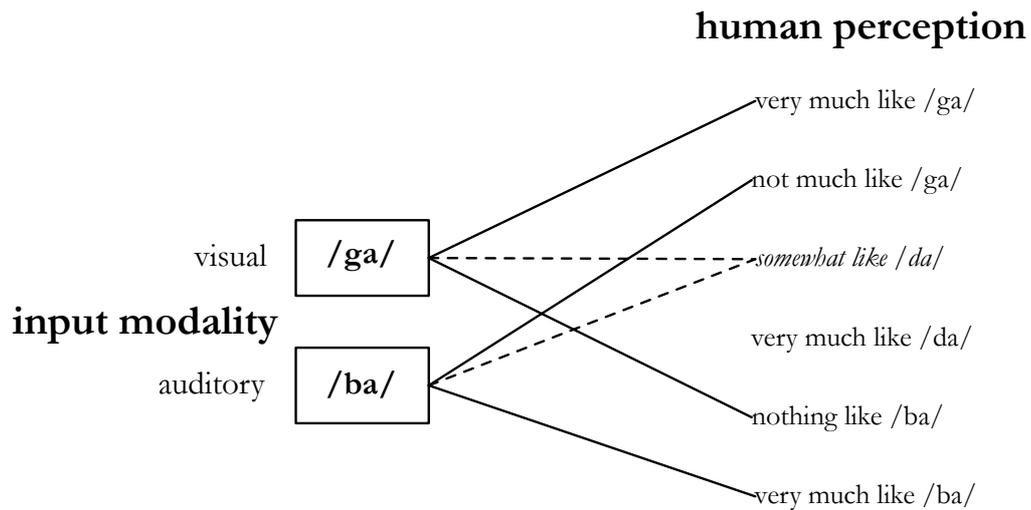


Figure 1: Example of the McGurk effect integrating /ga/ (visual) and /ba/ (auditory), results in the perceived /da/

The McGurk effect can be illustrated by pairing the visual /ga/ with the auditory /ba/; the viewer or listener often perceives the actual utterance as /da/. This has been explained with various justifications. McGurk and MacDonald believed that visible speech determines the perception of place of articulation whereas the audible speech determines the perception of voicing. The Perceptual Science Lab group at the University of California at Santa Cruz reasons the human brain's *multimodal fusion* mathematically using a fuzzy logic model of perception (Figure 1). Using fuzzy degrees of support, each perceived output is assigned a support value using multiplicative integration. In the example in Figure 1, *very much like* = 0.9; *somewhat like* = 0.7; *not much like* = 0.3; and *nothing like* = 0.1. One can see that /da/ would have almost twice as much support as the other options. *Support for ga* = $0.9 * 0.3 = 0.27$; *support for ba* =

² Phonemes are the smallest segmental unit of sound used to form different utterances in language.

$0.1 * 0.9 = 0.09$; *support for da* = $0.7 * 0.7 = \mathbf{0.49}$ (Perceptual Science Lab 2012).

The ability to process multimodal channels of information much like above has become an essential part of human cognition, communication, and survival. Humans and other living organisms also use multimodal integration to compensate for one sense with another, when the environment places constraints on a particular sense. For example, a person may rely more heavily on their ears and sense of touch as they slowly navigate a dark room. If the lights were on, they might rely more heavily on their sense of sight. This can be thought of as a somewhat Bayesian approach, which says that a degree of belief should rationally change when given new context or evidence (Bayes and Price 1763). This approach has been examined over the years in a number of disciplines however to date it has been largely underexplored in physical musical-computer interaction. Musical performance is rich in both physical and acoustic relationships, thus this research reasons that multimodality can be highly effective by offering the machine a more Bayesian vantage between the physical and acoustical aspects of musical performance. This is supported by recent applications of multimodal techniques in musical scenarios, and this research shows some of the unique affordances and possibilities of multimodal musical interaction. The remainder of this section describes the concept of multimodality in further detail, its history as part of the greater human-computer interaction field, and its relation to this research.

1.3 A Definition of Multimodality

In reviewing the published literature on multimodality (not only within music-related research but also within HCI, the cognitive sciences, and other related fields, see 2.1 for more history and related work) basic terms and concepts vary in definition and scope. Thus, it is important to first clarify a few key concepts and set up a taxonomy in which this research conforms. Firstly, a clarification of basic terms is presented, in accord with the work and definitions of Laurence Nigay and Joëlle Coutaz, early HCI pioneers in multimodal interaction (Nigay and Coutaz 1993):

1. **Modality:** The type of communication channel used to express and receive information, and to describe the interaction of communication.
2. **Mode:** The way/context in which the information is interpreted.

A modality defines the type of communication channel or data being exchanged, and the mode describes the context in which the data is interpreted. For example, the human auditory modality enables one to “hear”, while Bongers and Veer say the mode (that is expressed or interpreted) can be symbolic (verbal speech), iconic (non-speech), and expressive (non verbal, i.e. tone, etc.) (Bongers and Veer 2007). Bongers elaborates that in fact, human communication tends to use these modes at the same time, and that the modes are dependent on the context in which they are used.

Thus, for a system to be multimodal, the system must support the capacity to communicate with the user along these different (multi) channels (modalities and modes) of information simultaneously. Particularly as is the interest of this research, this is achieved by combining analysis of the acoustical output from an instrument/performer (auditory modality), with multi-sensory information obtained from various sensors measuring physical aspects of musical performance.

Furthermore, there are at least two distinct agents involved in multimodal interaction (the human and the machine), and multimodal interaction can be further reduced into a human-centered view and a system centered view (Schomaker et al. 1995). The human-centered view deals with perception and communication channels while the system-centered view focuses on the modes of computer input/output (Raisamo 1999). In general this research is in accordance with (Schomaker et al. 1995) in that although physically separated, a multimodal system is one that exchanges information through a number of communication channels between both agents.

As this research is primarily concerned with multimodal human input (into the system), for our purposes we define a unimodal system as a system that makes use of only one input modality whereas a multimodal system makes use of multiple input modalities. When making this distinction, it is important to note

that although a unimodal system has only one input modality, it can very make use of ‘n’ instances of a singular input modality (Figure 2).

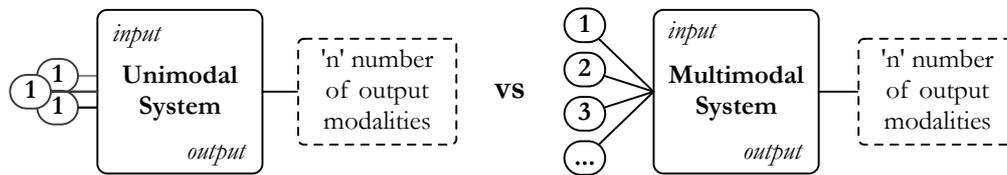


Figure 2: Unimodal vs. Multimodal Musical Interfaces

To further illustrate this, imagine you would like to perform gesture analysis on the performance of a dancer. One common approach in this type of scenario is to use optical or vision based tracking methods. Setting up one camera in front of the dance space might not be sufficient to capture the performance; perhaps there are objects (scene/props) involved in the piece that might occlude the dancer from the front of the space during certain movements, or the dancer might go outside of the cameras field-of-view. One obvious solution would be to position multiple cameras at different vantage points in the space, and to combine the information captured from all cameras. This is a unimodal example of having multiple input sources coming from a single input modality—and while it may provide similar goals and benefit as true multimodal input, fundamentally they are distinctly different approaches as we will see. By the definition conformed to in this research, multimodal systems require multiple (heterogeneous) communication channels between the agents.

It is also possible to have asymmetrical input and output modalities. This simply means that the multimodal system is not constrained to being output in the same modalities or communication channels of the input (and more generally to the same number of information channels). In this way, multimodal systems are also commonly feedback-based systems.

Additionally, two multimodal-related ideas that are elemental to this research are concepts of *complimentary modalities*, and *multimodal fusion*. Oviatt says that the “explicit goal [of multimodal interaction is] to integrate complementary modalities in a manner that yields a synergistic blend such that each mode can be capitalized upon and used to overcome weaknesses in the other mode” (Oviatt 2000). Lets take for a minute the example of the dancer described previously. The vision-based tracking system might be well suited for tracking the location

of the dancer within the space, as well as generalized movements and gesture, however, affordable vision-tracking systems often exhibit less-than-ideal camera resolution and frame rates (this is especially true in musical scenarios where response times of less than 20ms are often desired). It could be useful to compliment the system with other direct physical or biometric sensors. Although the direct sensors might be better suited for capturing more precise physical measurements (as they are intrinsically related directly to the body or biological systems of the dancer), they may be insufficient in the higher-level performance context, spatialization, and localization of the dancer. This scenario begins to shed light on the power of complimentary modalities—the ability for disparate modalities working together within a multimodal system to enable a broader range of information to be obtained.

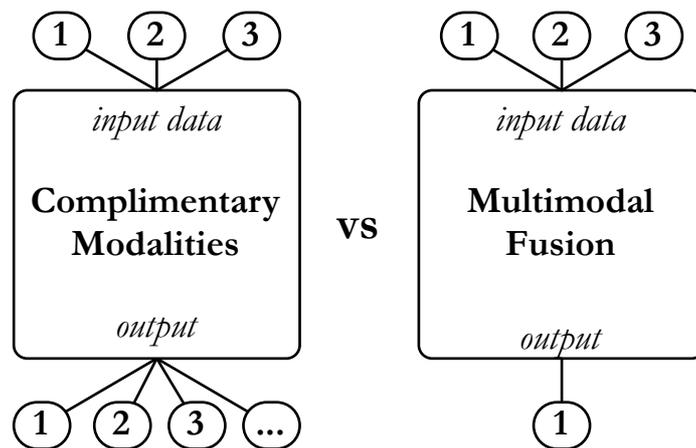


Figure 3: Overview diagram of Complimentary Modalities vs. Multimodal Fusion

While Oviatt presents an interesting view of complimentary multimodality, it is an important distinction from multimodal fusion. As in the McGurk effect example described earlier, multimodal fusion is when information from separate input modalities is combined into one final output. Similarly, Nigay and Coutaz also describe this distinction (what they call “concurrent” vs. “synergistic” uses of modalities), as part of their design space for multimodal systems (Nigay and Coutaz 1993). In their design space, they describe that fusion may also be performed with or without *meaning* of the data streams. The distinction of “levels of abstraction” in fusion (meaning/no meaning) is important to make, as the actual implementation results in different fusion approaches, namely early and

late-fusion. Early fusion (also called subsymbolic fusion) fuses data at the feature level, and is typically suitable when there are strong (close) temporal bonds between the input modalities (this is the primary technique used in 4.3, 4.4, and 5.4). Late fusion (also called symbolic fusion) fuses data at the semantic level (after the feature data has been analyzed for meaning), and is typically suitable when there are weak temporal bonds between the input modalities (although it can also be useful when there are strong temporal relationships between modalities, as will be seen in Chapter 6).

In real world scenarios, however, it is important to note that often the power of multimodal systems emerges by exploiting both the possibilities of complimentary modalities and multimodal fusion, often simultaneously, depending on the desired outcomes. As such, this is one of the primary goals of this work—to harness the potential of these two techniques on multimodal musical input.

1.4 Overview

In order to examine multimodal musical interaction in this dissertation, it is important to first understand what has already been explored. Chapter 2 presents related work by other researchers in the field and is organized as follows. A brief history of related work in HCI and musical multimodal systems is provided in 2.1, followed by a review of musical physical computing (that has informed this work) in 2.2. In 2.3, related works in machine musicianship are presented, which have directly influenced the data mining and metrics work used throughout this research. Lastly, as a large portion of the work in this dissertation turns to machine learning, a brief history of related machine learning in music is provided in section 2.4.

The body of research contributions and experimental trials contained in this dissertation are presented in Chapter 3 through Chapter 7. As illustrated in Figure 4 the multimodal systems used throughout this research will first be introduced (section 3.1). This includes descriptions of the instruments and sensor systems employed in the research, as well as Nuance, the multimodal data

recording software system custom created to support the research carried out in this dissertation.

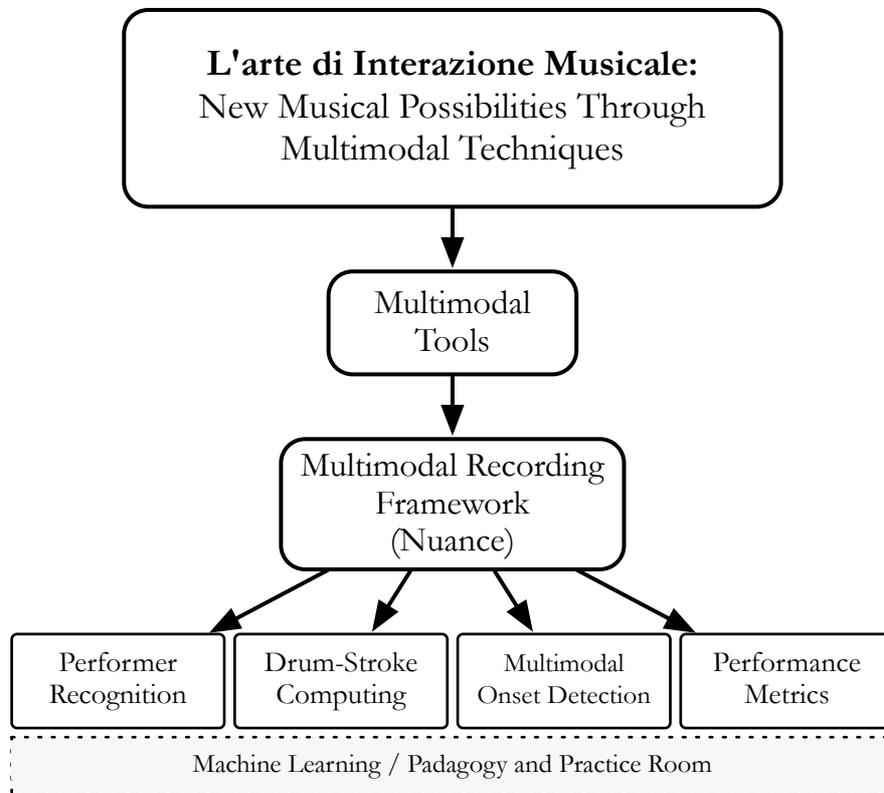


Figure 4: Overview of Research

Chapter 4 through Chapter 7 investigates the possibilities of multimodal musical interaction in a number of scenarios, which support a performer’s musical practice, and creative processes. The individual research cases include multimodal techniques for performer recognition (Chapter 4), drum-stroke computing (Chapter 5), onset detection (Chapter 6), and performance metrics tracking in musical learning environments (Chapter 7).

This research hopes to show that the “art of musical interaction”, today and in the future, is a computer-mediated combination of effective musical practice, and affective musical performance. To this end, the holistic goal of this research is to investigate the role of multimodality in musical HCI. Specifically, this research aims to show that multimodal approaches can in fact support a musician’s craft, in terms of daily practice, analysis scenarios, and in the creative processes.

1.5 Summary of Contributions

The following list provides an overview of the multimodal contributions presented in the dissertation, and is organized by order of appearance.

1. Nuance Software

- a. First cross-platform application specifically designed for capturing multichannel, multimodal data streams in musical scenarios.
- b. Provides support for nearly any instrument, hyperinstrument, and sensor system via serial, MIDI, Open-Sound-Control, and audio channels.
- c. Delivers sample-synchronous data capturing with high sampling rates (up to 192kHz)
- d. User-configurable with a drag-n-drop interface, designed to be operated by researchers and musicians alike, without the need of computer programming or patching.

2. Performer Recognition

- a. First research that quantitatively shows the significance of a multimodal approach for performer recognition tasks over previous audio-only based approaches.
- b. Provides a test bed to experimentally look at the data and features extracted from the Esitar and snare drum performance to support future investigations into performance metrics, tracking, and musical pedagogy in the remainder of the dissertation.
- c. Can train the computer to recognize sitar and drum performers from beginner to advanced skill levels.

3. Drum-stroke computing

- a. First work in automatic drum-hand recognition, which can be used in many tasks ranging from performance metrics, to automatic transcription, and rudiment recognition.

- b. Provides a multimodal look at drum performance metrics and statistics, introducing multimodal features such as cross-modal Onset Difference Time (ODT).
- c. Uses multimodal surrogate data training to automatically label training data in machine learning scenarios.

4. Multimodal Onset Detection

- a. Novel algorithm for improving onset detection accuracy using multimodal fusion.
- b. Late-fusion technique is algorithm independent, meaning it can be used with current (and future) onset detection functions.

5. Multimodal Performance Metrics and Musical Pedagogy

- a. First focused investigation into the roles of multimodality for musical practice and pedagogy scenarios.
- b. Provides multimodal analysis into meaningful performance metrics and statistics for practicing bowed string instrument players, including tempo analysis, and bow articulation metrics.
- c. Delivers first long-term performance study of bowed string instrument performance using multimodal analysis.

Section II

Related Work

Chapter 2

Background and Motivation

The work in this dissertation combines concepts in multimodality, music related physical computing, machine musicianship, and machine learning. As such, this chapter begins by discussing related work in multimodality in 2.1. Multimodality is presented both in its foundations in the greater field of human-computer interaction, as well as in the field of Affective Computing, followed by early examples of multimodal techniques in musical applications. In 2.2, a brief history of music related physical computing is introduced, specifically focusing on instruments and interfaces that influence this research. Finally, a general overview of influential machine musicianship and machine learning research is provided in sections 2.3 and 2.4 (respectively). While not exhaustive, this chapter serves to provide an overview of related work (and areas) in which this research draws upon or is inspired by, in its application of multimodal techniques to musical interaction.

2.1 A Brief History of Multimodality and HCI

As human interaction is highly multimodal in nature, the perceptual and cognitive sciences have explored multimodal theory and approaches³. As such, related fields with computer driven mediums, such as HCI, have also adopted multimodal approaches, and multimodality has now become an important aspect of modern user experience and interaction design. Multimodality in HCI emerged with Bolt's "Put-That-There" voice and gesture system, developed at

³This section does not go into depth regarding multimodality's foundations in the cognitive sciences. Dumas et. al. provide a good overview and historical context, specifically in cognitive load theory, gestalt theory, and Baddeley's model of working memory in (Dumas, Lalanne, and Oviatt 2009).

the Architecture Machine Group at MIT in the early 1980's. In this system, users could issue commands for a large screen to display, by simultaneously pointing and speaking. Pointing to a position on a large screen and saying "put a green square there", would fuse the location detected from a sensor measuring where the users hand was pointed, with the recognition of the voice commands, instructing the computer to create a green square at that particular location. This was an early example showing the convergence of multiple modalities, and how they can fuse to provide a natural interface with "increased precision in its power to reference" (Bolt 1980). The point-and-speak method of multimodal interaction set the tone for much of the subsequent multimodal HCI research, such as the CUBRICON mouse-and-speech recognition system (Neal, J.G. and Shapiro, S.C. 1991), and other notable early work such as a system that enabled interacting with 2D and 3D maps by integrating speech, gaze, and hand gestures (Koons, Sparrell, and Thorisson 1993). Generally speaking, Dix et al. concluded in "Human-Computer Interaction" (first published in 1993), that multimodality is an important aspect of HCI in that it enables

1. Increased *bandwidth* of interaction between the user and the computer, and
2. More *natural* human-computer interaction (closer to everyday human-human interaction),

while at the same time reducing the amount of overload which may occur on a particular modality (e.g. visual) when a system and its behaviors become increasingly complex (Dix et al. 2003).

In recent years, multimodal HCI has ventured outside the point-and-speech paradigm that emerged from Bolt's "Put-That-There" system, looking to other modalities to further increase the bandwidth and richness of user interaction. Other fields with strong connections to HCI have since also begun to investigate multimodal integration, particularly Affective Computing, and also music and interactive arts. In the following sections, additional historical references in the aforementioned fields will be briefly discussed. For additional information on the

history of multimodal HCI and other early examples of multimodal user interfaces, please refer to (Raisamo 1999; Dumas, Lalanne, and Oviatt 2009).

2.1.1 DETECTING AFFECTIVE STATES

Multimodal theory and its application to human-computer interaction are also deeply connected in the field of Affective Computing. Affective Computing, as described by visionary pioneer Rosalind Picard, is “computing that relates to, arises from, or deliberately influences [human] emotions.” Picard’s Affective Computing Group at MIT Media Lab and other researchers in the field believe that emotion plays an crucial role in the human experience; thus, affective computing builds off the fundamental principle that everyday tasks such as cognition, communication, decision-making, and learning, heavily rely on the (human) ability to process multiple channels of affective information simultaneously. In order to make human-computer interaction more meaningful, affective computing explores the use of sensor-systems and technologies to make computer systems more emotionally “intelligent”, or aware of its users.

Early research in affective computing has focused on unimodal analysis, for example, detecting human emotional states using video-based motion capturing systems (Asha Kapur et al. 2005). In recent years, however, the field has largely moved towards multimodal signal processing for detecting affective states. In “Multimodal Affect Recognition in Learning Environments”, Kapoor and Picard present a framework for recognizing affective states while learning (Kapoor and Picard 2005). The multimodal system designated in the research can detect affective states by extracting non-verbal behaviors (features) from facial expressions and postures. This is achieved using real-time face tracking and a posture-sensing chair. Many other examples in affective computing exist are also applying multimodal techniques. Busso et al. used decision and feature level fusion (late-fusion and early-fusion) of motion capture (facial expressions) and speech (acoustical) data to recognize four emotional states of a user (sadness, anger, happiness, neutrality) (Busso et al. 2004). More recently, Kessous et al. explored recognition of eight emotional states from ten participants, integrating

multimodal data from facial expressions, body movement and gestures, and speech (Kessous et al. 2010).

2.1.2 SELECTED EXAMPLES OF MULTIMODAL MUSICAL SYSTEMS

While the above are examples from research showing the applications of multimodality in the field of affective computing, multimodality has begun to permeate musical research and performance. The Casa Paganini InfoMus Lab at the University of Genova was established in 1984, and has long been interested in human gesture recognition for musical and multimedia performance. As such they have led many investigations in multimodal analysis for musical performance; one example being a vision tracking and sensor-based performance system used in the music theatre production *Cronaca del Luogo* by Luciano Berio (Berio 1999). The InfoMus Lab is also responsible for developing EyesWeb (Camurri et al. 2007), a platform for research and applications in multimodal analysis and gesture processing. Providing a patching environment where multisensory inputs and gesture recognition blocks can be connected and synchronized, EyesWeb has been used in many real-time performances and research experiments. Additionally, features of EyesWeb motivated the development of the Nuance system described in 3.2.

Hyperinstruments (discussed in greater detail in section 2.2.2) are typically multimodal in nature. One such hyperinstrument that has influenced many aspects of this research is the Esitar (Ajay Kapur 2008). Using a variety of sensors to measure various aspects of the performers technique and playing (e.g. thumb pressure sensor, fret detection sensor, instrument tilt sensor), Kapur's work with the Esitar is an early musical example demonstrating the far-reaching affordances of integrating multimodality and musical HCI. Motivating examples which have inspired this research include transcription of multimodal performance data for musical pedagogy (Ajay Kapur et al. 2007), late-fusion tempo tracking for human-robot performance (Benning et al. 2007), among others.

Another hyperinstrument that has influenced particular aspects of this research is the Hyperbow (Young 2002). In addition to engaging with a violin (or

cello) to produce its regular acoustic output, the Hyperbow streams multiple channels of information to the computer from multiple modalities, making it extremely expressive in both performance and data mining contexts. These include various position measurements from the bow, as detailed in 2.2.2. The sensing technologies can be used in a number of applications, from analyzing player performance data, to controlling performance parameters and synthesis of physical models in real-time.

In (Tanaka and Knapp 2002), a multimodal, multichannel musical control system is implemented using Electromyogram (EMG) bio-signal sensing, and relative position sensing (pictures in Figure 5). The authors describe the scenario where an EMG on an individual's bicep would report copious activity if the individual were steadily holding a heavy weight, but not portraying active movement to the audience. Because EMG sensing (which measures muscle activity) may or may not reflect actual perceived muscle motion, a multimodal approach integrating the EMG data with other motion sensing makes the system more controllable, and expressive for the performer.



Figure 5: EMG biometric and gyro-based position controller (arm bands, headbands and base) used in (Tanaka and Knapp 2002)

In this way, the authors intend position to serve as the primary musical control, which is then further modified by the muscle-tension information provided by the EMG (and vice-versa). The authors further warrant that due to

the fact that both modalities can be multichannel, the system provides a highly expressive, and fluid, multidimensional musical environment for performance.

Multimodal musical interfaces can also augment the performance space (rather than the playable instrument directly). One example of this is the Multimodal Music Stand (*MMMS*) (Bell et al. 2007), which provides musicians an untethered means of sensing continuous and discrete performance gestures for real-time musical processing. Instead of creating a new interface or a hyperinstrument for a musician to learn and perform, the MMMS enables hands free augmentation for traditional electro-acoustic performance. This is realized via a variety of capacitance sensors (sensing location in 3-dimensions), combined with a microphone for incoming sound processing, and vision based tracking for additional gesture recognition. Using multimodal fusion of all three modalities, the accuracy of the MMMS gesture sensing is greatly increased, while simultaneously providing complimentary streams of performance data. The idea of creating systems that can multimodally augment traditional musical scenarios (either by themselves or in combination) was one of the inspirations to create the XXL system used throughout this research.

Other examples of music related multimodal research has appeared in recent years, often extending multimodality out of the physical-space, and into the symbolic. This is particular true in the field of Music Information Retrieval, where multimodality has been applied to tasks such as genre classification. One such example is in improving automatic genre classification systems using audio (acoustic) features combined with social tags (Zhen and Xu 2010).

Another example is a system that combines audio features with song lyrics, and visualizes the content using the self-organizing map metaphor. In this work, users can navigate the musical material provided by the multimodal linking of the audio library (Neumayer and Rauber 2008). Combining the two modalities (audio-based features and symbolic lyrics) can lead to interesting outcomes, as both modalities intrinsically provide different varieties of data. Whereas the audio feature may provide information about the sonic qualities and content of the music, the lyrics may relate more to the semantics of the content. Combining and thinking about these channels in various ways can lead to interesting approaches to musical navigation, appreciation, interaction, and experiences.

2.1.3 SUMMARY

Multimodal HCI has shown promise in a variety of areas. As demonstrated by systems presented as early as 1980, and other developments in related fields such as affective computing, multimodal HCI can foster affective collaborations between humans and computers. A large focus in the HCI community has been in applying multimodality to every day computer interactions, as well as assistive technologies; however, as exemplified in this section, multimodality has also influenced musical interaction systems, performance, and analysis techniques. This dissertation is primarily concerned with the latter, examining the design and implementation of multimodal systems for capturing physical information from musical performers, and the affordances and possibilities thereof.

2.2 A History of Related Physical Computing

Affording novel musical interactions through new devices and sensor systems

Naturally the research presented requires new musical interface and sensor systems to enable multimodal input, and so the following section provides a brief history of related work in the realm of “Physical Computing” (O’Sullivan and Igoe 2004). Physical Computing, a branch of HCI is a field that has significantly influenced musical interactions in recent years, enabling expressive new modes of interaction and sound sculpting to musicians and composers. This section presents a general overview of hardware systems and techniques that are influential to this research. Section 2.2.1 provides an overview of musical interfaces that enable new modes of musical interaction, while not explicitly augmenting acoustic instruments (although many draw influence in terms of design, musical family, or playing technique). Contrastingly, an introduction to hyperinstruments and other instruments that have been modified with sensor systems can be found in section 2.2.2.

Musical physical computing is an extremely active field, as demonstrated by the popularity of conferences such as the International Conference on New Interfaces for Musical Expression (NIME), and developing communities such as

Arduino⁴, CreateDigitalMusic⁵, and the Monome⁶. As such it is out of the scope of this dissertation to present an overview of the ever-expanding list of musical interfaces and sensor systems currently being created. Rather, this section aims to present a concise set of work that has significantly informed the goals and considerations of this dissertation, specifically the enabling of multimodal musical interaction.

2.2.1 NEW INTERFACES & CONTROLLERS: BUILDING ON AND DIVERGING FROM EXISTING METAPHORS

Many musicians, technologists, and researchers have explored creating completely new interfaces and controllers (often called NIMEs or new interfaces for musical expression) in an attempt to enable new sonic engagements. One can say that these interfaces are new in the sense that they are built from the ground up (as opposed to augmenting other traditional instruments). In terms of interaction however, they can either provide completely new means of input (diverging from existing metaphors), or build on top of existing metaphors (input interactions). An early example of a “new interface” that made use of existing musical metaphors is the percussion-based interface called the Radio Baton (Mathews and Schloss 1989). Built at Bell Labs by Bob Boie, and further improved by computer music pioneer Max Matthews, the Radio Baton measures the individual capacitances between the tips of two batons, and five antennas placed within a base-surface. The system is able to localize the batons in 3-dimensions (providing x, y, and z dimensions of control). Similar to the Radio Baton is the Buchla Lighting III⁷, another baton-based digital interface which also provides x, y, and z degrees of freedom, using infrared based optical triangulation. Lastly, The Rhythm Tree (Paradiso 1999) is another example of an interface utilizing common percussive striking techniques. One of the largest electronic percussion instruments, the Rhythm Tree has over 300 drum pads,

⁴ <http://www.arduino.cc>

⁵ <http://www.createdigitalmusic.com>

⁶ <http://www.monome.org>

⁷ <http://buchla.com/lightning3.html>

sensitive to various kinds of striking (top, side, sharp, and dull), and is equipped with LEDs providing visual feedback to the performer.



Figure 6: Max Mathews and the Radio Baton (left) and the Buchla Lightning III (right)

All three of the examples provided (the Radio Baton, the Buchla Lightening II, and the Rhythm Tree) are self-contained instruments (whether or not they produce sound themselves, or send control signals to other sound producing agents like a synthesizer or computer). With time they can be learned, composed for, and performed. These three interfaces have been exemplified here not only because they have been longstanding influential interfaces in the community, but also because they build on top of existing musical [interaction] metaphors. The benefit of building on top of traditional instrumental techniques is providing a common access point for musicians and composers who have already spent years learning a particular instrument and technique. The user input resembles action paradigms that have been refined and proven effective over years. At the same time, as demonstrated, they can afford new user engagements, both sonically (as demonstrated by the added dimension of control in the baton interfaces), and visually (visual feedback on the Rhythm Tree).

The idea of building on top of existing metaphors will be revisited again in a discussion on hyperinstruments, and exemplified throughout the remainder of the dissertation. However, diverging into completely new interactive domains also poses great potential for new sonic exchanges. One example in particular that has inspired certain aspects of this research is the work of Dutch composer, inventor, and electronic musical instrument pioneer Michel Waisvisz. While at

STEIM (the S^Tudio for Electro Instrumental Music in Amsterdam, Netherlands) Waisvisz created *The Hands* (Waisvisz 1985), a MIDI controller that converts hand, finger and arm movements, and tilting gestures into musical gesture. The use of gestural control is one that has been of great interest in recent years, inspiring adaptive gestural systems in this research 3.1.4, and many other examples in the greater NIME community. Other influential work includes early non-contact based instruments such as the *Theremin*, created in the early 21st century by Russian inventor Léon Theremin (Glinsky 2000). While the in-air playing technique of the *Theremin* is particularly hard to master, the *Theremin* is one of the oldest examples of a radical electronic instrument which similarly to acoustic instruments, can provide amazingly intricate and subtle musical expressivities when mastered.

Thinking outside the typical sound-resonating box has led to the exploration of various musical interactions including an emerging computing paradigm—tabletop surface interaction. The *Reactable* (Jordà et al. 2005) is one such device that uses an infrared vision-tracking system to track various objects (called *fiducials*) and touch events on its surface. Much like the modular synthesizers of the 1970's, each object represents a separate module with the ability to interact with other objects on the surface by manipulating its spatial location and rotation. Some examples of object functions include sound generators (oscillators) and audio modifiers (filters, sequencers, etc.). Another example of musical tabletop surfaces which similarly convert the motion of the tracked objects and touch events on the surface into musical gestures include the *AudioPad* (Patten, Recht, and Ishii 2002). The (multi-user) interactions and visual feedback mechanisms made possible by these large-scale tabletop surfaces can offer many unique musical experiences, and have influenced this research in appendix A.4.

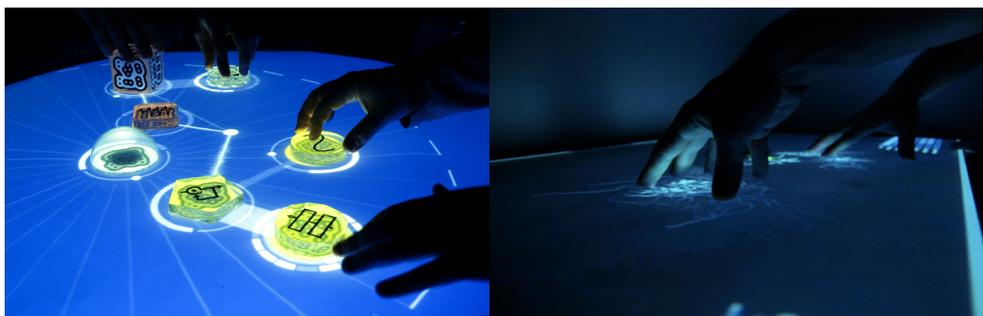


Figure 7: Collaborative music making on the *Reactable* (left), and *Bricktable* “*Roots*” (right)

These examples are all influential to this research in a number of ways. In particular, the question of building on top of or diverging from existing performance metaphors and paradigms has greatly influenced the implementations of the multimodal systems throughout this research. Examples such as the Radio Baton inform this research by showing the affordances of building on top of established interactions when designing new interfaces, while adding completely new degrees of expressive musical freedoms such as in air position sensing in 3 dimensions. Contrastingly, completely new interface paradigms such as tabletop surfaces (tangible and multi-touch) enable completely new modes of musical interactions, and encourage other interactions such as collaborative music making. Others such as Waisvisz's *The Hands* highlight how the human body can be taken one step closer to the interface itself, and the musical possibilities of controllers that enable highly physical, and gesticulated performance.

2.2.2 HYPERINSTRUMENTS

Companions of the NIMEs discussed in the previous section are Hyperinstruments (Machover and Chung 1989; Machover 1992)—instruments designed with the goal of using technology to expand the possibilities of traditional instrumentation. Commonly built upon traditional (or redesigned) acoustic instruments, hyperinstruments are used extensively in this research, not only to provide new channels of control for music parameters, but as windows into performance data from human performance and gesture. The term hyperinstrument was first coined by composer and inventor Tod Machover, of the Hyperinstrument Group at MIT Media Lab. Hyperinstruments have been embraced by a wide-range of notable performers and musicians, including Yo-Yo Ma (hypercello), Prince, and many others. The following provides an overview of the set of hyperinstruments in which this research draws upon.

Diana Young and the Hyperinstrument Group at the MIT Media Lab developed the Hyperbow (Figure 8) to capture the intricate aspects of violin bowing technique (Hyperviolin) from virtuosic players. Once captured, the physical gesture data can be mapped to audio effects parameters to process the

instrument's sound, as well as to control real-time sound-synthesis and physical models of the violin. The original Hyperbow work primarily focused on providing information such as position of the player's hand. However, Young revisited the Hyperbow for her Masters and PhD theses at MIT Media Lab, broadening the scope of the data capturing capabilities to include invaluable information from all applied forces to the instrument by the player (articulation, force, acceleration, changes in position, and changes in downward and lateral movements) (Young 2007; Young 2002). This was achieved by augmenting an electric violin (the RAAD violin designed by Richard Armin) with additional sensors including strain gauges, accelerometers, a 6-degrees of freedom inertial measurement unit, and an electromagnetic field measurement system for position. Young's work also investigates classification of six bowing techniques.



Figure 8: Final Hyperbow violin design by Diana Young

The *Bowed-Sensor-Speaker-Array (BoSSA)* (Trueman and Cook 2000) is an amalgamation and extension of previous work by creators Dan Trueman and Perry Cook. BoSSA combines the *R-Bow* hyperbow designed by Trueman and Cook (providing motion data via a biaxial accelerometer and pressure data via a force-sensing resistor) with sound-spatialization via a multidirectional 12-channel speaker array embedded within a dodecahedron. Using the various sensor

streams provided by the R-Bow, additional sensors on the violin itself (Trueman and Cook 2000), and the 12-channel speaker array, BoSSA is a rich hyperinstrument which not only provides interesting means of sound manipulation beyond that of a traditional violin, but also concurrent sound-diffusion, with the ability to simulate the directivity of many different instruments.

Another influential hyperinstrument to this research is Curtis Bahn's *SBass*. A modified upright bass, Bahn's sensor design was influenced by his signature pizzicato playing, resulting in the decision to have various sensors on the bass itself instead of focusing on the bow, like others designs (including Bahns own *Edilruba*). While many of the hyperinstruments listed have focused primarily on instruments from the western music tradition, together, Bahn and Ajay Kapur have also explored hyperinstruments in non-western contexts, focusing on North Indian Classical music (Ajay Kapur 2008). Kapur's *Esitar* is used for performer recognition in 4.3, and a description can be found in section 3.1.1.

Because hyperinstruments are typically traditional instruments (or instrumental peripherals) modified with sensors, they can be played and practiced as regular instruments—requiring little to no adjustments by the performer. This allows musicians (even beginners) to easily engage with the instrument, without having to become comfortable with an unfamiliar interface. As the majority of this research is concerned with obtaining performance data from traditional instruments, hyperinstrument are an essential element of this research. They offer a nuanced vehicle to obtain performance data from musicians, while enhancing traditional instruments with unparalleled means of musical expressivity beyond their original designs. While hyperinstruments are related to other NIMEs, particularly NIMEs that build on established performance metaphors and techniques, both approaches (building on and diverging from) can be appropriate depending on the task. Both approaches to interface design possess great potentials for musical interaction. As this research will show, both can benefit from and enable new musical interactions in the practice room and performance space, by harnessing multimodal designs and techniques.

2.3 Towards Machine Musicianship

Can quantitative and qualitative tools give the computer ears?

In his book *Machine Musicianship* (Rowe 2001), Associate Director of Music Technology at New York University, Robert Rowe, details the idea that computers must be programmed to recognize and reason about human musical concepts. Just like humans, essential musicianship skills of listening, performance, and composition are required if one wishes to engage with the computer in (musically) meaningful ways. In doing so, it will be possible to create more useful applications for composition, performance, and practice.

In this chapter, we will look at selected analysis and retrieval based methods (many inspired by the work in the field of Music Information Retrieval, or MIR) that inform the research carried out in this dissertation. These methods represent higher-level features—algorithms that serve as descriptors in a musically communicative sense. Examples of these higher-level features include note onset (event) detection, pitch detection, melody extraction, key and chord recognition, beat tracking, etc. The process of obtaining these musically minded higher level features involves extracting various lower-level descriptors from a signal, which may be related to “physical auditory models or to spectral models of sound, or simply be mathematical quirks that happen to show some sort of promise as a sound descriptor” (Collins 2010).

Section 2.3.1 focuses on selected state-of-the art research in determining characteristics of rhythm from performers and section 2.3.2 focuses on pitch detection and estimation techniques.

2.3.1 RHYTHM DETECTION

“Music is to a great extent an event-based phenomenon for both performer and listener. We nod our heads or tap our feet to the rhythm of a piece...without [rhythmic] change, there can be no musical meaning.” (Bello et al. 2005)

Whether tightly codified or free in structure, rhythm is a key aspect of musical history, genre, and individual players' unique style, technique, and proficiency. In accord with Rowe's idea of Machine Musicianship, a primary goal of this research is to enable enhanced machine musicianship through multimodal channels. Thus, having the computer understand the many facets of rhythm in musical material is crucial to this research.

In many cases, making machines understand rhythm is really dealing with the process of dividing a continuous signal (musical performance) into discrete and musically significant events. Depending on the requirements of the task, there are many different ways to go about mining the various characteristics of rhythm, such as tempo tracking, meter tracking, beat deviation tracking, pattern recognition, among others. Here, we will look at a few selected examples that directly influence this research.

The first challenge in mining rhythm is accurately detecting when musical events occur. Bello et al. describe the onset of a musical note as "a single instant chosen to mark the temporally extended transient. In most cases, it will coincide with the start of the transient, or the earliest time at which the transient can be reliably detected." (Bello et al. 2005) In *Onset Detection Revisited* (Dixon 2006), Dr. Simon Dixon explores the use of spectral analysis to improve rhythm detection in situations where the musical material lacks strong percussive instruments.

Another challenge is not only detecting individual musical events in isolation, but also how those events relate to one another in a musical sense. "Beat tracking" is the process in which a machine determines locations of beats in musical material. It is an innate part of human musical cognition, as demonstrated by the tapping of a foot or the synchronization of musicians performing together. In this way, beat tracking is extremely useful in assisting in other lower level tasks, such as defining boundaries and the best way to segment musical material for further feature extraction, like tempo, metric meter, etc. BeatRoot is a system developed by Dixon to perform beat tracking and metrical annotation of audio-based (recorded) and symbolic (MIDI) musical material (Dixon, Simon 2007). BeatRoot builds upon fundamental techniques such as onset detection, as discussed in (Dixon 2006; Dixon 2001).

Researchers have also explored other means of rhythm detection. Examples include Scheirer's work in tempo and beat analysis using a small number of bandpass filters and psychoacoustically inspired processing, to produce onset trains which are fed into banks of comb-filters for tempo estimation (Scheirer 1998); and, Goto and Muraoka's work on real-time rhythm tracking based on chord-changes and higher level compositional structures (Goto and Muraoka 1999). The aforementioned tasks (onset detection and beat tracking) are elemental in musical analysis, and are used throughout the research in this dissertation. More in depth details on note onset detection can be found in Chapter 6.

2.3.2 PITCH DETECTION

Pitch detection (often used to describe the estimation of a sound's fundamental frequency) is another important aspect of machine musicianship. Musical applications of pitch detection are broad, from informing one about their intonation, correcting out of tune vocals and instruments in recordings, or as an expressive sound-processing tool⁸. While basic pitch detection is simple in theory using techniques such as zero-crossing rate (the rate at which a signal changes from negative to positive), these simple techniques prove unreliable in real-world situations. This is attributed to a number of reasons, including the fact that even basic signals can be highly complex waveforms (consisting of multiple sine waves with varying periods), and that in many cases additional noise is present in the signal. The aperiodicity of speech and music signals has led to a wide body of interest and research into fundamental frequency (further referred to as F_0) estimation.

Common methods of F_0 estimation utilize autocorrelation, an algorithm in which a signal is cross-correlated against itself (compared to itself looking for similarities), as a function of a time lag applied to one of the signals. In 1993, Boersma introduced an autocorrelation-based algorithm for periodicity estimation that proved to be considerably more accurate than traditional pitch-

⁸ Antares Auto-Tune (www.antarestech.com) & Melodyne (www.celemony.com)

detection algorithms, even at low-registers where pitch detection algorithms tend to have larger error rates (Boersma 1993). More recently, Alain de Cheveigné from IRCAM presented an algorithm based on autocorrelation that offers many interesting improvements (error rates up to three times lower than competing algorithms) over traditional autocorrelation techniques in *YIN, a Fundamental Frequency Estimator for Speech and Music* (De Cheveigné and Kawahara 2002). Additionally, Geoffroy Peeters has explored an approach for periodicity estimation by combining both spectral and temporal representations (which also make use of autocorrelation). This technique adequately estimates pitch and can visualize signals with multiple pitch content, while reducing octave ambiguity (errors) in pitch estimation. For a review on many different monophonic and polyphonic pitch detection methods, please refer to (Cheveigné 2006).

Time and time again machine musicianship is at the core of musical HCI. It is no surprise, as the aim of machine musicianship is to program the computer to explicitly understand human musical concepts such as pitch, harmony, timbre, intention, etc. As such, components of machine musicianship are present in almost all areas of this research. Whereas traditional machine musicianship approaches deduce musicalities by analyzing the acoustic signal of performances, this research reasons that at the same time, it is important to extend machine musicianship into the physical domain. In doing so, multimodal approaches are proposed in which more nuanced channels of machine musicianship can be established between the computer and human performers.

2.4 Music and Machine Learning

Teaching computers to learn complex musical relationships

Machine learning is a science (stemming from “artificial intelligence”) in which algorithms are composed to learn how to behave, without being explicitly programmed to behave. It is teaching the computer to learn by experience, and to infer the proper output by formulating ideas based on previous experiences. In this way, machine learning can be thought of as emulating the ways in which humans learn through every day encounters with the world. And when

something in the world changes, people learn to adapt in their actions. This is also true of machine learning algorithms—the ability for machine learning algorithms to model complex relationships between data, and to refine or optimize the model’s view in the light of new data.

The past decade has seen innumerable advancements as a result of machine learning, from bioinformatics to robotics, gaming to computing. From speech recognition to self-driving cars, machine learning is a field that is at the forefront of modern innovation and industry. It is not surprising then that machine learning has become increasingly relevant in answering today’s musical questions.

2.4.1 SUPERVISED LEARNING AND MODELING COMPLEX RELATIONSHIPS IN MUSIC

Musical performance is rich in complex relationships in the physical, auditory, and psychoacoustic domains. The way in which humans experience music is through very complex interactions between the various physical, acoustical, and affective properties and phenomena. As such, machine learning enables the ability to model the complex relationships of high and low level musical features (see 2.3 machine musicianship), unlocking a world of possibilities in musical pedagogy, live performance, composition, and many other musical scenarios. A brief primer on supervised machine learning and terminology is provided in Appendix D.

Machine Learning has seen an explosion of interest in recent years, particularly in the music information retrieval (MIR) community. For an extensive review of the field please refer to (Orio 2006). The following section details general trends and topics in the field, and how they relate to the contributions of this research. Much of the musical focus of machine learning in the field has been on music content retrieval, recommendation, and classification tasks. An early example of this can be found in (Wold et al. 1996), but a review of the field will show many more examples. Rather than recapitulate (Orio 2006), this section will briefly mention a few active areas of machine learning in music.

One such active area is in automatic genre classification. Genre classification attempts to label a piece of music with a music genre (tag), which can help in

many tasks such as content browsing, organization, recommendation, etc. Many approaches work on short time low-level and high-level features related to rhythm, pitch, and timbre. An early example of this was proposed by Tzanetakis and Cook in the classification of Classical, Country, Disco, Hip Hop, Jazz, Rock, Blues, Reggae, Pop, and Metal genres in (G. Tzanetakis and Cook 2002). Many other recent approaches have been proposed (Cataltepe, Yaslan, and Sonmez 2007; Seyerlehner and Schedl 2009), with some focusing on the difficult task of automatic classification and browsing of musically similar sub-genres, such as electronic music (Diakopoulos et al. 2009). Alternatives to traditional low-level features have also been proposed, such as the use of explicit semantic analysis (Aryafar and Shokoufandeh 2011); as well as other combinations of symbolic data such as social tags (e.g. artist) (Zhen and Xu 2010), and lyrical content (Mayer and Rauber 2011) to aid in classification.

Another active classification task that this research draws upon is in recognition. One popular example that has gained considerable attention is in bow stroke recognition of string players. This has been actively investigated in the recent research of Diana Young, Fiebrink, and others (Young 2007; Fiebrink 2011; Rasamimanana, Flety, and Bevilacqua 2006; Peiper, Warden, and Garnett 2003). Other examples in gesture recognition have also been explored. Fiebrink and collaborators have applied real-time gesture and feature extraction using a tool called the Wekinator, for composition and performance in (Fiebrink 2011). Brecht and Garnett, proposed work in recognizing beat patterns of a conductor as early as 1995 (Brecht and Garnett 1995).

There are many other active areas where machine learning is being applied in the domain of music. In recent years many approaches have been proposed for automatic instrument identification, spanning acoustic instruments (Herrera, Klapuri, and Davy 2006; Kitahara et al. 2007; Eggink and Brown 2003; Livshin and Rodet 2004; Little and Pardo 2008) and even digital and synthesized instruments (Somerville and Uitdenbogerd 2007). Dannenberg et al. proposed a musical style classifier for interactive performance systems in (Dannenberg, Thom, and Watson 1997). Automatic accompaniment systems have been explored, for example, a system where a computer-driven orchestra learns from a solo performer in (Raphael 2010). Other musical applications that are actively

being researched include automatic playlist generation, musical fingerprinting, automatic segmentation and transcription of music and instruments. Recommendation systems have stirred great interest in recent years, and typically use machine learning based on analysis of musical semantics and tags of a users music collection (e.g. Cano, Koppenberger, and Wack 2005; Yoshii et al. 2006).

This research is particularly interested in the unique opportunities when approaching musical machine learning from a multimodal perspective. To that end, this research shows how multimodality can benefit machine learning tasks such as performer recognition scenarios in Chapter 2, which further motivates a multimodal approach for the other research presented in the dissertation. In addition, multimodal machine learning is used for automatic drum stroke recognition in 5.4, which can be useful in a number of scenarios such as rudiment training and recognition, automatic transcription, and in live performance.

2.5 Summary

Multimodal techniques have greatly influenced (and continue to influence) the world of human-computer interaction. The field of affective computing has made great efforts in adapting and establishing new multimodal techniques to encourage more affective communication between humans and computers. As demonstrated, this can lead to many interesting scenarios in HCI, from every day interactions, to assistive technologies and learning. Because music itself affectively engages both the performer and listener, multimodal techniques are a natural extension of musical interaction. In fact, musical interaction normally occurs across multiple modalities, including aspects in the physical, auditory, and psychoacoustic domains. Thus, this chapter has provided examples of recent work in multimodal musical interaction.

At the core of multimodal interaction is the physical input of the performer (from multiple modalities). As such, this chapter also looked at “physical computing” to investigate the ways in which musicians can input into the computer. In relation to this research, physical computing is presented through two approaches. These approaches either build on top of, or diverge from,

existing performer interaction paradigms. Examples are provided which explore these approaches both using novel performance interfaces or NIMEs, as well as hyperinstruments.

Effective multimodal interaction however requires not only the ability for the user to input data into the system, but also to enable the computer to understand and reason meaningful musical qualities from human performance. As such this chapter also introduced related fields and topics in machine musicianship and machine learning. Machine musicianship attempts to program the computer to explicitly understand musical traits and characteristics such as pitch, harmony, rhythm, timbre, etc. By enabling the computer to deduce human musical concepts, a world of possibilities opens up in musical HCI.

This has been further investigated by recent applications of machine learning in music. Using machine learning, the computer *learns* to deduce complex relationships between musical features and concepts. Popular topics and examples were provided in this chapter in which machine learning is used in a diverse set of musical tasks, from music recommendation and content browsing, automatically labeling of music, bow stroke recognition, and other classification scenarios such as genre and style classification.

In this chapter, an overview was provided of significant work in the aforementioned fields. While the areas are related, they are often investigated separately, or with loose relationships. It is the belief of this research however, that it is through conscious exchanges between physical computing, machine musicianship, and machine learning, that novel (multimodal) musical interactions are possible. Thus, through the examples presented in this chapter, a foundation emerges in which multimodal musical interaction can thrive.

Section III

Research and Implementation

Chapter 3

The Toolbox

Overview of the Multimodal Instruments and Sensor Systems Used in the Research

In pursuing this research a wide-range of multimodal instruments and sensor systems have been custom designed. In doing so this research attempts to lay a solid foundation from which multimodal musical interaction design can be further investigated in the future. The musical universe is one that is immensely complex, and the scope of this research cannot possibly reach all families of instruments or musical contexts. However, it is a primary goal to explore the affordances of multimodality in both western and non-western musical traditions, and across a variety of instruments, from melodic to percussive. In the process, practical design considerations for effective multimodal musical interaction design have been identified, and are later discussed in section 8.3. Provided in this section is an overview of the multimodal systems used throughout the research, and which put these design principles to practice.

3.1 Instruments, Interfaces, and Sensor Systems

This section looks specifically at the various instruments and sensor systems that have been used throughout the research. These systems range from custom built hyperinstruments to auxiliary sensor systems, and have been used in a variety of tasks investigating the role of multimodality for musical performance and practice.

3.1.1 ESITAR

The Esitar (Figure 9) is a multimodal hyperinstrument designed by Dr. Ajay Kapur (Ajay Kapur 2008). Its unique sensor system is designed to capture the performance actions of classical North Indian sitar technique. The Esitar provides a fret-detection system implemented via a series-connected resistor circuit. Essentially, when the performer plays a note, current flows through the string and through every resistor between ground and the currently played fret, resulting in a voltage drop (determined by the sum of the resistors in series up to the played fret). While this provides a fairly robust measure of which fret was played, because the sitar enables, and often requires, the performer to pull the note up as much as a Major 6th on any given fret, the Esitar typically fuses the fret-detection data with real-time pitch detection for increased accuracy in pitch tracking (Ajay Kapur et al. 2007).



Figure 9: Esitar sensor systems, close up of thumb sensor (left), and usb, standard audio jack, knobs, buttons, and switches (right)

In addition to fret-detection, the Esitar employs a thumb-pressure sensor to measure the amount of force applied by the player's plucking hand. Traditional sitar technique requires the player to place their right-hand in a specific location on the neck of the instrument, and is elemental to proper playing technique of the instrument. This is a prime example of how with careful design, a sensor can become specifically embodied to represent elements of a particular instrument, musical technique, and other performer attributes. In addition, a tri-axial accelerometer is embedded into the headstock of the Esitar to measure the

instrument's angle and tilt. These sensors are combined with a series of switches, buttons, and knobs, which enable the performer to engage in the musical performance on many levels, including score following, event triggering, enabling effect and signal processing, as well as algorithmic processes. Over one USB port, the Esitar provides a high level of gestural control, while building off of existing concepts of user interaction. The Esitar was used as part of the multimodal performer recognition experiments found in 4.3 and has also influenced the (research's) established philosophies on multimodal design considerations.

3.1.2 EZITHER

The Ezither (Johnston and Kapur 2012) is a hyperinstrument designed and built by collaborator Blake Johnston under the supervision of Ajay Kapur, Owen Vallis, and the author. The Ezither (Figure 10) is a 10-string zither like instrument that resembles other members of the citre family. The Ezither has a force-sensing resistor placed either underneath or on the side of each bridge (depending on the intended use), five buttons, and three potentiometers, that send information back to the computer via USB MIDI. Additionally the Ezither is played with a modified bow that connects directly to the instrument and sends data from a triple-axis accelerometer to the computer as MIDI. The Ezither was used for multimodal onset detection in Chapter 4, performance metrics tracking of bowing technique in Chapter 7, and in the performance report presented in appendix A.2.

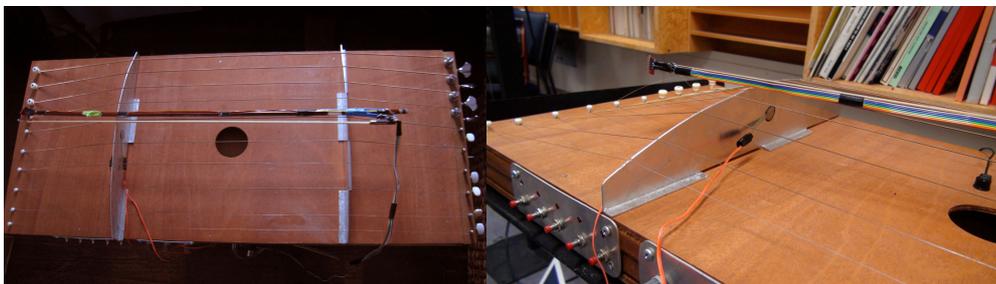


Figure 10: Pictures of the Ezither hyperinstrument and bow

3.1.3 ESULING

The Esuling (Figure 11) is a traditional Balinese ring flute (suling) that has been retrofitted with a multimodal sensor system for real-time musical interaction and data capturing (Erskine and Kapur 2011). The author co-advised the design and build of the Esuling with Ajay Kapur, to create a highly flexible and capable hyperinstrument. Near the air jet of the instrument is a microphone providing an audio stream of the instrument's output. A tri-axial accelerometer is also affixed to the body of the instrument, converting the performer's playing gesture into real-time control signals. Attached ergonomically onto the shell of the instrument are buttons that enable various performance tasks to be executed by the performing musician, as well as a pressure (force-sensing resistor) sensor and position sensor (linear soft-pot FSR). The Esuling is used in the performance discussed in Appendix A.2.

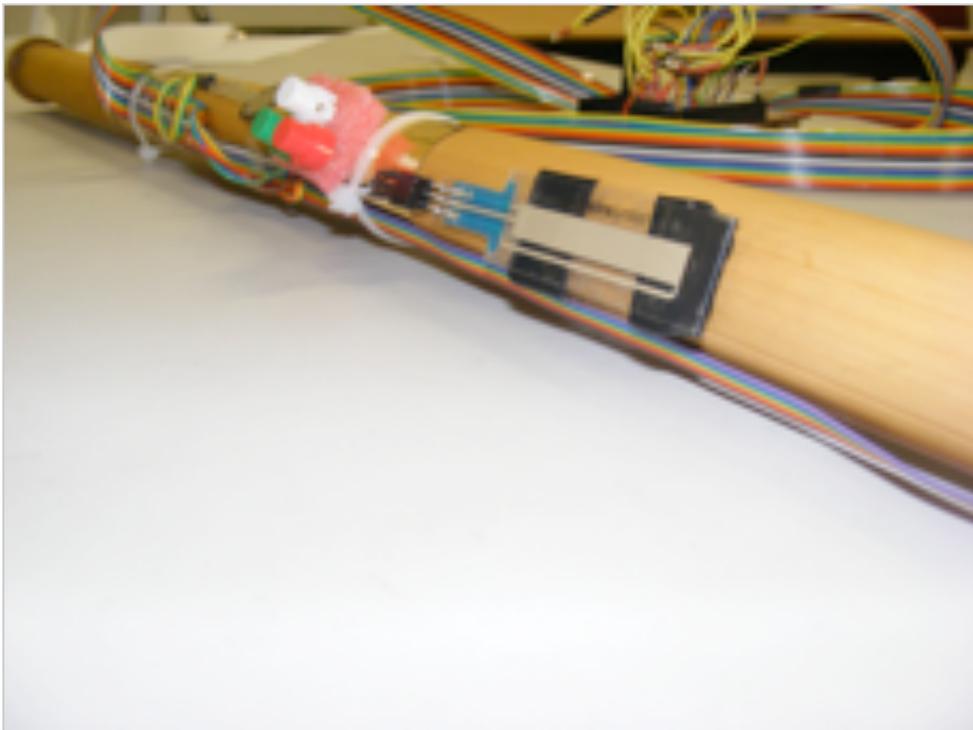


Figure 11: Picture of the Esuling controller showing the two FSRs and buttons

3.1.4 XXL

XXL (pronounced double-accel) is an all-purpose wireless accelerometer system used to quickly capture gestural information from performers, instruments, or anything else they can be attached to. XXL consists of two tri-axial accelerometers that can be affixed to the desired object(s), and a wireless communication system that transmits the accelerometer data to a receiver module (connected to the computer via USB). This can be read directly in a capturing system (e.g. Nuance, see section 3.2), or in any MIDI/OSC (Open Sound Control) capable application via a serial-to-MIDI-and-OSC translator application called XXLSerial (Figure 12 right). XXLSerial provides a “map-mode” function that bypasses data transmission, and enables individual MIDI or OSC messages to be sent for easy parameter assignments. Additionally, XXLSerial provides a calibration mode and sensitivity adjustment to customize the response and feel to the user’s preference.

The transmitting device contains an Arduino Fio which samples the current state of each accelerometer axis with 10-bit resolution (over two IDC ribbon cables), and transmits each reading to a nearby computer over wireless XBee (ZigBee) RF communication.

XXL is used for data collection during drum experiments in sections 4.4, 5.4, 5.5, and on a bow and dancer for gestural control in live performance in appendices A.2 and A.3.

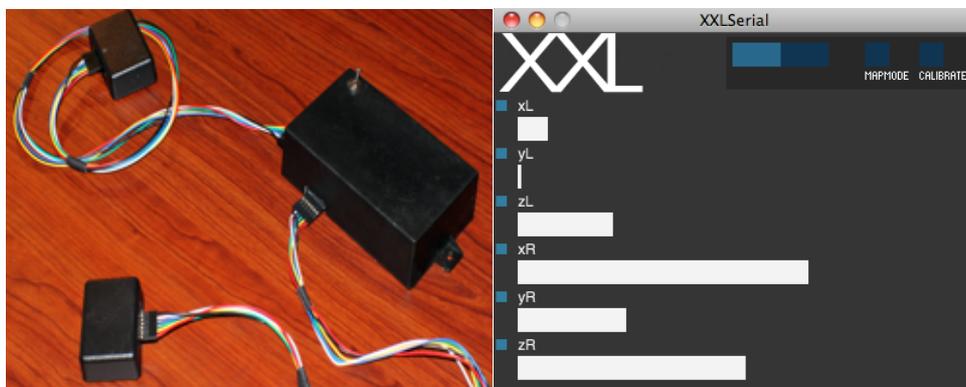


Figure 12: XXL sensor system (left) and screenshot of XXLSerial MIDI/OSC translator (right)

3.2 Nuance: A Software Tool for Capturing Synchronous Data Streams from Multimodal Musical Systems

The previous section discussed the various hardware systems used throughout the research. In this section, the capabilities of a novel multimodal software application called Nuance are presented. Nuance is a software application for recording synchronous data streams from modern musical systems that involve audio and gesture signals. The application currently supports recording data from a number of input sources including real-time audio, and any instrument, musical interface, or sensor system, which outputs serial, OSC, or MIDI. Nuance is unique in that it is highly customizable to the user and unknown musical systems for music information retrieval (MIR), allowing virtually any multimodal input sources to be recorded with minimal effort. Targeted toward musicians working with MIR researchers, Nuance considerably minimizes the set-up and running times of MIR data acquisition scenarios. Nuance attempts to eliminate most of the software programming required to gather data from custom multimodal systems, and provides an easy drag-and-drop user interface for setting up, configuring, and recording synchronous multimodal data streams.

3.2.1 INTRODUCTION TO NUANCE

As described previously in related work, multimodal signal processing is a fundamental aspect in every day human interaction. Humans process information from a variety of senses to deduce meaning when engaging with others (verbal communication, body language, etc.), or with their environment. Humans and other living organisms can also compensate for one sense with another, when the environment places constraints on a particular sense.

As such, processing information from a variety of channels is an emerging area of research in computer and cognitive sciences; fields such as HCI and affective computing have proven some of the benefits of multimodality for more emotively aware interaction between humans and computers. As music is a domain rich in information on many levels (from the score to the physical attributes of a particular performance), researchers have begun to investigate

applying multimodal techniques to musical analysis. While analyzing information from both the acoustic output and the output from various sensors on instruments, performers, and other kinds of systems is promising, there are many challenges ahead, even for the simple task of acquiring the data.

Imagine a common scenario where a researcher is investigating some music related problem. Whether the task is a classification problem, clustering, pattern matching, query/retrieval, musical perception and cognition problem, etc., all tasks share the initial step of acquiring and preparing the data set. While this point seems quite trivial, consider the following. Say the task is a performance metrics problem and the data set is a collection of features extracted from microphone recordings of a drummer. The researcher would like to perform a similar experiment with a saxophonist. No problem, there are tools the experimenter could easily use to record the audio, perform feature extraction, and finally analysis. This scenario, however, becomes much more difficult when the experiment involves custom instruments, interfaces, and multimodal/multisensory input systems. Let's say the drummer mentioned is playing a drum modified with various sensors on the drumhead and stick, the data of which is to be captured alongside the audio recording. Similarly, an accelerometer and air-pressure sensor measures other characteristics of the saxophone performance. Given the highly individualized nature of working with different instruments and musical contexts, each problem requires a different software tool to be written for acquiring the data set. Imagine being a recording or live sound engineer and requiring a specific piece of hardware, or software plug-in, to interface with each instrument being used in a performance. In this section we describe a software tool called Nuance, which begins to address such scenarios. Nuance aims to bring the task of gathering multimodal data sets for MIR one step closer to the ease, usability, and productive workflow refined in traditional Digital Audio Workstations (Duignan, Noble, and Biddle 2010).

The remainder of this section is as organized as follows. Section 3.2.2 describes the motivations behind Nuance, based on the shortcomings of other available solutions. Section 3.2.3 describes the software architecture and capabilities of Nuance, the program workflow is discussed in 3.2.4, and lastly conclusions are discussed in section 3.2.5.

3.2.2 BACKGROUND AND MOTIVATION

Before creating Nuance, a number of available software options were considered. While not comprehensive, the tools discussed in this section were the most ubiquitous tools that appeared to fit the required use cases. The main requirement was to output synchronized recordings from a variety of input sources including audio, MIDI, OSC, serial sensor interfaces, and hyperinstruments. Figure 13 offers an input requirement comparison between five of the available software and framework candidates studied.

The three candidates represented by fully dashed rectangles in Figure 13 (MARSYAS, ChucK, and the CREATE Signal Library or CSL) are popular programming languages or frameworks that are capable of multimodal data collection. Both MARSYAS (George Tzanetakis and Cook 1999) and ChucK (Wang 2008), for example, have many features for performing data capturing, analysis, machine learning, retrieval, and synthesis. While they are capable of receiving audio, MIDI, and OSC input streams, they do not currently support general purpose COM/Serial IO. Serial communication is a significant factor as many of the custom interfaces and sensor systems used in these types of scenarios output serial messages. Another key factor in deciding not to use these three candidates was that a major requirement was to use a tool that required little to no programming to operate. With all three candidates, a custom application would have to be written for each particular experimental setup, as well as implementing a synchronization scheme from the ground up. We desired an application that practicing musicians could run independently, and which requires as little technical know-how and investment of time as possible. To do so, the application would need to provide an easily navigable user interface (GUI).

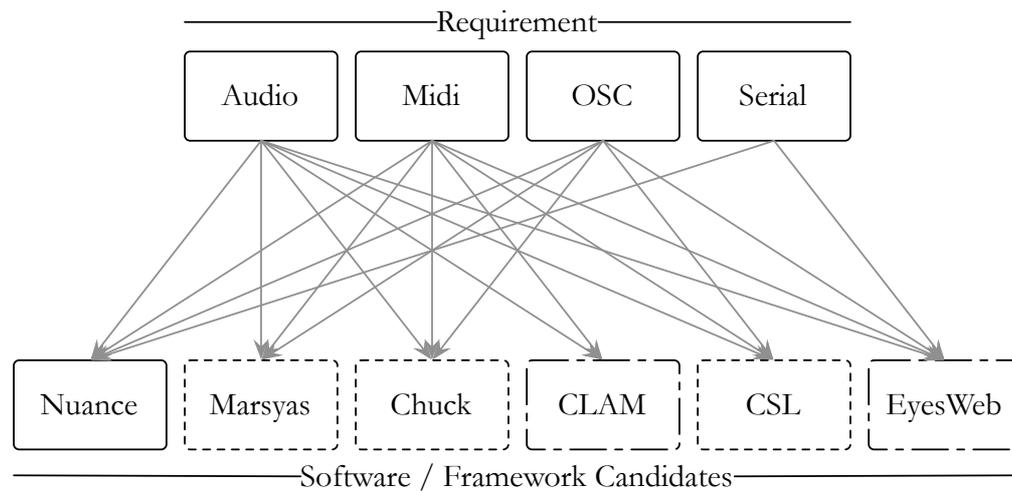


Figure 13: Requirement comparison of other software and frameworks considered as of May 2012

CLAM (Amatriain, Arumi, and Garcia 2006) and EyesWeb XMI (Camurri et al. 2007) (semi-dashed rectangles in Figure 13) are two frameworks that offer a wide array of features and an interactive node-based patching environment. CLAM includes a Data2Audio transformation module as well as a module to export the audio stream to disk. EyesWeb supports all required input streams, as well as providing support for additional input streams, like motion capture data. Additionally EyesWeb XMI provides a configurable data synchronization scheme. While both options seemed viable at first, they did not meet the requirements in the following ways. Firstly CLAM does not support (to our knowledge) OSC or serial input. Secondly we found that the node-based patching environments of both CLAM and EyesWeb were powerful solutions for configuring many complex scenarios and experimental systems. However, the goal was to utilize a tool that focused solely on data capturing, which unsupervised, could be easily configured and used by practicing musicians. In our trials, having to patch and synchronize each instrumental setup individually was found to be too labor intensive and a more tailored solution was desired. Other common visual-based programming languages such as Max/MSP and Pure Data are also capable of the desired tasks, and provide additional support or accessing data from other inputs streams, but similarly required bespoke software patching, synchronization, and configuration for each experiment.

While the above tools all share the ability to capture data from various sources, and some even provide additional machine learning capabilities under one package, none fulfilled our requirements in dealing with the scenarios as described in section 3.2.1.

The necessity for a highly adaptable multimodal data acquisition system is growing as analysis tools continue to get better, and as multimodal strategies become more reliable in solving MIR related problems (Benning et al. 2007; Hochenbaum, Kapur, and Wright 2010; Hochenbaum and Kapur 2012; Ajay Kapur et al. 2007; Tanaka and Knapp 2002). As previously stated, current solutions require the time-consuming task of writing individualized programs or patches for each instrument or sensor system. This is counter productive as hyperinstruments continue to gain popularity, and as industry produces hybrid digital instruments⁹. In this way, we aim Nuance towards the ultimate goal of being a software solution that enables tapping into these types of instruments, with the ease and usability achieved in the common audio-recording software paradigm. We imagine that it is possible to work within an environment where capturing multimodal musical data, whether during the sessions of an album recording, or for MIR related research, is as easy as working with typical multi-track audio recording software.

3.2.3 ARCHITECTURE AND IMPLEMENTATION

Nuance has been designed such that it can synchronize and record data from a variety on inputs and modalities. This section provides an overview of the Nuance recording system and its capabilities.

DESIGN OVERVIEW

As mentioned in section 3.2.2, the primary aim of Nuance was to develop a recording application with a traditional DAW-like workflow. The software should be intuitive to use by regular musicians, while providing a high degree of

⁹ E.g. Gibson HD.6X digital Les Paul Guitar, YouRock MIDI Electric Guitar, Rock Band 3 Stratocaster Pro, Fretlight Guitar

flexibility and support for a variety of heterogeneous input data streams. As such, the following list provides an overview of the main software requirements:

- Support for a variety of input sources including audio, MIDI, OSC, and serial sensor interfaces
- Minimal programming required (little to no programming or “patching”)
- The ability to save, load, and modify recording setups and sessions
- Easily configurable user-interface
- Recording all data in .wav format for analysis

SYSTEM OVERVIEW

The general flow of the software system is detailed in Figure 14. A user provides various multimodal input streams, which are recorded as audio files. By default, all streams are recorded as 16-bit uncompressed .wav files, at a sample-rate of 44.1kHz. This can be adjusted in the program preferences panel, depending on the requirements and capabilities of the user’s system, up to 24-bit resolution, and a 192kHz sample-rate.

SYNCHRONIZATION

Nuance implements a synchronization scheme driven by the computer audio card’s sample-rate clock (Figure 14). Each sensor or input is responsible for updating itself asynchronously at its own independent rate, and all data-streams are read and recorded within a guaranteed synchronous and thread-safe audio callback system. Whenever a new audio buffer¹⁰ is available, each recorder is simultaneously notified to record its data. For an audio input, this simply means writing its current block of audio. For serial, OSC, and MIDI data, the most recent sample is copied into an array (of equal size as the audio-block) and synchronously written to disk. This sample-and-hold and up-sampling of sensor data happens at a much faster rate than common sensor systems supply new data, and we have found it to be more than sufficient in terms of speed and resolution for MIR applications. Other synchronization schemes are possible, and may be

¹⁰ Buffer-size is adjustable via the “preferences panel”

required in the future if additional data sources are added. Additionally, Nuance has been written to support additional output formats (e.g. SDIF/GDIF¹¹) in the future.

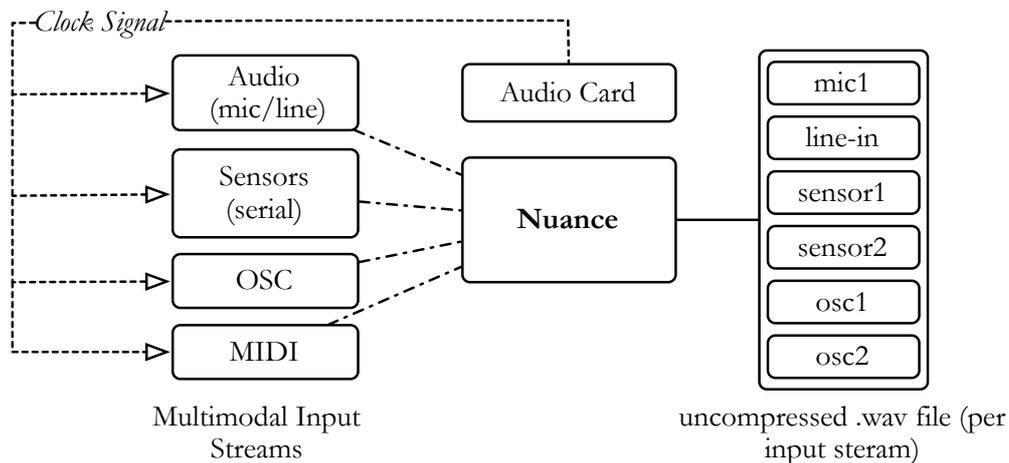


Figure 14: Overview of Nuance input synchronization and output scheme

MULTIMODAL INPUT

A primary concern with Nuance was to support heterogeneous input channels. While the initial four supported input channels are audio, serial, OSC, and MIDI, the Nuance codebase has been written with future extensions in mind. In the following section we describe Nuance’s multimodal capabilities in greater detail.

AUDIO

Mono audio recording is achieved in Nuance by adding an Audio Recorder track to a Nuance session (Figure 15). Each Audio Recorder has the following parameters: real-time waveform visualization, input channel selector, a gain slider, and a record arm button.

¹¹ Sound and Gesture Description Interchange Formats

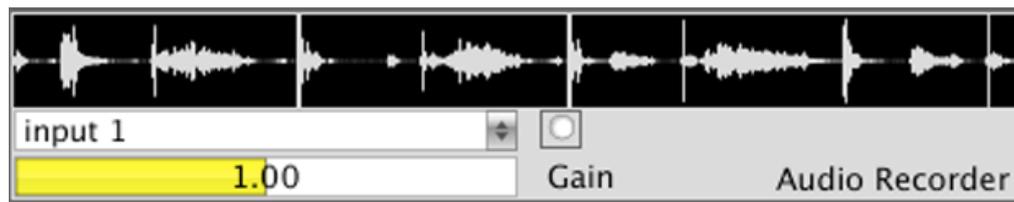


Figure 15: Audio Recorder object

SENSORS (SERIAL DATA)

As many projects in the community utilize Atmel/Arduino/PIC microprocessors, supporting serial communication was a major design consideration. For generalization purposes, Nuance currently supports serial devices outputting data in the following serial format:

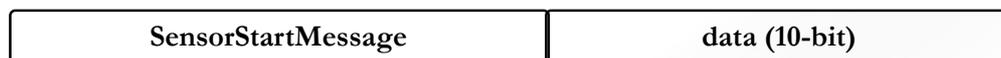


Figure 16: Serial Message Format

A typical use-case using an Arduino microcontroller with two force-sensing resistors connected to analog inputs 0 and 1 might look something like Figure 17.

```
void loop() {
  int fsr1Value = analogRead(0);
  int fsr2Value = analogRead(1);
  Serial.print("fsr1");
  Serial.println(fsr1Value);
  Serial.print("fsr2");
  Serial.println(fsr2Value);
  delay(10);
}
```

Figure 17: Example Arduino serial out messages for two analog sensors

In this example, “fsr1” and “fsr2” would be the *SensorStartMessages*, which are immediately appended by the data, and finally followed by a new line character (via `println`). Nuance uses the new line character to delineate each serial message. Once the serial messages are streaming in the correct format, the user must provide an .xml file (Figure 18) to each sensor recorder object. The .xml file

outlines the expected sensor start messages (“start”), paired with a human-readable name (“ID”) to appear in the Sensor Recorder’s input selector. A built in message configuration panel is being considered for a future release, enabling start messages (and paired human-readable names) to be defined and automatically available to all sensor recorders without having to load an .xml file. The serial-protocol currently implemented was designed for simplicity; however, other more optimized protocols are being considered in the future. For serial-based interfaces that cannot conform to the supported protocol format however, it is still possible to capture data via the OSC and MIDI recorder objects.

```
1 | <TEST_PROTOCOL>
2 |   <DATA>
3 |     <ITEM ID="Force Resistor 1" start="fsr1"/>
4 |     <ITEM ID="Force Resistor 2" start="fsr2"/>
5 |   </DATA>
6 | </TEST_PROTOCOL>
```

Figure 18: Example .xml configuration

Each Sensor Recorder has the following parameters: XML-Protocol loading button, record arm button, serial-device selector (which connected serial-device to acquire data from), input range for automatically normalizing incoming data, and a real-time slider to visualize incoming sensor data.

OPEN SOUND CONTROL

Open Sound Control (OSC) is a versatile communication channel that allows data to be streamed via external sources. The OSC Recorder greatly extends the capabilities of Nuance, making it possible to record data streaming from other applications on the host machine, and from applications and sensor systems connected to networked or remote computers. Additionally, the OSC recorder provides the ability to record sensor-systems or hyperinstruments that do not or cannot follow the generic serial protocol (via a serial-to-OSC middleware).

Example external sources can be anything such as iPhones and mobile devices, vision tracking and analysis systems, real-time feature extractors, and other derived-data outputs. OSC support allows Nuance to support nearly any

input modality or source natively, while keeping its feature set focused solely on the task of providing high-quality, intuitive multimodal recording. Figure 19 shows the GUI elements associated with Sensor, OSC, and MIDI recorder objects.



Figure 19: Sensor (serial), OSC and MIDI Recorders

MIDI

The MIDI Recorder enables data from any native MIDI device to be captured in Nuance. Each MIDI recorder can be configured to listen to individual MIDI note or control change (CC) messages from specific devices, including MIDI-over-network and IAC (InterApplication Control) Bus connections. As all data in Nuance is treated as a continuous stream, when recording MIDI note messages, Nuance does not differentiate between note-on and note-off messages. During analysis however, the rising and falling edges where values transition between zero and the value can be interpreted as note-on and note-off event locations. In the future, when Nuance supports additional output schemes (such as SDIF/GDIF), note-on and note-off events will be preserved in their normal form.

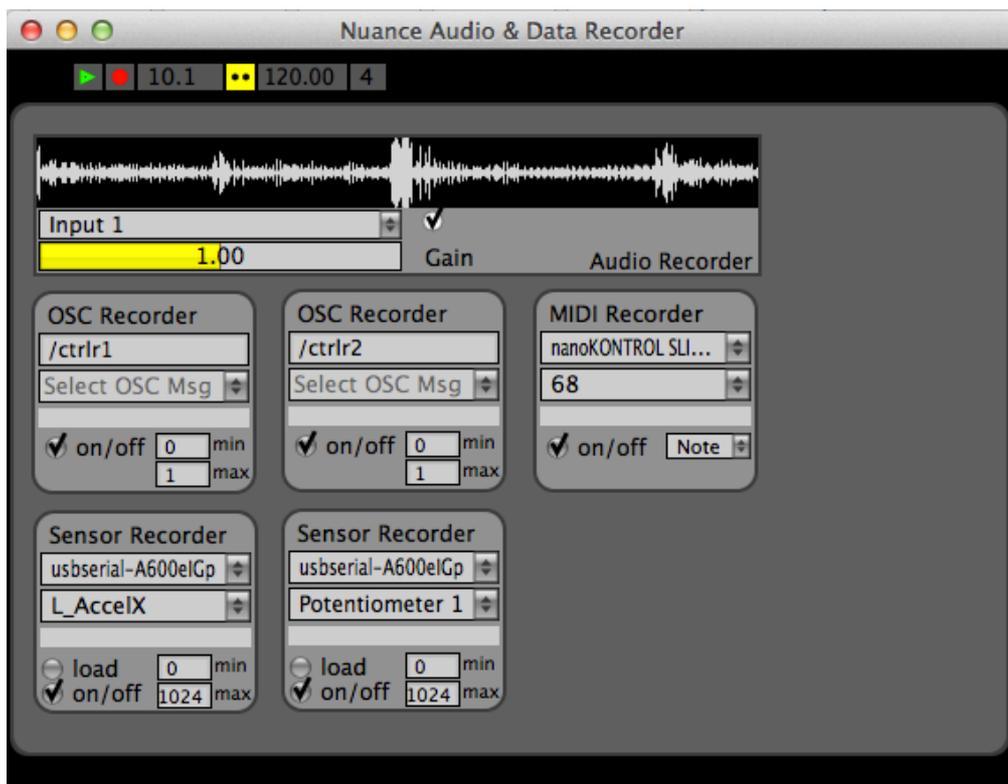


Figure 20: Nuance session main editor panel screenshot

3.2.4 WORKFLOW

A typical use scenario begins with a new empty *session*. Sessions can be thought of as project files, or serializable experiment configurations. Figure 20 shows an example Nuance session including several multimodal data-streams. Most of the user-interaction happens in the session editor panel. Right clicking anywhere in the panel brings up a contextual menu that enables various functions to be performed. These functions include adding recorder objects, unlocking the editor panel to resize and position recorder objects, and saving/reloading sessions. Once a session has been configured, it can be saved for reuse at a later time. Modifications to the session can be made any time during the process and re-saved for future use. A built-in metronome and count-in can also be enabled from the main transport bar, for experimental setups with tightly controlled timing requirements. Lastly, a bar/beat counter is provided to give feedback to the user about how long they have been recording.

3.2.5 SUMMARY

This section described Nuance, a software tool for recording synchronous multimodal data streams. Nuance currently supports high-resolution audio input, as well as input from nearly any musical instrument or sensor system via serial, OSC, and MIDI protocols. Nuance is different than other solutions in that it is a no patching, near-codeless application. Nuance has been designed to be operated by musicians and researchers alike, and has already been used in many real-world scenarios including performer recognition, drum-stroke identification, and performance metrics tracking as demonstrated in the remainder of this dissertation. We look forward to a future where capturing multimodal data streams are integrated into the general workflow of practicing, composing, and performing musicians and composers. Working with multimodal musical instruments enables many unique artistic possibilities, from directly manipulating sound parameters, to extracting higher level features and using them as control parameters. There currently exists just a small list of generalized tools which begin to facilitate these interactions outside of the research laboratory (Fiebrink 2011), and we hope for Nuance to help guide the way in making this accessible to today's musicians and composers. With this in mind, we have written the core of Nuance such that it would remain unchanged in the future if we were to author a cross-platform version in VST/Audio Unit/AAX plug-in formats. Not only can Nuance increase productivity in MIR scenarios, but we hope it points to and establishes a foundation for other future musical endeavors, in the MIR-laboratory, the studio, and the musical classroom.

Chapter 4

Performer Recognition

Can multimodal fusion make the computer understand a human performer?

4.1 Background and Motivation

If one considers music as a temporal evolution of events, occurring within various notions of tonality/atonality, form, harmony/inharmonicity, timbre-space, (pseudo) random and other algorithmic processes, social contexts, etc., music is given function, meaning, or interpretation, when placed within the intent of the composer and performer(s). In turn, this is perceived by the listener when observing a performance (live or recorded), whether on an analytic or purely affective level. Similarly to the ways humans connect on these levels with a piece of music, this research imagines establishing a deeper understanding between musicians and computers through a new multimodal language. In this multimodal dialogue, the computer receives multiple channels of information from the performer and interprets these data to derive meaningful information and communication between the two agents (musician and computer). It is important for the computer first to understand who is the performer, in order to tailor a specific and meaningful interaction. Thus, the musician recognition framework described in this chapter aims to (1) establish a foundational multimodal language that fosters future interactive and educational experiences between musicians and computers, and (2) begins to investigate features or stylistic signifiers between multiple performers' interpretations of musical material.

The common approach to performer recognition uses audio-based techniques to identify characteristics from a recording (Ramirez et al. 2008;

Ramirez et al. 2007; Stamatatos, Efstathios and Widmer, Gerhard 2005; Stamatatos, Efstathios 2002; Stamatatos and Widmer 2002; Stamatatos 2001; Widmer 2001). Stamatatos and Widmer explored this approach to quantify aspects of multiple players' performance "styles" and classify/identify performers using stylistic subtleties (Stamatatos, Efstathios and Widmer, Gerhard 2005). Their use of simple audio-based classifiers to distinguish among a small set of highly trained and stylistically polished players inspired our approach for data capturing.

The approach of this research instead is multimodal in nature, combining audio with data from sensors capturing aspects of a performer's physical performance. Past research on other tasks in the field of Music Information Retrieval produced higher success rates through the use of multimodal instruments as compared to traditional audio-only approaches, while still maintaining transparency between user and instrument. An abundance of musical information resides not only in the sound produced, but also within the performer's physical interaction with the instrument, and this research shows that this physical information is beneficial to the difficult task of player identification.

Two different instruments were used to test a multimodal approach. First, a modified North Indian sitar was used as it is an extraordinarily difficult instrument to master, and requires very specific and demanding techniques for both the musician's left and right playing hands. Additionally, the instrument is rich in subtle expressivities and allows each musician to develop an individual "style" of playing, adding individualized variability to the sitarist's technique. This makes the sitar a great candidate for an empirical study of a particular player's technique, because the musical literature and tradition ask for specific physical actions to be performed by the musician, while the musician develops individual characteristics of his/her own.

Secondly data was collected from ten drummers playing rudiments on a snare drum to extend the task across both plucked string and percussion families of instruments. Drummers also develop strong rhythmical personalities and groove, which could possibly be significant identifiers exposed by multimodal analysis.

4.2 Process

This section provides an overview of the various tools and methodologies used in the experiments. For sitar performer recognition, this included the Esitar hyperinstrument, as well as an early prototype of Nuance codenamed SuperRecorder for capturing synchronous audio and sensor data. For the drum experiments, this consisted of drummers playing a regular snare drum while wearing gloves housing the XXL gesture system described in section 3.1.4.

Figure 21 shows a general overview of the data capturing scenario. Performers play a modified instrument (in this research this is either a sitar as pictured or a snare drum while wearing specialized gloves) and a computer captures the audio output and sensor data. The computer then extracts features from the performance and stores them in a feature vector that is used to train a machine-learning algorithm for player classification/recognition.

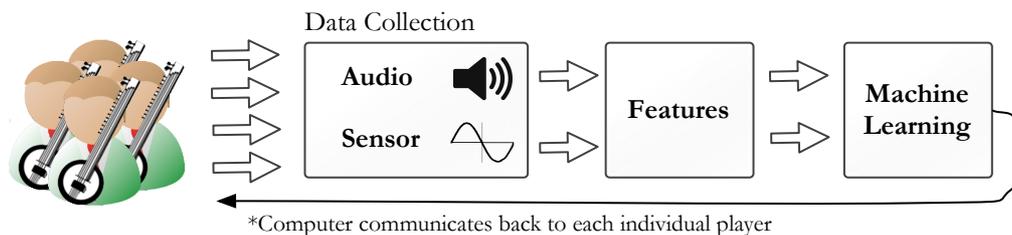


Figure 21: Overview of the performer recognition system (only sitar shown in figure)

The remainder of this chapter is divided into two sections. In section 4.3 sitar performer recognition is discussed, followed by drum performer recognition in 4.4.

4.3 Sitar Performer Recognition

This section explores the task of performer recognition specifically for sitar players. The remainder of this section is as follows. The different musical material (data sets) gathered for the sitar performer recognition experiments is described in 4.3.1. In 4.3.2 we describe the various features extracted from the data sets, and an overview of specific windowing and classification details are

provided in sections 4.3.3 and 4.3.4 respectively. Finally, results and findings are discussed in section 4.3.5.

4.3.1 DATA COLLECTION

Using the system described in the overview, a group of five sitar performers (beginner, intermediate, and expert) were recorded. Each player performed three sitar performance data sets, ranging along a continuum from strictly codified material to improvisation. In each case we recorded audio, thumb, and fret sensor data from each musician.

DATA SET 1 – “EXERCISES” (PRACTICE ROUTINE)

The first data set was designed to record a player’s individual performance characteristics during disciplined practice exercises. Two central exercises from the vast literature of classical North Indian practice methods were chosen: *Bol* patterns and *Alankars* (Akbar Khan, Ali 2004). *Bol* patterns are specific patterns of *da* (up stroke), *ra* (down stroke), and *diri* (up stroke and then down stroke in rapid succession), which are explicitly used in sitar practice plucking training, as well as in performance.¹² *Alankars* refer to scalar patterns that can be modally transposed; they form the basis of many musical ornaments and are also often used for melodic development and fretting practice. We used the *Bol* patterns and *Alankar* exercises shown in Table 1, played in the Indian Rag *Yaman*¹³ at 220 beats per minute. Each of these 15 exercises was repeated as necessary to achieve a duration of 60 seconds.

¹² In general *da* represents the dominant stroke, which for sitar is upwards but for other North Indian instruments such as sarode is downwards.

¹³ *Rag Yaman* uses the Lydian scale, i.e., major with a sharpened fourth scale degree.

Table 1: Bol Patterns and Alankar exercises (data set 1)

Stroke	Da	Ra	Diri
Symbols		-	/\

Bol #	Pattern	Bol
Group1		
3	Da Ra Da	-
5	Da Ra Da Ra Da	- -
7	Da Ra Da Da Ra Da Ra	- - -
9	Da Ra Da Ra Da Da Ra Da Ra	- - - -
Group2		
2	Da Diri	/\
3	Da Diri Da	/\
4	Da Diri Da Ra	/\ -
5	Da Diri Da Ra Da	/\ -
6	Da Diri Da Diri Da Ra	/\ /\ -
7	Da Diri Diri Da Diri Da Ra	/\ /\ /\ -
8	Da Diri Diri Diri Da Diri Da Ra	/\ /\ /\ /\ -

Alankar	Notes	Bol
3	SRG,RGM,GMP...	-
4	SRGM, RGMP, GMPD...	- -
5	SRGMP, RGMPD, GMPDN...	- - -
2+3	SRSRG, RGRGM, GMGMP...	- -

DATA SET 2 – “YAMAN GAT” (COMPOSITION)

A *gat* is a fixed instrumental composition that provides the main theme(s) of a piece. Data set 2 contained ten 60-second recordings of each performer (50 total) repeating a particular *gat* in *rag Yaman* (Akbar Khan, Ali 2004) eight times at 132 bpm.

DATA SET 3 – “IMPROV”

Data set 3 consisted of sensor data and recordings collected from five players each performing ten different 60-second long free improvisations. This data set was completely unconstrained in terms of performers’ technique; it was designed to support experiments to determine whether player performance data is context/piece specific, or truly a technique-based identifier.

4.3.2 FEATURE EXTRACTION

Each sensor outputs continuous information and the recorded audio is also continuous, making the total amount of data a linear function of duration. For classification purposes, regardless of machine learning technique, we need a set of features, each of which collapses the recorded time-series data into a fixed number of scalar quantities. We examined several features from both the audio and sensor data; Figure 22 shows the ones that yielded the best results (best avg. classification).

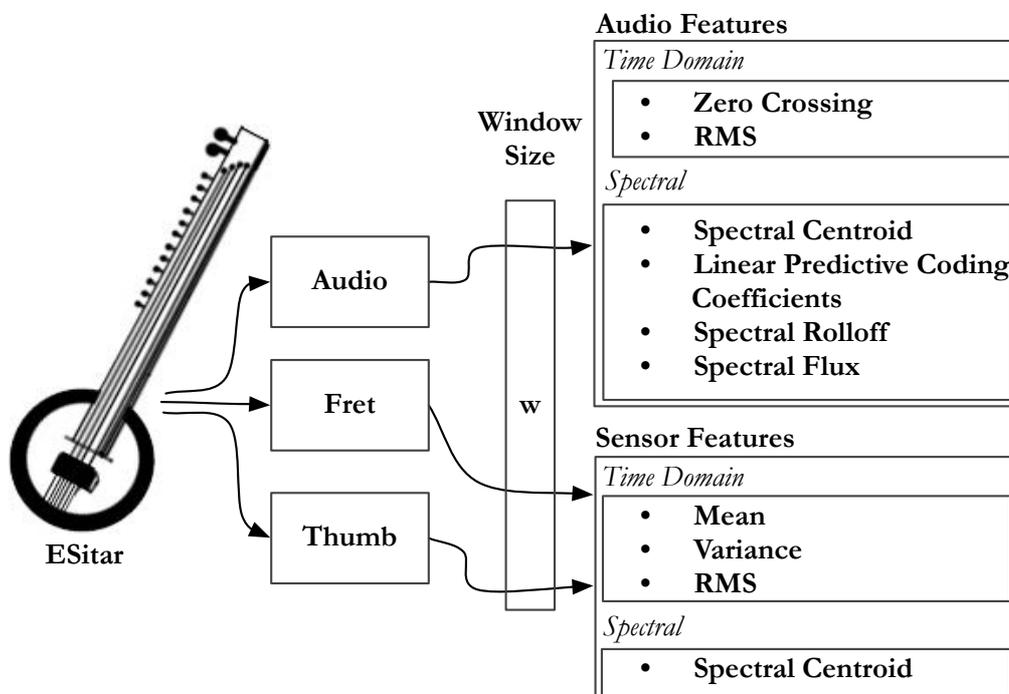


Figure 22: Overview of data capturing and feature extraction

THUMB PRESSURE FEATURES

The arithmetic mean is a simple method of extracting a single characteristic average of the thumb pressure sensor data for each recording. The player-pool included players from various skill-levels; we hypothesized that more highly trained sitarists might maintain a more consistent range of thumb pressure for the duration of a performance as compared to beginner players. To examine this hypothesis, variance was used. Spectral centroid was used to examine high-

frequency transients produced while plucking, effectively relating to the amount of change (from subtle to jerky) in each player’s plucking technique.

FRET FEATURES

Fret *Mean*, *Variance*, *RMS*, and *Spectral Centroid* were also extracted from the fret sensor network. We were interested in determining if data from fretting tendencies and abilities could be effective player identifiers. For example, in data sets 1 and 2, fret mean could be an indicator of how frequently a player’s left hand lost contact with the string. In data set 3, fret mean is a crude indicator of pitch register for each improvisation. The amount of fret variance per window could perhaps suggest the amount of distance and range covered by the players fretting hand at different moments of a performance.

4.3.3 WINDOWING

Each of the data sets consists of 60-second recordings. In addition to computing each feature once per 60-second performance, “windowing” the performances into non-overlapping time segments and computing the features once per segment was also explored. For example, with 10-second segments, each 60 second data recording would be divided into six 10-second “chunks”, and the features would be computed for each chunk, multiplying the amount of training data by a factor of six. After trying various window lengths, a 15-second window was found to yield the best results, and is discussed later in *Windowing Results*.

4.3.4 CLASSIFICATION

Five different classifiers were used in the machine learning experiments. These included a support vector machine trained using *Sequential Minimal Optimization* (SMO), a *multi-layer perceptron* (MLP) backpropagation artificial neural network, *IBk*, which implements the k-nearest-neighbors classifier, decision tree (*J48*), and *Naive Bayes*. More detailed information about these classifiers and *Weka*, the data mining tool used in these experiments can be found in Appendix D and (Witten, Frank, and Hall 2011) respectively.

4.3.5 RESULTS AND DISCUSSION

This section describes the outcomes obtained from the various machine-learning experiments. In each case performance was evaluated using 10-fold cross validation. As each trial included five performers, chance classification accuracy was 20%.

RESULTS: AUDIO ONLY

This section demonstrates the classification results achieved by examining only the features extracted from the audio recordings. The advantage of this technique is that it can be performed with any instrumental player, using only the sound output of their instrument (either with a microphone or direct line), without requiring any modifications to the instrument.

Table 2: Accuracy achieved using audio only (15-second window)

	Exerc. (%)	Yaman (%)	Improv (%)	All (%)
MLP	96.33	100	90	85
SMO	79.33	95.5	81	66.43
Naive Bayes	87.33	98	71.5	58.14

Table 2 shows the classification results achieved using three different classifiers, for each data set alone, as well as all three data sets combined into one large corpus. Multilayer Perceptron proved to be the most accurate classifier in these tests, with the best accuracy being achieved on the exercises and Yaman gat composition data sets. For each pass in those two data sets, each player repeated the same sequence of defined notes/plucks for the duration of 60-seconds. Additionally, in data set 2, each pass contained the same pattern being played for its entirety. These two best accuracies may therefore be the result of slight data over fitting. Still, accuracy on the free improvisation data set, as well as combining all the data into one large pool yielded very satisfactory results.

RESULTS: SENSOR ONLY

Table 3 shows the results of the same machine learning processes applied to only the sensor data. While the accuracy percentage achieved was slightly below the

results from our audio features, the results are very exciting because they show that useful data does indeed reside in the musicians' physical gestural information.

Table 3: Accuracy achieved using sensors only (15-second window)

	Exerc. (%)	Yaman (%)	Improv (%)	All (%)
MLP	84.33	100	89	75.15
SMO	63.67	100	67	60
Naive Bayes	55.67	99	69	46

Again, the highest accuracy was achieved using the Yaman gat data set, for which the sitarists were instructed to play the same scalar and plucking patterns repeatedly, for 10, 60-second long passes. Part of the success of the achieved accuracy may be attributed to the fact that the repetition asked of the players by the data set routine afforded the players ample time to get into a comfortable physical pattern, requiring the least amount of physical change and adjustment compared to the other data sets.

Using features derived only from the sensor data, the improvisation data set yielded the 2nd most accurate player identification across all three classifiers. While this could be the result of chance, it raises the possibility that when improvising, the musicians might have fallen into physical comfort-zones or patterns that they naturally tended to play. In the exercise routines (data set 1), for each pass the sitarists were required to change the fretting and plucking patterns to a hard defined set of practice routines. Because the exercise data set required specific plucking patterns that changed on each pass, and the improv data set allowed the musicians to freely play whatever came to them naturally, it is possible that specific plucking tendencies of the players' technique were exposed through the improv data set, resulting in higher classification accuracy than the exercise data set.

RESULTS: SINGLE SENSOR FEATURES

In addition to testing using a combined set of sensor features, each feature was tested independently to see which features extracted from the sensor data were the strongest. Table 4 shows the results using a 15-second window on the sensor data obtained from all of the data sets combined. The best results were 62.29%

accuracy using Multilayer Perceptron with the thumb-pressure mean feature. In choosing the final feature-set combination for the system (described in section 4.3.2), different combinations of sensor features were experimented with. When comparing Table 3 and Table 4, it is evident that a multi-sensor approach helped increase the performer recognition accuracy by 12.86% (from 62.29% - thumb mean alone, to 75.15% - all sensor features) on all data sets using Multilayer Perceptron.

Table 4: Accuracy achieved using individual sensor features on all data sets, T=Thumb F=Fret (15-second window)

	MLP (%)	SMO (%)	Naive Bayes (%)
Mean (T)	62.29	57.86	59.15
Variance (T)	36	36	34.71
RMS (T)	37.57	34.86	36.43
SC (T)	20.57	24.57	23.43
Mean (F)	21	20.71	20.43
Variance (F)	22	18.71	23.57
RMS (F)	27	21.14	23.71
SC (F)	20.57	24.43	20.57

RESULTS: MULTIMODAL

The results in this section were achieved by combining both the audio and sensor features into a multimodal database. Table 5 shows the accuracy of the same three classifiers applied to all of the data sets as in Table 2 and Table 3. Multilayer Perceptron proved to be the best classifier here, yielding 100% accuracy on all data sets. While the previous trails using either the audio data (features) only or the sensor data (features) only were satisfactory, combining them together into a multimodal database proved to be the most effective solution for performer recognition. This corroborates the use of a multimodal approach to improve systems for musical metrics tracking and performance.

Table 5: Accuracy achieved using multimodal data (15-second window)

	Exerc. (%)	Yamen (%)	Improv (%)	All (%)
MLP	100	100	100	100
SMO	97.33	100	92.5	86.14
Naive Bayes	85.33	100	93.5	67

DISCUSSION: WINDOWING RESULTS

Table 6 shows the accuracy of the system using Multilayer Perceptron over a variety of window periods. The machine-learning experiments yielded the best results with a window size of 15-seconds.

Table 6: Identification accuracy of sensors vs. audio vs. multimodal fusion using a combined corpus from all data sets (at various window periods)

Window Size (seconds)	Audio only (%)	Sensor only (%)	Multimodal (%)
60	84.57	72	93.14
30	85.71	74.57	96.28
15	85	75.15	100
10	84.09	79.24	98.85
5	82.33	76.38	97.76
3	74.97	72.43	96.29

The decrease in reliability of the computer’s ability to perform musician recognition around the 15-second window sweet-spot can be attributed to a variety of factors. As the window size decreases, size of the training set increases accordingly, however, as a result, each feature describes a smaller piece of music. For example, the mean value derived from the thumb pressure sensor at 5-second windows, while providing more “mean values” than larger window sizes may not provide a large enough chunk of music for the extracted mean to be meaningful. Contrastingly, 30-second intervals may not be an appropriate representation of the actual thumb-pressure mean because the mean was not determined frequently enough. Furthermore, (with the one exception of the sensor corpus at 10-second windows), the accuracy identification at 10-seconds, 5-seconds, and 3-seconds, reduces. This suggests that the features need to be determined over a longer window period to allow enough information (samples) to be examined for an accurate representation of the feature.

DISCUSSION: TRAINING AND TESTING ON DIFFERENT SETS

For this experiment the machine learning algorithms were trained using the Exercises and Yaman gat data sets, and then player recognition was attempted on the improvisation data. This is a much more difficult, but perhaps a more “real”

situation, in which the system is trained on a defined set of data and asked it to classify freely improvised playing.

Figure 23 compares the results achieved for audio-only, sensor-only, and multimodally, using the Multilayer Perceptron classifier in each case. In contrast to previous trends, sensor features alone had a slightly higher success rate than audio features alone (30.4% accuracy vs. 28% accuracy). But as with previous experiments however, the multimodal approach was the most successful, with an accuracy rate of 39.2%. Although these results are far from perfect, they are very encouraging in that a multimodal approach improves successful musician recognition even in this more difficult case.

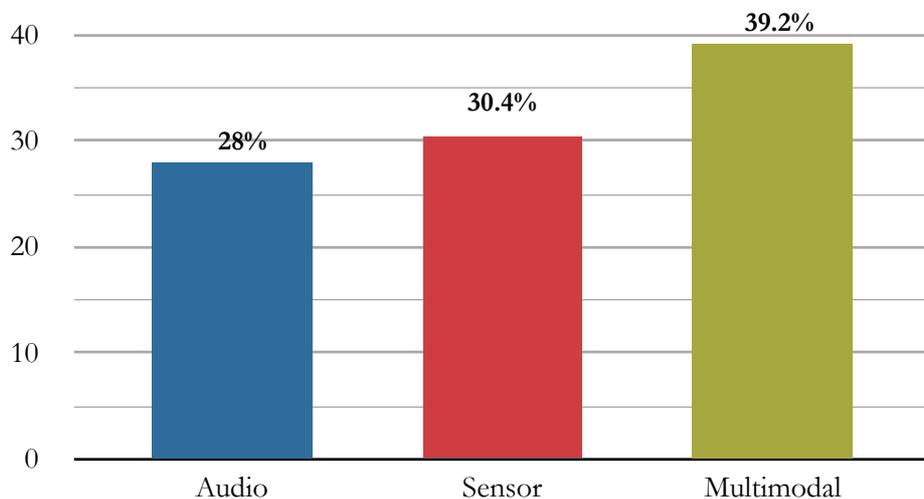


Figure 23: Audio vs. sensor vs. multimodal accuracy achieved for improv data set after training with Exercise and *Yaman* data sets

The fact that the audio features alone performed only 8% more accurately than chance indicates that there may be room for improvement of the system's audio features. Nonetheless the results in the experiments validate the usefulness of a multimodal approach.

4.4 Drum Performer Recognition

In this section similar classification techniques to those explored for sitar performer recognition were applied to snare drum playing. In the aim to support future multimodal musical analysis, in this section we collected a larger

performer pool than in the previous sitar performer recognition experiments. We also begin to investigate useful sensor systems and metrics from drum performance, and establish a framework for future drum analysis in Chapter 5. Lastly, we begin to explore data visualization as a useful tool in machine learning scenarios, and how visualizing features can be useful in identifying performer characteristics.

4.4.1 DATA COLLECTION

SOFTWARE AND SENSOR SYSTEM

Gesture sensors were embedded within lightweight biking gloves that the performers wore while playing the snare drum (Figure 24). Biking gloves were chosen as they typically expose the fingertips and are made from thin lightweight materials, which minimized their interference on the performers' technique.

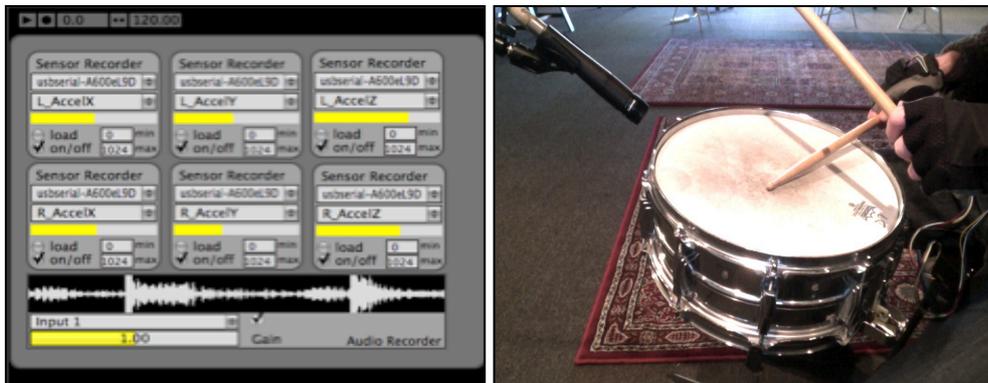


Figure 24: Nuance software (Left) and custom sensor system (Right)

In the experiments Nuance was used to synchronously record three axis of motion from two accelerometers placed on the hands of the performers, as well as a single mono microphone recording the acoustic drum signal. The ADXL335 tri-axial accelerometer was used, as well as a Shure SM57 for recording the audio output of the snare drum. The two accelerometers placed on the topsides of the performers' hands were connected to a wireless transmitter (Figure 24 right). The data was recorded directly over a serial-connection with the receiving XBee module using Nuance. More information on the sensor system called XXL can be found in 3.1.4.

DATA COLLECTION: DATA SETS

An initial data set of 2917 hits from two performers was used to test, and was later replaced with a larger data set collected in these experiments and also for the drum-stroke computing analysis and metrics in Chapter 5. There were ten drummers in total and they were instructed to play four fundamental drum exercises from the Percussive Arts Society¹⁴ International Drum Rudiments; these included the Single Stroke Roll (referred to as D1 throughout the remainder of the work), the Double Stroke Open Roll (D2), the Single Paradiddle (D3), and the Double Paradiddle (D4) as shown in Figure 25.

Figure 25 displays four drum rudiments and paradiddles, each with its corresponding musical notation and R/L labels:

- Single Stroke Roll:** Tempo $\text{♩} = 120$, 4/4 time signature. Rhythmic notation: RLRLRLRLRLRLRLRL.
- Double Stroke Open Roll:** 4/4 time signature. Rhythmic notation: RRLLRRLLRRLLRRL.
- Single Paradiddle:** 4/4 time signature. Rhythmic notation: RLRLRLRLRLRL.
- Double Paradiddle:** 3/4 time signature. Rhythmic notation: RLRLRLRLRLRL.

Figure 25: Overview of drum rudiments and paradiddles performed by all performers for drum performer recognition in 4.4 and drum stroke computing in Chapter 5

Each exercise was repeated for roughly three minutes, resulting in a total of 14,761 hits (7,353 left hand hits and 7,408 right hand hits). While the individual hits are labeled with a binary label (1 for left hand and 2 for right hand) and analyzed later in 5.4, in this section no differentiation is made between left and right hand hits. All data regardless of “hand” are labeled with a specific

¹⁴ The PAS is the world’s largest international percussion organization. More information on the PAS can be found at <http://www.pas.org/>

performer class, where each of the ten performers is assigned a class number from 1 to 10.

4.4.2 FEATURE EXTRACTION

Feature extraction follows the method proposed in 5.3. Rather than extracting features within a moving window as per the sitar performer recognition experiments, onset locations are predetermined by an onset detection function, and the audio and sensor data are windowed around each individual note onset. In this way the feature vector is calculated only once per event (strike), and only when an event is detected. Please refer to section 5.3 for more information about the onset detection algorithm and sensor preprocessing.

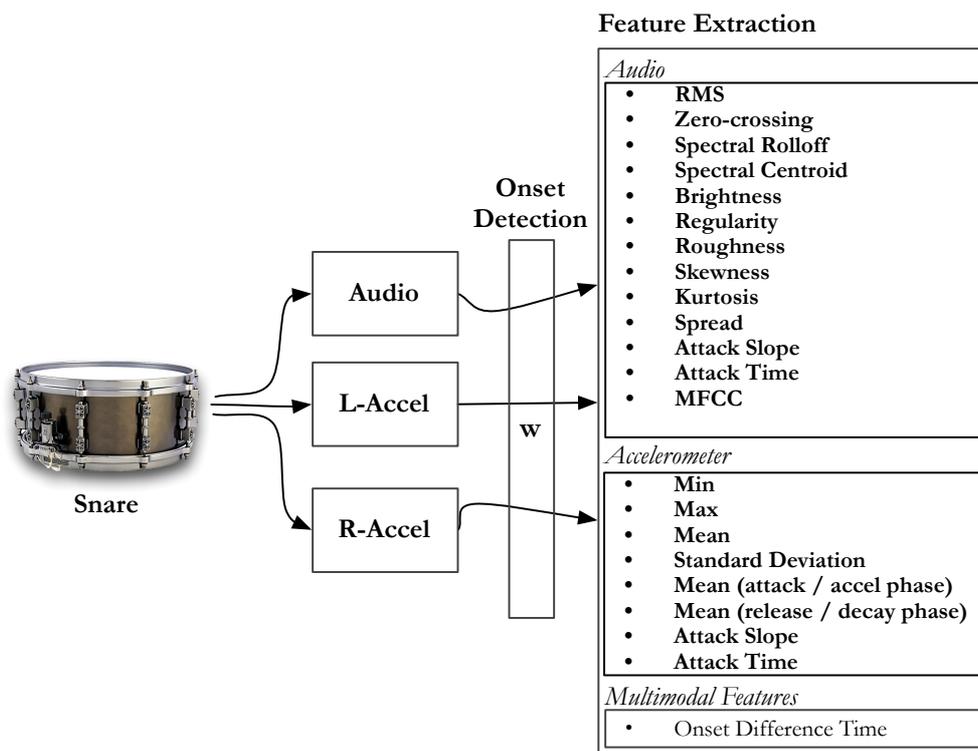


Figure 26: Overview of features extracted at each event in the data set

A 22-feature vector is collected at each drum event, consisting of thirteen audio features, eight accelerometer features, and one hybrid “multimodal” feature which looks at both audio and sensor data. Audio features included root-mean-squared (RMS), zero-crossing, spectral rolloff, spectral centroid, brightness,

regularity, roughness, skewness, kurtosis, spread, attack slope, attack time, and Mel-frequency cepstral coefficients (MFCC, 0). Accelerometer features included the minimum acceleration in the acceleration envelope, the maximum acceleration in the envelope, the average (mean) acceleration in the envelope, the standard deviation, the mean of the attack phase (positive acceleration), the mean of the release phase (deceleration), the attack slope of the attack phase, and finally, the attack time of the attack phase.

4.4.3 UNDERSTANDING DATA THROUGH MULTIMODAL VISUAL FEATURE CLUSTERING

Visual feature clustering is a valuable technique that can be used to understand complex relationships in data. By assigning individual features to the dimensions of a plot, e.g. plotting one feature's data on the x-axis of a scatter plot, and another on the y-axis, it is possible to observe and understand similarity and other relationships between the features. It is also possible to deduce other higher-level relationships and descriptors from the visual clustering of the features. In this research, features are plotted on two-dimensions, although higher dimension feature plots are possible and can expose more complex feature relationships.

The selection of audio and sensor features were initially motivated by specific acoustical and physical properties of snare drum performance. Visual feature clustering was used to refine the feature set, and to observe how well two-features clustered *the performers*. In general the more the individual performers independently cluster when plotting their features, the greater the features individually segregate the performers, which is the fundamental task of performer recognition.

Looking at the two feature plots in Figure 27 one will notice much more defined visual clustering on the right-hand plot, which shows the audio features' regularity plotted against spectral centroid. On the left hand side the regularity feature data are plotted against the roughness feature data. While in a recognition scenario one might favor the individual clustering of the performers, and the goal is to find relationships that segment the performers, overlap in clustering

can also show similarities between performers; thus, comparing the visualization with the machine learning trial results, as well as a priori knowledge and evaluation of the performers, can lead to many interesting conclusions.

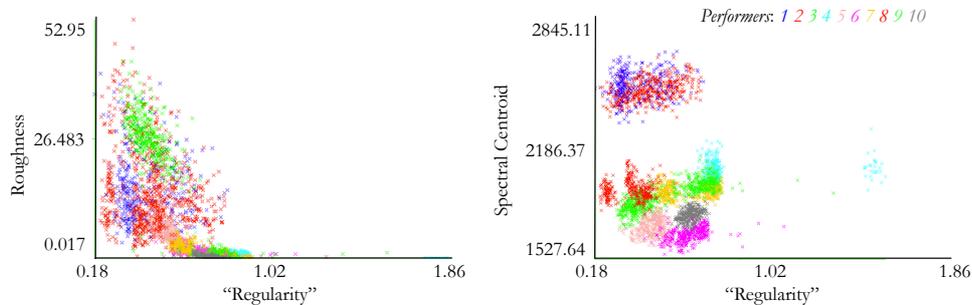


Figure 27: Feature scatter-plots of audio features *regularity* vs. *roughness* on the left and *regularity* vs. *spectral centroid* on the right

One observation from Figure 27 that reappears in the machine learning trials is the overlap between performer one (blue) and performer two (red). Observing this overlap one could see that the two performers' data were quite similar in terms of their regularity and spectral centroid features. By looking at feature cluster plots for performer one and two's other features, it is possible to observe a generalized notion of how the features distinguish the performers, and how the features relate to their performance.

Plotting feature data extracted from the audio recording against a feature data from the sensor-data can also prove to be a useful technique in evaluating the inter-connectivity of the acoustical and physical performance spaces. Figure 28 plots the audio feature *spectral rolloff* against the sensor feature *average release phase deceleration*; one can see noticeable clustering, and similarities between previous audio-only clustering characteristics. Again performer one (blue) and two (red) cluster and overlap with one another, although there is now greater separation between than two (compared to the audio only feature plots in Figure 27). Plotting multimodal features can lead to interesting observations about the physical and acoustical properties of the players' performance. For example, spectral and magnitude features from audio can be paired with the gestural features from the performer (attack acceleration and release deceleration, etc.), to find links between the physical actions and the acoustic output.

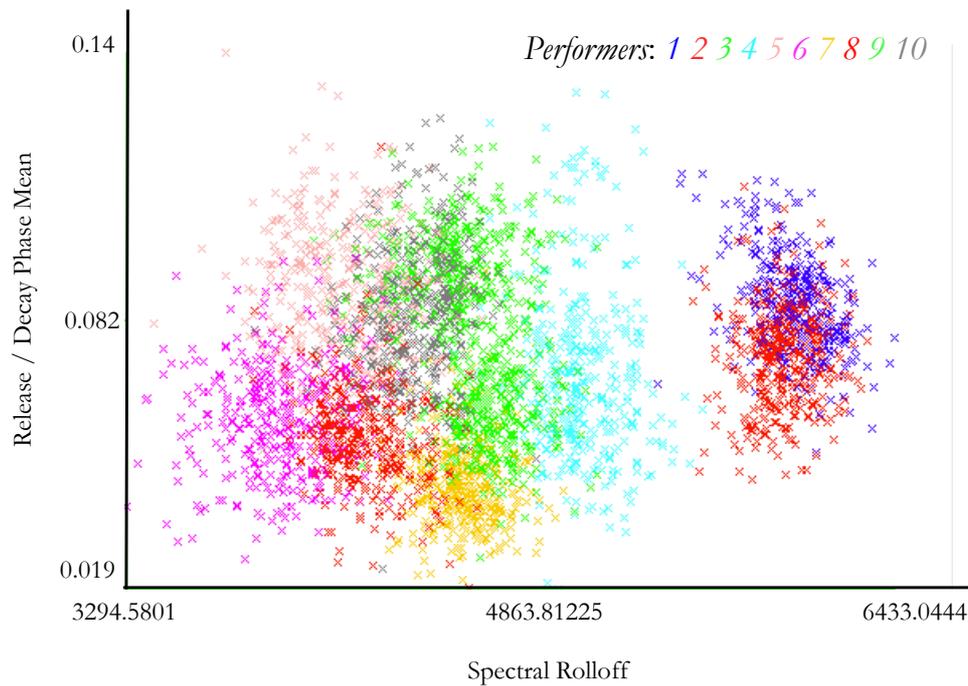


Figure 28: Feature scatter-plot of audio feature *spectral rolloff* vs. sensor feature *average (mean) release phase deceleration*

Many other relationships can be drawn from feature clustering. At a glance it is possible to get a generalized idea of how spread out the feature-pair data is for each performer. Evaluating the distance between the well-clustered performers can be as useful as looking at the general clustering and overlap. For example, in Figure 27 (right) performers 3, 5, and 6 all individually cluster independently, although performers 3 and 6 are furthest away from each other (when comparing these three specific performers).

These ideas and relationships are fundamental to understanding features, and the many complex relationships between data across multiple modalities. Similar techniques and visualizations will be used throughout the remainder of this dissertation, in both machine learning and other metric or feature-based contexts.

4.4.4 CLASSIFICATION

Four different classifiers were used in the drum performer recognition experiments. As per the sitar performer recognition experiments, these included a support vector machine trained using Sequential Minimal Optimization (SMO),

a multi-layer perceptron backpropagation artificial neural network (MLP), and Naive Bayes. Logistic Regression was also used in these experiments. Additional information about the classifiers and algorithms can be found in Appendix D.3 and (Witten, Frank, and Hall 2011).

4.4.5 RESULTS AND DISCUSSION

This section shows the performer recognition results for audio features only, sensor features only, and a combined multimodal feature space. 10-fold cross validation was used in all recognition tests. The player pool included ten players and so chance classification in all tests was 10%.

RESULTS: AUDIO FEATURES ONLY

As earlier in the sitar performer recognition trials, it is useful to investigate the classification accuracy of the audio features independently from the sensor features. In this way it is possible to evaluate the effectiveness of the individual modalities for the given classification task, how they can be improved (e.g. feature selection), and how well they perform multimodality.

Best classification performance was achieved using Multilayer Perceptron on data set D3, yielding 96.17% recognition. Combining all of the data sets into a large corpus however, adds more of the player's variability into the training/testing sets. Each data set increased in difficulty and may have influenced the player's performance; so, combining and testing all sets together can help reduce the homogeneity of the set, while allowing the classification algorithms to better generalize the performer's playing. In doing so Multilayer Perceptron achieved the best performer recognition results, yielding 92.58% recognition. Logistic regression was the second best classifier yielding 90.96% accuracy, followed by SMO (87.38%), and finally Naive Bayes (81.88%). Average performer recognition on all data sets using audio features only (and the ODT feature) achieved around 88% accuracy, showing promising results from analyzing the audio stream of the player alone.

Table 7: Performer recognition accuracy using audio features (and ODT feature) only

	Single Stroke (%)	Double Stroke (%)	Single Paradiddle (%)	Double Paradiddle (%)	All (%)
MLP	94.00	95.86	96.17	95.53	92.58
SMO	87.76	94.06	93.27	91.11	87.38
Naive Bayes	85.23	91.29	90.72	89.48	81.88
Logistic	90.98	95.78	95.79	95.53	90.96

RESULTS: SENSOR FEATURES ONLY

Investigating performer recognition on the sensor features only provides indication of the uniqueness of the performers' gesture data. Thus these tests serve to motivate this dissertation work in general, and one of its primary goals to investigate multimodal metrics and performance data. Again best performance was achieved when testing and training on a particular data set using Multilayer Perceptron (data set D2, 56.18%). Testing on all data sets, MLP yielded the highest recognition with 47.49%, 37.49% above chance. SMO returned second-best classification (39.86%), followed by logistic regression, and finally Naive Bayes.

Table 8: Performer recognition accuracy using sensor features only

	Single Stroke (%)	Double Stroke (%)	Single Paradiddle (%)	Double Paradiddle (%)	All (%)
MLP	54.95	56.18	52.46	49.42	47.49
SMO	43.03	44.09	44.61	39.92	39.86
Naive Bayes	37.75	37.45	39.43	31.14	32.12
Logistic	55.03	53.84	54.03	51.22	47.35

While classification results for sensor features only were not as high as audio only, it is revealing that useful performance data can be gained from exploring the data further. A lower recognition rate in performer recognition results for sensor features only does not whole-heartedly mean that the data is less useful than the audio features.

		Predicted Class										%
		<i>perf</i>	1	2	3	4	5	6	7	8	9	
Actual Class	1	636	167	226	112	43	171	5	30	48	12	0.439
	2	236	442	22	120	31	180	101	254	64	17	0.301
	3	123	0	1178	31	35	13	12	3	32	59	0.793
	4	64	100	29	753	222	48	47	98	75	39	0.511
	5	34	46	85	293	849	12	26	23	49	63	0.574
	6	124	175	29	70	13	512	219	202	150	29	0.336
	7	28	87	8	74	34	55	906	221	24	18	0.623
	8	29	184	3	76	18	84	212	844	29	3	0.57
	9	109	128	46	173	86	229	93	109	395	139	0.262
	10	117	64	114	115	249	74	43	12	153	495	0.345

Figure 29: Confusion matrix for all data sets and sensor features only using the MLP classifier

For example, the overlap between performers one and two in the clustering visualizations showed that in fact the performers were similar in terms of their performance. This is extremely useful information when analyzing a performers' metrics, abilities, style, etc. Similar observations can be illustrated by looking at the confusion matrix output of the classification task. The confusion matrix in Figure 29 summarizes the distribution of the classified instances (feature vectors) when classifying all data sets and sensor-only features using MLP. For example, class 1 (performer one) was recognized (classified) correctly as performer one 636 times, but misrecognized as performer two 167 times, performer three 226 times, and so forth. Investigating the confusion matrix, one can see which performers were "confused" for one another the most, and further investigate their sensor data to see how that relates to the physicality of their performance and gesture. For reference, the bold diagonal line in the matrix makes it easy to see the number of correctly identified instances for each performer (class). Specifically in the task of performer recognition it is the goal to reduce the amount of confusion or misclassification, thus, combining the confusion matrix with feature visualizations (e.g. clustering) can help re-evaluate and refine useful features for recognition.

RESULTS: ALL FEATURES

In this section both feature sets (audio and sensor) were combined into a large corpus to test the performance of the classification algorithms on all features together. Although performer recognition was already quite high on audio features alone, could multimodal relationships between audio and sensor features help improve results?

Table 9: Performer recognition accuracy using all features (audio & sensors) combined

	Single Stroke (%)	Double Stroke (%)	Single Paradiddl e (%)	Double Paradiddl e (%)	All (%)
MLP	95.86	97.52	97.34	96.73	94.21
SMO	93.24	96.57	94.95	93.98	90.59
Naive Bayes	90.22	94.28	93.32	90.92	86.18
Logistic	95.78	97.79	96.58	96.33	93.83

Again the two best classifiers were MLP and logistic regression. Best single data set recognition was achieved with logistic regression, achieving 97.79% recognition for data set D2.

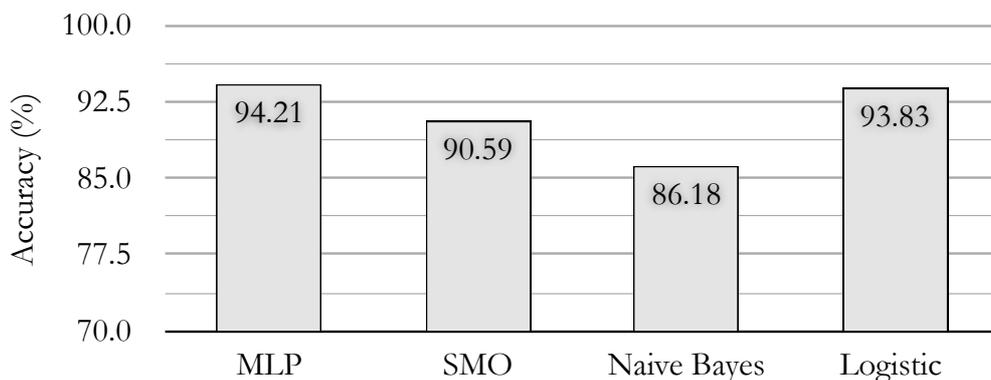


Figure 30: Performer recognition accuracy for all classifiers using all features and all data sets D1-D4

When combining all of the data sets and features together, Multilayer Perceptron yielded the highest recognition accuracy with 94.21% (Figure 30). This is about a 46.72% increase in recognition over the sensor features alone,

and an additional 2% increase over audio-feature only recognition. Comparing multimodal feature results to audio-only feature results, logistic regression increased by about 2.87%, SMO by 3.21%, and Naive Bayes by about 4.3%. Across the board performer recognition results were improved by combining both feature sets (audio and sensor) into a large multimodal feature vector.

In Figure 31, all of the classifiers results for all data in all sets instances (D1-D4) are averaged, for audio-only, sensor-only, and combined multimodal features independently. Averaging the classifiers' results in this way, the data set with the best results was D2 with 94.25% audio-only recognition, 47.89% sensor-only recognition, and 96.54% multimodal recognition. This figure presents a general overview of performer recognition accuracy as an average of all of the classifiers' results. Further averaging all data sets by modality (i.e. averaging D1-D4 results in Figure 31 for each modality) shows that using a sensor-only data set the classifiers can perform recognition on our pool of ten players with about 45.57% recognition accuracy; using an audio-only feature set can achieve about 91.77% recognition; and combining both into a multimodal feature achieves about 94.31% recognition.

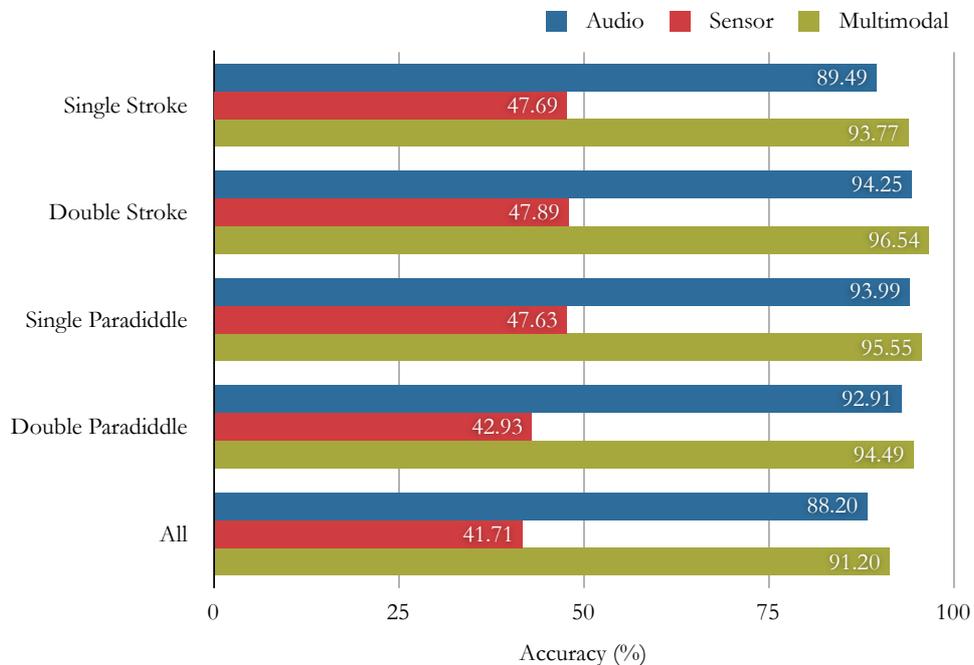


Figure 31: Accuracy for audio-only features vs. sensor features vs. multimodal features by averaging all classifiers

4.5 Discussion

This chapter explores multimodal performer recognition for both sitar players and drummers. While previous research into performer recognition has looked at high-level performance features extracted from audio analysis and symbolic data (from MIDI-enabled pianos), this research shows that it is also vital to explore the physical performance space through various sensors. As such we explore thumb pressure and fret-sensor data from the sitar performers, as well as gestural accelerometer data that captures the motion and trajectory of the drummers' hands while performing.

In exploring one of the overarching goals of this dissertation, which is to establish more meaningful communication channels between performer and machine, this chapter establishes a foundation showing that multimodality can help improve the computer's understanding of *who* the performer is. While this dissertation further analyses the actual performers' metrics elsewhere, this chapter shows that combining audio and sensor features into multimodal feature vectors, increased performer recognition accuracy (for both sitar players and drummers) across the board.

Improvements in recognition rates could be made in a number of ways for both experiments. One area to spend more time could be experimenting with additional features, especially for sensor data. Improved feature selection could help increase performance recognition both for single-modality recognition, but also when combined into the multimodal corpus. In the drum performer recognition, it could also be useful to experiment with resetting up the microphone a number of times during data collection (for each performer), to further minimize the effect of environmental and experimental conditions on the results. In the sitar recognition experiments, this is not necessary as the audio signal was directly from a pickup on the instrument that is not susceptible to such variables.

Most importantly, this chapter motivates the rest of this dissertation's work investigating the significant relationships existing both separately within individual modalities, and also in between modalities. This chapter's research into

performer recognition has also exposed performance features that are meaningful and help portray the characteristics of the actual performance.

The remainder of this dissertation in whole is particularly interested in evaluating the relationships that exist between the acoustical and physical dimensions of musical performance, however, performer recognition in itself possesses many possible functionalities and uses. In the post-desktop world of “ubiquitous computing”, performer recognition can make the computer more intelligently aware of its users. One example could be tailoring feedback to individuals in a group without having to switch between user profiles and setups that can be cumbersome in a multi-user situation. In music schools, this could be useful for group or ensemble exercises and practice. Another use case could be specifically using high-level “style” features for classification, and comparing the stylistic elements that distinguish multiple performers’ interpretations of a musical piece.

Chapter 5

Drum-Stroke Computing

Can multimodal training give an audio recording eyes?

This chapter revisits the drum data used in the drum performer recognition trials to analyze various aspects of drum performance. These include automatic labeling of audio strikes using surrogate data training, automatic drum-hand recognition, and investigations into multimodal drum performance metrics.

5.1 Background and Motivation

Combining machine learning techniques with percussive musical interface and instrument design is an emerging area of research that has seen many applications in recent years. Tindale investigated drum timbre recognition (Tindale et al. 2004) and later applied similar techniques to turn regular drum triggers into expressive controllers for physical models (Tindale 2007). Other examples have been proposed which even enable human-machine interaction with mechanical percussionists who can listen to human performers and improvise in real time (Weinberg and Driscoll 2006). In the previous chapter, we investigated automatic performer recognition of ten drummers.

In terms of signal processing, there are now robust onset detection algorithms for percussive performance (Bello et al. 2005; Dixon 2006) enabling accurate identification of when musical events occur. Researchers (including in this dissertation) have also been actively investigating other areas of musical performance such as tempo estimation (Dahl 2005; Gillet and Richard 2008), beat tracking (Dixon, Simon 2007; Goto and Muraoka 1999), and percussive instrument segmentation (Goto and Muraoka 1994). Combining many of these

techniques together, researchers have explored the task of automatic transcription of drum and percussive performance, (Fitzgerald 2004; Gillet and Richard 2008; Paulus and Klapuri 2003; Tzanetakis, Kapur, and McWalter 2005). Great advances have been made in the aforementioned tasks; however, the majority of research into drum interaction scenarios which combine musical interfaces/instruments and machine learning have been concerned with the segmentation or isolation of individual drums from a recorded audio signal. While mono and polyphonic drum segmentation is a key aspect to tasks such as automatic drum transcription, a vital feature of drum performance (that we've yet to see explored in current drum analysis literature) pertains to the physical space of drum performance. Not only is it beneficial to know when and which drum is played in a pattern, but also *which hand* is striking the drum. This research investigates this question, demonstrating a multimodal signal processing system for the automatic labeling and classification of left and right hand drum strikes from a monophonic audio source.

There are many real-world cases where drum stroke recognition is useful. In fact, traditional exercises which practicing drummers study emphasize the practice of specific left and right hand patterns. A key element in automatic transcription scenarios that has been missing up until now is transcribing which hand performed a drum hit. In order to understand ones performance fully, it is important to know how the player moves around the drum(s), the nuances and differences present in the strikes of their hands (independently), and the possible stylistic signifiers resulting from the physical aspects of their individual hand strikes. This presents a large problem, as it is nearly impossible to determine which hand is hitting a drum from a monophonic audio recording alone.

Using direct sensors such as accelerometers on the performer's hands, however, it is possible to capture exceptionally accurate information about the movements of the performer's hands. This comes at the cost of being invasive and possibly hindering performance. In a typical controlled machine-learning situation it is of course possible to place constraints on the data-capturing scenario. One solution would be to only record left hand strikes, and then separately record right hand strikes, labeling them accordingly when performing feature extraction. A primary goal of this research however is to not only capture

each hand playing in isolation, but in context of actual performance and practice scenarios. As such, the interplay between left and right hand playing is of utmost importance. Another option would be to manually label each audio event as being either from the left or right hand, based on a priori knowledge of a specific pattern played. As many data capturing scenarios (including ones in the research) involve specific patterns to be played, this is a common but time-consuming approach to labeling drum training performance data. Additionally this approach is blind to inevitable playing errors in the performance, which require manual adjustment when labeling the training data. We are also interested in investigating the improvisatory elements of drum performance, making the task of manually labeling hand-patterns nearly impossible. To overcome these challenges, this research turns to an exciting new technique inspired from Surrogate Sensing (Tindale, Kapur, and Tzanetakis 2011) to enable the automatic labeling of drum hand patterns for classification.

One of the earliest studies of drum performance showed how physical factors such as the start height of a stick could impact the resulting amplitudes and durations of the sound produced (Henzie 1960). More recently, Dahl showed similar relationships between the correlation of strike velocity and the height and shape of the stroke in user studies (Dahl 2005). Dolhansky et al. modeled the shape of a percussive stroke to turn mobile phones with accelerometers into physically-inspired percussive instruments (Dolhansky, Mcpherson, and Youngmoo 2011). There are many ways which people have attempted to analyze the gesture of drum performance and its effect on the dynamic and timbre spaces; Tindale et al. provides a good overview of sensor capturing methodologies in (Tindale et al. 2005). The research mentioned and other countless examples confirm the strong link between the physical space in which a performer's actions exist, and the fingerprint imparted on the musical output. To this end we begin to investigate these ties in this chapter by not only looking at drum-hand recognition, but also at statistical measures observable by multimodal analysis of acoustical instrument (drum) output paired with sensor systems.

The remainder of this chapter is as follows: in section 5.2 an overview of the data collected is detailed, followed by an overview of the analysis framework

(including the implementation of surrogate data training for automatic hand labeling of training data) in 5.3. Drum hand recognition results are presented in section 5.4, and multimodal drum performance metrics in section 5.5. Finally, a discussion and conclusion are provided in section 5.6.

5.2 Data Collection

In this section the data capturing and analysis system used in the drum-stroke recognition experiments is described. From a high-level view, the drum-hand recognition experiment employs a three-step process including a data collection phase, an analysis phase, and finally the testing and machine-learning phase as illustrated in Figure 32.

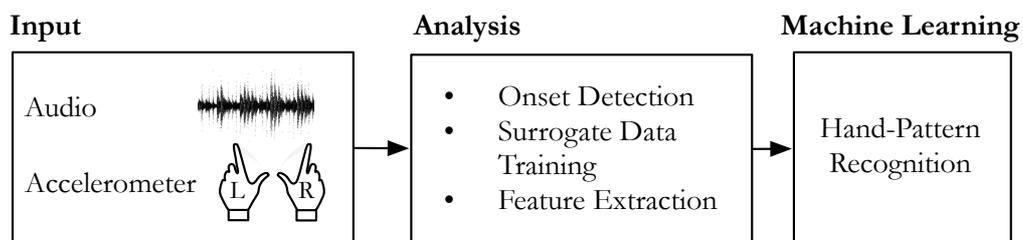


Figure 32: Overview of drum hand recognition system

The performance data used previously for drum performer recognition was used in these experiments. Ten performers played four fundamental drum exercises for roughly three minutes each, which resulted in a total of 14,761 hits (7,353 left hand and 7,408 right hand). Please refer to section 4.4.1 for more information on the specific exercises and data collected for these experiments.

5.3 Analysis Framework

The analysis framework is comprised of two main steps, surrogate data training and onset detection / peak picking. The following sections describe in detail the data analysis and signal processing used to label and train the system from drum hand identification.

5.3.1 SURROGATE DATA TRAINING

One of the biggest hurdles for musical supervised machine learning is obtaining and labeling a large enough training data set for true results. As described earlier in the introduction, manually labeling the training data is not an efficient process, nor does it easily deal with errors that are common in the data collection process. By using a technique that can automatically label training data, the training regimen can be more loosely defined, even allowing the performer to improvise (unless there was specific desire to record particular patterns as in our case). Common disturbances in the data collection process such as performance mistakes, which normally must be accounted for by the researchers manually are also no longer an issue. We turn to a new technique inspired by Surrogate Sensors (Tindale, Kapur, and Tzanetakis 2011) enabling us to quickly record and label each hit in the audio recordings by using known information from direct sensors (accelerometers) to navigate unknown information in the data from our indirect sensor (microphone). The direct sensors provide the benefit of near perfect accuracy making the technique extremely robust (see Table 10). The method is also transferable to other sensors and modalities, and the particular implementation in this research is described in the following section on onset detection.

5.3.2 ONSET DETECTION

A triple-axis accelerometer was placed on each of the performer's hands while recording the data sets. The ultimate goal was to use gesture onsets from the independent hands' accelerometers to navigate and label the note onsets in the audio streams. As shown in Figure 33, each axis (per accelerometer) is first preprocessed in Matlab by removing the DC offset and full-wave rectification. The accelerometers each have their three axes summed and averaged to collapse the data streams into a single dimension. Next jerk¹⁵ is calculated for each accelerometer, followed by a threshold function to remove spurious jitter. To

¹⁵ "Jerk" is the derivative of acceleration

further smooth the signals before onset detection is applied, the envelopes of the signals are extracted, and smoothed with a low pass filter. The onset curve is then calculated and peak-picked at local maxima. Lastly onset detection was also performed on the audio recording, and all three streams' (one audio, two accelerometer) onset locations (in seconds) are stored in independent vectors. More detailed information on the onset detection algorithm can be found in (Lartillot, Olivier et al. 2008).

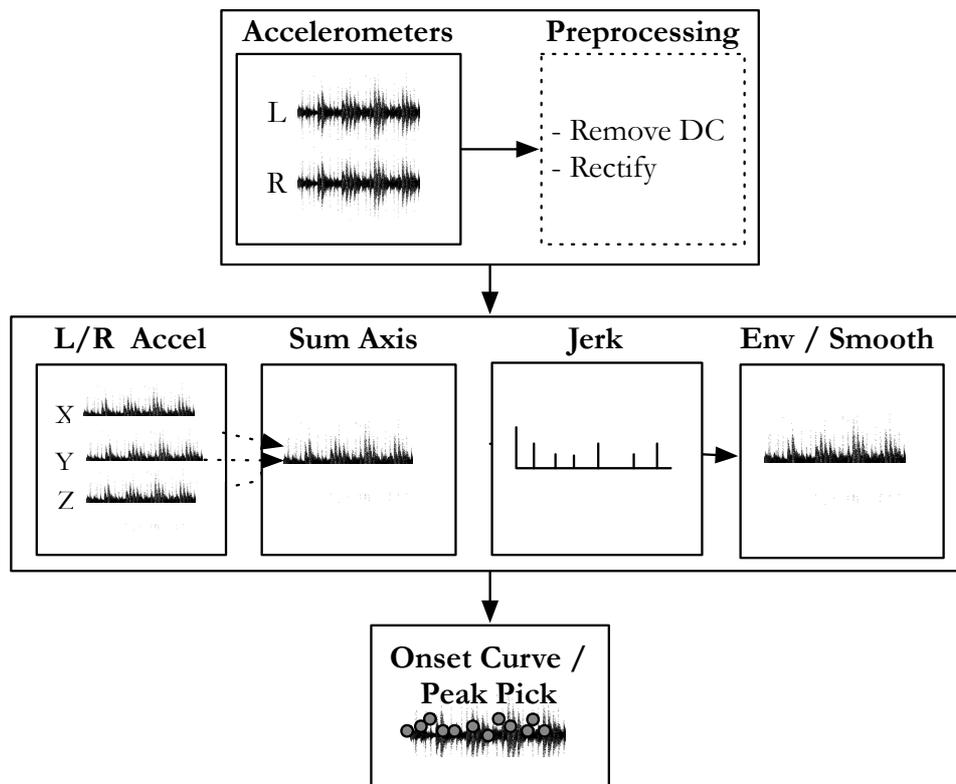


Figure 33: Overview of onset detection algorithm

ONSET DETECTION ACCURACY

Onset detection accuracy of the accelerometers (for performers 1 and 2) is shown in Table 10 using standard measurements called *precision*, *recall*, and *f-measure*. Generally speaking, these measurements give a sense of the accuracy and quality of the results; measuring how many true positives were returned, false

positives, etc. For an in depth explanation of the measurements, please refer to section 6.4.4.

The high yield (99%) in accuracy of the accelerometers makes them a great candidate for surrogate labeling the audio onsets as either left or right hand onsets. The onset vectors were also exported as .txt files and imported into a beat-tracking application called BeatRoot (Dixon, Simon 2007) to visualize and correct any errors in the accelerometer onsets detected. It should be noted that although correction was performed, the correction step was not necessary, as the minor number of falsely detected onsets would not impact the data very much. However, we desired 100% ground truth and so any false-positive and false-negative onsets were manually corrected in BeatRoot prior to feature extraction.

Table 10: Accelerometer onset detection accuracy for performers 1 and 2

	Precision (L/R)	Recall (L/R)	F-Measure (L/R)
Performer 1	0.997	0.997	0.997
Performer 2	0.993	0.997	0.995

5.3.3 FEATURE EXTRACTION

After onset detection, features were extracted in Matlab by taking the accelerometer onset positions for each hand and searching for the nearest detected onset (within a certain threshold determined by the frequency and tempo of the strikes) in the audio onsets. The strike in the audio file was then windowed to contain the entire single-hit and various features were extracted over the windowed segment. The feature vector was labeled with the appropriate class (1.0000 = Right, 2.0000 = Left) and exported as an .arff file for machine learning analysis in Weka. For each strike a 14-dimension feature vector was calculated from the audio-onset containing: RMS, Spectral Rolloff, Spectral Centroid, Brightness, Regularity, Roughness, Skewness, Kurtosis, Spread, Attack Slope, Attack Time, Zero Crossing, MFCC (0th coeff.), and the Onset Difference

Time (ODT¹⁶) between the detected audio and corresponding accelerometer onsets.

5.4 Drum Hand Recognition

Once the data was collected it was imported into Weka for supervised learning. The primary focus of this experiment was to investigate if a machine could be trained to reliably classify which hand was used to strike a snare drum.

5.4.1 CLASSIFICATION

Five classifiers were used in the tests including a Multilayer Perceptron back-propagation artificial neural network (MLP), the J48 decision tree classifier, Naive Bayes, a support vector machine trained using Sequential Minimal Optimization (SMO), and Logistic Regression. 10-Fold cross validation was used in all tests and the entire 14-dimension feature vector was utilized during testing.

5.4.2 RESULTS: ABOUT THE TESTS

The following results sections investigate drum-hand recognition in various capacities. Individual and combined recognition results are examined in 5.4.3, the effects of training and testing on different data sets in 5.4.4, and cross-performer training and testing in 5.4.5. As this is a binary classification scenario (classification can either be left or right hand), the chance classification baseline for all results is 50%.

5.4.3 RESULTS: TEST ONE – ALL DATA (INDIVIDUAL VS. COMBINED SCORES)

Using the entire data set and 10-fold cross validation, the best results for all performers were achieved using both the multilayer perceptron (MLP) and logistic regression (Logistic) classifiers. Classification results and test size

¹⁶ The Onset Difference Time (ODT) is a feature / metric that describes the difference time between the onsets detected in an acoustic signal, and a signal derived from a sensor in another modality.

information for each performer is provided in Table 11. In general, all of the algorithms appear to do a decent job at generalizing over the entire data set and provide similar classification results with smaller subsets of the feature vector.

The best single-performer drum-hand recognition results were achieved for performer #8 using multilayer perceptron (97.64%), followed by performer #5 (if ignoring performer #8's logistic regression classifier) using logistic regression (96.42%). Interestingly performer #8 was one of the most advanced percussion players from the test group whereas performer #5 happened to be at a beginner level. Achieving high classification accuracy across these two performers demonstrates that the feature vector may generalize across skill level quite well (not favoring a particular skill level or consistency over another), making the technique robust and applicable to the entire range of performers.

Table 11: L/R Drum hand recognition accuracy for all performers and data

<i>Perf</i> #	MLP (%)	SMO (%)	Naive Bayes (%)	Logistic (%)	J48 (%)	# L Hits	# R Hits	Total Hits
1	84.21	81.10	64.97	83.86	79.38	724	726	1450
2	84.25	81.87	75.12	84.32	81.12	731	736	1467
3	78.47	76.99	65.01	77.66	71.40	747	739	1486
4	59.59	55.86	54.85	61.90	57.56	739	736	1475
5	95.95	90.34	82.64	96.42	88.92	741	739	1480
6	87.59	80.83	72.55	84.11	78.20	756	767	1523
7	88.87	87.22	82.82	91.48	86.60	731	724	1455
8	97.64	92.98	84.21	96.63	93.25	748	734	1482
9	79.69	72.00	67.02	73.79	73.99	749	758	1507
10	89.00	79.67	79.81	86.77	80.85	687	749	1436
All	76.01	60.19	53.82	61.26	75.40	7353	7408	14761

Accuracy often achieved over 95% for a single performer, and the average classification accuracy for each classifier (across all performers) from best to worst was MLP (84.53%), logistic regression (83.69%), SMO (79.89%), J48 (79.13%), and finally Naive Bayes (79.90%), as can be seen in Figure 34.

As expected combining the feature vectors from all performers into one large corpus of 14,761 left and right hand drum hits achieved slightly lower classification results. When combined multilayer perceptron achieved the best results yielding 76.01% accuracy. Second best recognition performance was

achieved with the J48 classifier, yielding 75.4%. The recognition achieved when combining the feature vectors from all performers into a single data set showed promising results in the performance generalization across multiple performers and skill levels. A performer may have never directly trained the system and yet satisfactory results are still achieved, as investigated further in 5.4.5.

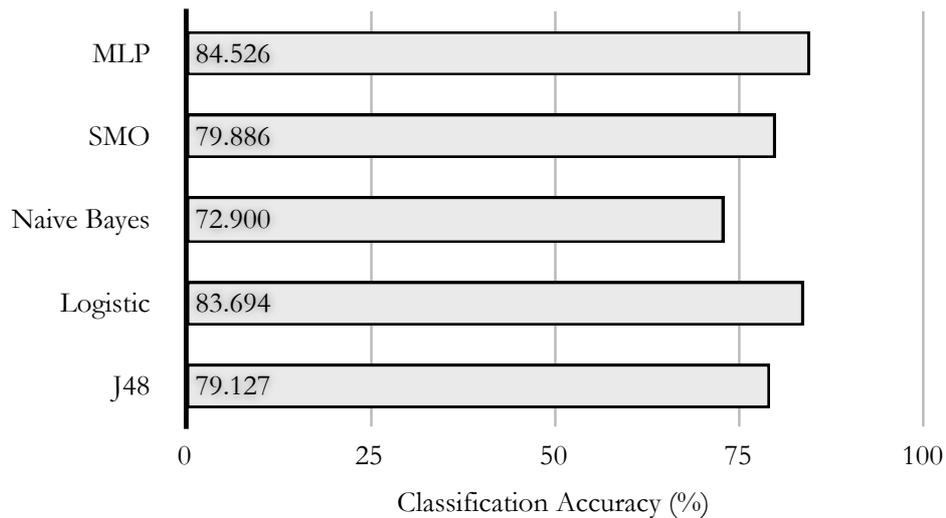


Figure 34: Average drum-hand recognition accuracy (%) across all performers for each classifier

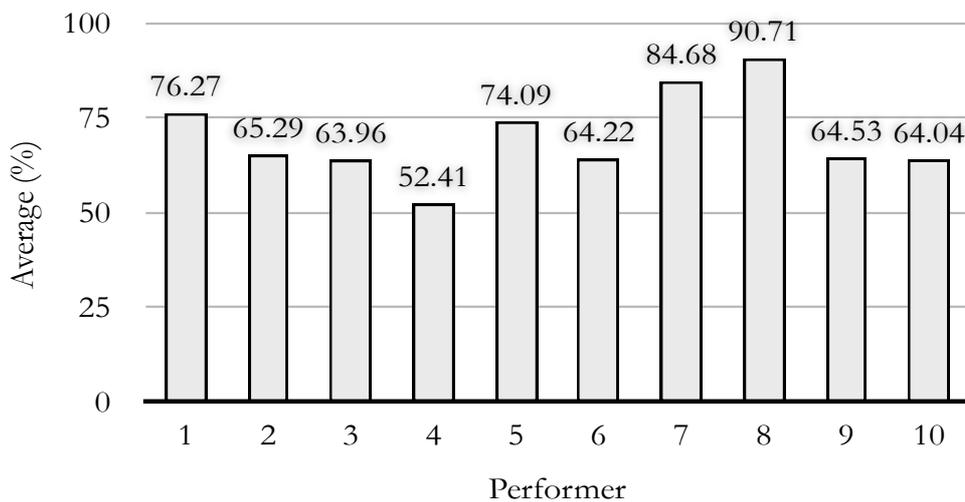


Figure 35: Average classification of all classifiers for each performer

To see which performer had the most reliable left/right hand classification accuracy overall, the results from all classifiers were averaged and are displayed in

Figure 35. Overall, performers #7 and #8 had the best classification accuracy when averaging all five classifiers, with 84.68% and 90.71% classification (respectively).

5.4.4 RESULTS: TEST TWO – DATA SPLIT

In this test the data was split into two partitions. The first partition contained the feature vectors from first two rudiments (D1 and D2) and was used for training, while the data from the second two rudiments (D3 and D4) was used for testing.

As the level of complexity in the patterns played increased with each rudiment, training on the earlier rudiments and testing on the latter can perhaps provide insight into how well the classification generalizes across the range of the players' performance, having only seen a restricted context of their actual performance. This may also provide insight into how consistent the players may have performed as a function of time, possibly connected to the experience or skill of the individual performers.

Table 12 shows the classification results of this test and highlights the following. Firstly certain classifiers such as Naive Bayes and sometimes J48 seemed to have a difficult time generalizing without seeing the entire data set while the neural net (MLP), SMO, and Logistic Regression seemed to perform reasonably well. In certain cases this caused classification results to drop as much as 20% (e.g. performer #5, Naive Bayes, split set which dropped from 82.64% down to 62.01%). Interestingly Naive Bayes seems to perform all right on the split training/testing sets for certain performers. For example, performer one was a more advanced player than performer two, which may suggest a higher consistency over the entirety of the performances. This may account for better classification for probabilistic classifiers such as Naive Bayes, which rely on the independent variances of class variables vs. the entire covariance of the feature space.

As the amount of training and testing data is roughly halved when testing on the split sets, further investigations in how training size might affect classification was explored. To do this, training and testing was performed on half the data (D1 and D2), which yielded similar classification accuracy as testing and training

on the entire data set (which also randomly reorders the data when it is tested over k folds).

Table 12: Classification accuracy using separate rudiments for training and testing

<i>Perf</i> #	MLP (%)	SMO (%)	Naive Bayes (%)	Logistic (%)	J48 (%)
1	79.56	80.11	67.17	81.06	73.43
2	76.53	63.98	57.03	66.03	62.89
3	69.06	64.28	55.38	71.18	59.89
4	48.92	54.723	51.22	52.84	54.32
5	77.05	78.79	62.01	85.10	67.52
6	61.34	67.10	61.07	59.76	71.82
7	84.32	87.16	82.57	87.43	81.89
8	95.57	89.66	85.64	94.63	88.05
9	62.25	68.08	64.64	70.99	56.69
10	62.36	63.62	67.28	65.87	61.10

Training and testing with D1 and D2 for players one and two for example yielded 70.16% accuracy for Naive Bayes and 78.88% with MLP. While it is clear that some classifiers seem to generalize quite well in all cases, more simple probabilistic classifiers such as Naive Bayes seem to benefit greatly from having a larger training set that covers a wider variance in feature data.

5.4.5 RESULTS: TEST THREE – LEAVE ONE (PERFORMER) OUT

The initial testing on individual performers and the entire corpus of data yielded satisfactory results for all player levels (beginner to advanced) individually, as well as the entire player pool together. This test was designed to further investigate the generalization of the data by leaving one performer out during training, and then testing against that performer’s data set during testing. How would the system respond to a user that it had never seen before? The two performers chosen for the “leave one performer out” experiment included performer #7 and performer #8, and were chosen as they scored the highest average classification when averaging all classifiers (Figure 35). In this way the two best performers results can be tested with and without their data in the actual training set.

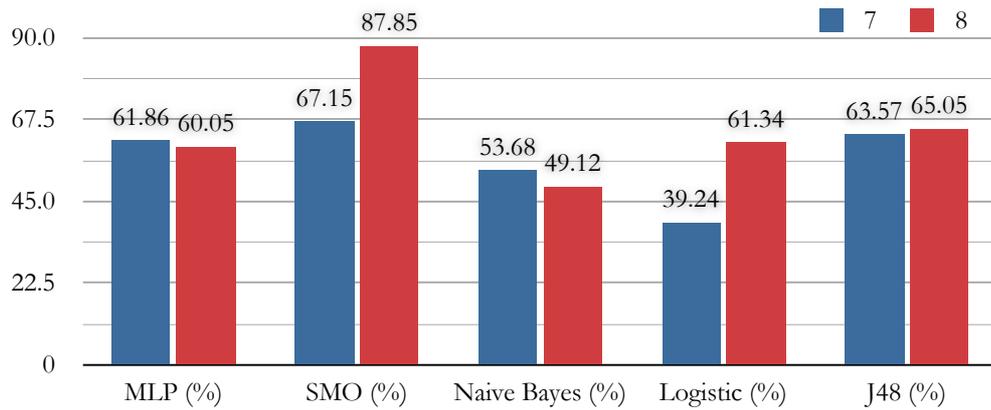


Figure 36: Classification results for the two best performing players when trained on all *other* data sets

As expected the classification results reduce in accuracy when testing without training on a performer’s actual data. The average classification in this scenario might need to be increased for robust real-time use, however the classification results are encouraging. In fact the highest classification results were achieved with the SMO classifier and still classified left and right hand hits for the “unknown” performer with nearly 90% confidence. Naive Bayes performed the poorest all around, similar to the data split test, barely achieving classification results above chance. Other classifiers normally achieved classification results above or around 60% - 65%. Nonetheless the results are encouraging and future results could be improved by extracting other useful features. With the current feature set the system would merely require the user to train and add their own data to a large corpus of all trained performers, or selecting a user profile that is trained on their own data only.

5.5 Drum Performance Metrics

Automatic drum hand recognition proposes exciting new possibilities including: more nuanced automatic drum transcription, preservation of performance technique from master musicians long after life, providing new controller data for live performance, and providing insightful information and metrics during regimented practice and musical training. However, the information from direct sensors can also be used in conjunction with indirect sensors to provide new

angles in statistical performance metrics and features. This section begins to look at some of these features, and how they may increase the ability to describe and deduce meaningful information from drum performance.

5.5.1 CROSS-MODAL ONSET DIFFERENCE TIME (ODT)

In traditional drum performance analysis, temporal information such as timing deviations and onsets of drum hits are normally investigated by analyzing an audio recording. Researchers have not only investigated the physical onset times (in audio), but have also looked at the perceptual onset and attack times (“PAT”) in order to measure when sounds are actually heard (Dolhansky, Mcpherson, and Youngmoo 2011). Here we consider the physical onset times from sensors on the actual performer in relation to the onset times determined from the acoustical output in what we call the Onset Difference Time, or ODT.

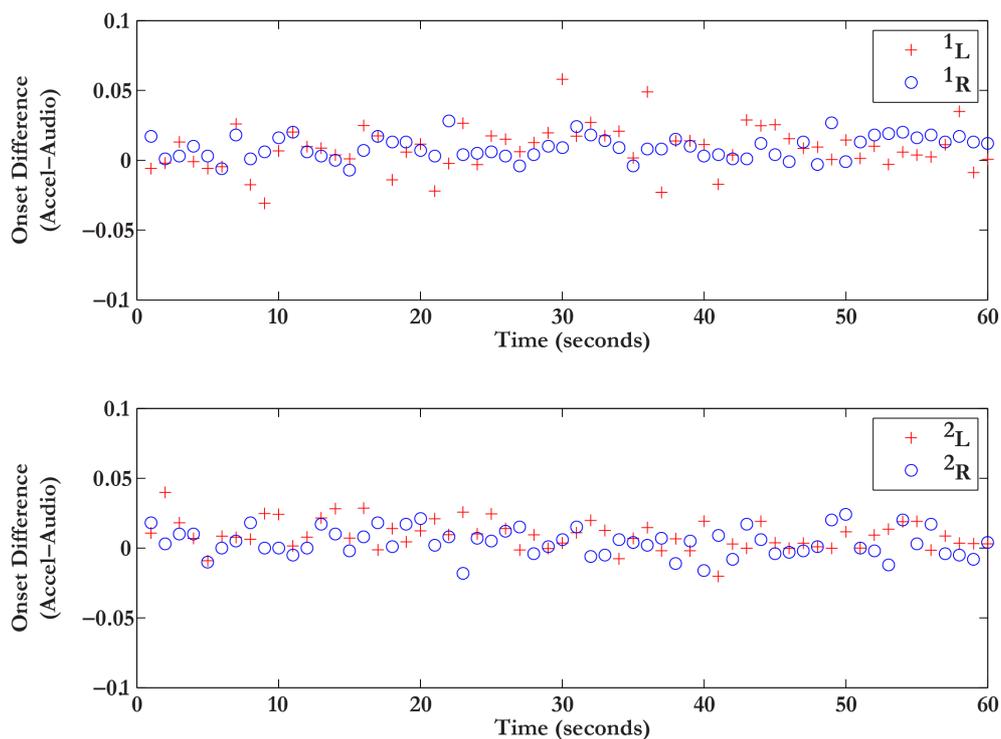


Figure 37: Onset difference times for the 60-sec. of D1 (performer one top, performer two bottom)

Figure 37 shows the onset difference times (in seconds) between the left (+, red) and right (o, blue) hand accelerometer onsets and their corresponding audio

onset times. A horizontal line at the center (0 on the y-axis) would mean a perfect match (zero difference) in onset times, and an observation of the graphs shows that performer two (bottom) had a generally lower onset differentiation than performer one (top). This observation is reaffirmed by player two's lower mean and range statistics shown in Table 13. Performer two was in fact a more highly experienced drummer, suggesting a great link in physical vs. acoustical onsets in this particular exercise. Observing Figure 37 it is also apparent that in this 60-second pass of D1, the onset difference times of performer two's individual hands were more closely related (in terms of mean onset difference) than that of performer one's. The following sections will begin to explore a statistical measure derived from the performers' ODT. Throughout, Table 13 and Figure 38 will be used to compare the various metrics.

Table 13: Average onset difference statistics for both performers

Data Set	Min (rush)	Max (lag)	Mean	Std. Dev.	Range
Performer 1	-0.0076	0.0148	0.0118	0.0116	0.0224
Performer 2	-0.0070	0.0149	0.0107	0.0133	0.0219
Performer 3	-0.0176	0.0164	0.0064	0.0286	0.034
Performer 4	-0.0230	0.0228	-0.0082	0.0348	0.0458
Performer 5	-0.0274	0.0186	-0.0203	0.0311	0.046
Performer 6	-0.0286	0.0283	0.0128	0.0375	0.0569
Performer 7	-0.0358	0.0205	-0.0205	0.0450	0.0563
Performer 8	-0.0216	0.0289	0.0011	0.0772	0.0505
Performer 9	-0.0468	0.0381	0.0098	0.0723	0.0849
Performer 10	-0.0383	0.0357	0.0056	0.0522	0.074

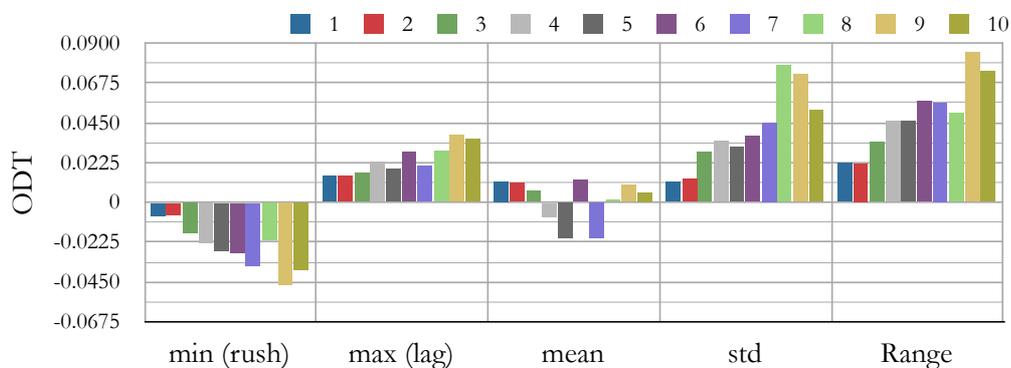


Figure 38: Bar graph visualizing average onset difference time metrics (Table 13 - rush, lag, mean, standard deviation, and range) for all ten performers, in seconds

GETTING AHEAD OF THE BEAT: RUSH METRIC

Table 13 and Figure 38 show averages from both hands of all performers and all data sets D1-D4. Min, or “rush”, is calculated as an average of the instances where the accelerometer onsets happened to come earlier than the audio onsets. As such it is determined by negative onset difference times. Rush is an interesting metric as the performer’s physical action reached its maximum acceleration before the strike’s acoustic onset. This was observed to be the case when the performer’s strike reached maximum velocity before slightly releasing the stick and transferring the motion to the drum.

On average, performers one and two had the smallest rush. Additionally, when performer two’s physical onsets rushed the audio onsets, it did so slightly less than performer one. Again this may be attributed to the fact that performer two was a faintly more experienced player, exhibiting tighter timing than performer one. Performers three and eight were two of the most advanced players, and also exhibited tight rush values among the group of performers.

Performer nine had the greatest average rush, meaning that when the performer’s physical onset rushed the audio onset, it did so more drastically than the other performers. Not surprisingly, performer 9 was one of the beginner level drummers in the group, which is further exposed in some of the other statistical measures.

GETTING BEHIND THE BEAT: LAG METRIC

Max or lag is calculated as an average of the instances where the accelerometer onsets were later than their respective audio onsets (positive onset difference times). In the physical world this could be the result of the performer continuing their strike gesture after initial contact with the drum. When compared to rush, there is less deviation lag times across all performers (Figure 38).

Performers one and two averaged almost identical lag times, further exhibiting their similarity in performance. Additionally, they had the least amount of lag, resulting in the tightest range (difference between lag and rush), or tightest timing in the player pool. Similarly as with the rush metric, performer three (who

was one of the most advanced performers) achieved an average lag time that was generally closer to the ODT (compared to the other performers) when he played behind the beat.

Performer nine had the greatest amount of lag, again exposing the fact that performer nine was at a beginner level, and exhibited greater elasticity or deviation from the target ODT (zero).

PUTTING IT ALL TOGETHER: MEAN, STANDARD DEVIATION, AND RANGE METRICS

Mean is determined as the average onset difference time calculated over the entire vector of ODTs for each performer. Comparing performers one and two again, performer two performed with slightly less distance between physical and audio onset times. Interestingly, performer two's standard deviation was slightly larger than performer one's. Essentially this means that the amount of dispersion from the performer's mean performance was greater. That being said, both performers achieved very similar timing characteristics across the board.

When analyzing rush and lag, performer nine generally performed the poorest in terms of having the greatest amount of rush and lag. Looking at the performer's average ODT times however, performer nine achieved an average ODT that appears to be very close to that of performers one and two. This stresses the importance of looking at metrics beyond the average value, in particular the rush (min), lag (max), standard deviation, and range. In fact, looking at both standard deviation and range, it becomes apparent that performer nine deviated from their mean performance significantly greater than the other performers. Performers one, two, and three achieved the smallest standard deviation and range values, providing insight into the precision and accuracy of their performances. These standard statistical measures are extremely useful in analyzing musical analysis, as will be seen throughout the remainder of this dissertation.

FREQUENCY AND OTHER OBSERVATIONS

The average (mean) onset difference time gives a general sense of the performer's physical tendencies. On average, was the performer's ODT behind or ahead of the (acoustic) beat? A positive average time would mean that the performer's ODT generally lagged behind the beat, true for the majority of the performer's (one, two, three, six, eight, nine, and ten) as illustrated in the previous section. However, it is also useful to know what percentage of strikes had rushed ODTs and what percentage had lagged onset difference times, to know a bit more about the distribution of strike onset differences.

Table 14: Rush/Lag distribution of performer ODTs

Data Set	% Rush	% Lag
Performer 1	13	87
Performer 2	18	82
Performer 3	30	70
Performer 4	68	32
Performer 5	84	16
Performer 6	27	73
Performer 7	73	27
Performer 8	55	45
Performer 9	33	67
Performer 10	41	59

As seen in Table 14, the percentage split follows the average ODT times accordingly. The more weighted a performer is to either side (greater percentage of strikes being either rush or lag), the more consistent the performer was in the physical motion of their strike. Under this criteria, performers one, two, and five were the three most consistent, with performer eight having the widest range of strikes falling on either side of the acoustic onset. Further investigation into the distribution could be useful in further research. For example, it is possible that the fairly even distribution in performer eight could be the result of human compensation of late and early beats. Other comparisons between the physical dimensions and the onset difference time could be useful to compare in terms of the relationship to the audible note onsets.

5.6 Discussion

This chapter investigated two ways in which multimodal signal processing and sensor systems can benefit percussive computation. In the first case, direct sensors (accelerometers) on a performer were used to automatically annotate and train the computer to perform drum hand recognition from indirect sensors (a single microphone). Classification results show that it is possible for the computer to identify whether a performer hit a drum with their left or right hand, and will be able to benefit future musical interaction in a number of ways. For example, once trained with the direct sensors, the machine can non-invasively transcribe the physical attributes of a percussionist's performance (with independence between hands), adding important detail to future automatic music transcription scenarios. Additionally automatic drum-hand recognition could be useful in many pedagogical scenarios such as rudiment identification, accuracy and other performance metrics scenarios. In live performance contexts where it may be desired to trigger musical events, processes, live visualizations based on particular sequences of strikes, and score following, drum stroke recognition using non-invasive methods can also be extremely powerful.

The second case looked at new performance metrics obtainable using a multimodal system. The research findings, synergistically utilizing data from direct and indirect sensors such as accelerometers and microphones (respectively) reconfirms the author's notion that it is important to look at both the acoustical and physical domains (simultaneously) when investigating musical performance. Research often chooses one or the other for analysis, however investigating the space in between possesses great potential.

At the core of much of this is the trade-off between direct and indirect sensors. Indirect sensors such as microphones have proven to be extremely useful and reliable sources for music information retrieval, with the added benefit of not hindering performance. At the same time they lack certain physical attributes that are only possible to obtain by placing more invasive direct sensors on the performer, and/or instrument. In one sense this research hopes to bring wider attention to the novel technique called surrogate sensing which reduces the negative impact of invasive sensors by constraining their application to the

training phase of a musical system or experiment. At the same time there is a lot of work ahead and the future definitely still holds an important space for direct sensors in these scenarios. There is the very real possibility of a future where direct sensors such as accelerometers are small and light-weight enough to be embedded within a drum stick without altering performance in any way; but also one where a trained machine can play back a recording from great musicians of the past and automatically transcribe the magical expressivities of their performances for future generations.

In the future it would be useful to see how well the techniques generalize to different snare drums (and eventually other drums in the drum set). It would also be particularly useful to add a third strike to the test set, when a player performs more complex patterns, including striking with both hands. In the future we also hope to continue work in metrics tracking for percussionists, enabling performers to evaluate their playing in live performance and in the practice room.

Of course at the heart of this research is the interplay between audio and accelerometer modalities, which can be explored well beyond the scope of surrogate sensing. It should be reiterated that the goal of this dissertation is not to reject direct sensors due to possible invasive qualities; rather this research aims to explore the various synergies between sensing modalities. Surrogate data training is one such exploration and the technique is powerful in cases where extracted features from an indirect modality provide sufficient performance data for the task, but insufficient means of event acquisition or segmentation. The crucial role of direct sensors on the performer and/or instrument is evident in the fact that the direct sensor is needed to properly train the system.

Promising work has further explored this idea, inferring the direct sensor data from the indirect sensors using multivariate regression (Tindale, Kapur, and Tzanetakis 2011). The act of surrogate training a system with direct sensor features and then synthesizing the direct sensor features from indirect sensors alone reinforces the importance of multimodal and cross-modal reciprocity in musical performance.

Chapter 6

Multimodal Onset Detection

Improving Onset Detection Algorithms in Non-Percussive Sounds using Multimodal Fusion

In this chapter, fusion and the idea of cross-modal reciprocity will continue to be investigated, with the goal of improving note and event onset detection algorithms.

6.1 On Music and Onsets

Across all genre and styles music can generally be thought of as an event-based phenomenon. Whether formal pitch relationships emerge in note-based music, or timbre-spaces evolve in non-note based forms, music (in one regard) can be thought of as sequences of events happening over some length of time. Just as performers and listeners experience a piece of music through the unfolding of events, determining when events occur within a music scenario is at the core of many music information retrieval, analysis, and musical human-computer interaction scenarios. Determining the location of events in musical analysis is typically referred to as *onset detection*, and in this section discusses a novel approach for improving the accuracy of onset detection algorithms.

Collecting and analyzing data for long-term metrics tracking experiments (in Chapter 7) revealed the need for a robust multimodal onset fusion algorithm. During initial observations of the performer’s improvisation data, the onset detection algorithms tested could not accurately segment individual notes under certain playing conditions. As such, other options were explored, ultimately leading to the multimodal approach presented in this chapter. This chapter begins by clarifying key terms and concepts, followed by an overview and implementation of the multimodal fusion algorithm developed.

An onset is often defined as the single point at which a musical note or sound reaches its initial transient. To further clarify what we refer to as the note onset, examine the waveform and envelope curve of a single snare drum hit shown in Figure 39. As one can see in the diagram, the onset is the initial moment of the transient, whereas the attack is the interval at which the amplitude envelope of the sound increases. The transient is often thought of as the period of time at which the sound is excited (e.g. struck with a hammer or bow), before the resonating decay. It should be noted that it is often the case that an onset detection algorithm chooses a local maxima as the onset from within the detected onset-space during a final peak-picking processing stage. This is true of the onset algorithm used in this experiment, and corresponds with the peak of the attack phase depicted in Figure 39.

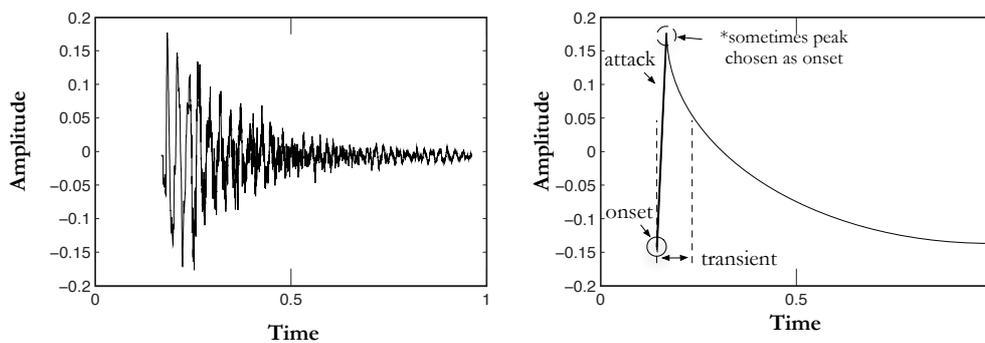


Figure 39: Snare drum waveform (left) and envelope representation (right) of the note *onset* (circle), *attack* (bold) and *transient* (dashed). Figure adapted from (Bello et al. 2005)

There are many established approaches to detecting note onsets (Bello et al. 2005; Dixon, Simon 2007; Dixon 2006; Goto and Muraoka 1999; Lartillot, Olivier et al. 2008; Scheirer 1998). For percussive sounds with fast attacks and high transient changes, algorithms in the time, frequency, magnitude, phase, and complex domains have been established and have proven to be accurate. Earlier in section 4.4 we successfully used onset detection during automatic drum-hand recognition. The task of onset detection however becomes much more difficult when sounds are pitched or more complex, especially in instruments with slow or smeared attacks (like the common stringed instruments in an orchestra).

Whereas nearly all of the common onset detection algorithms available perform analysis on the acoustic signal of the instrument alone, this research proposes a technique that fuses onsets from gestural sensor data with the onsets

detected from the acoustic signal of the instrument. Using multimodal fusion of acoustical and gestural onsets, the robustness and accuracy of the onset detection algorithm is improved, especially in non-percussive and demanding performance scenarios (e.g. quick tremolo playing).

Others have started to apply fusion techniques to the task of improving onset detection algorithms in recent years. Toh, et al. propose a machine learning based onset detection approach utilizing Gaussian Mixture Models (GMMs) to classify onset frames from non-onset frames (Toh, Zhang, and Wang 2008). In this work feature-level and decision-level fusion is investigated to improve classification results. Improving onset detection results using score-level fusion of peak-time and onset probability from multiple onset detection algorithms was explored by Degara, Pena, and Torres (Degara-Quintela, Norberto, Pena, Antonio, and Torres-Guijarro, Soledad 2009). Degara and Pena have also since adapted their approach with an additional layer in which onset peaks are used to estimate rhythmic structure. The rhythmic structure is then fed-back into a second peak-fusion stage, incorporating knowledge about the rhythmic structure of the material into the final peak decisions.

While previous efforts have shown promising results, there is much room for improvement, especially when dealing with musical contexts that do not assume a fixed tempo, or that are aperiodic in musical structure. Many onset detection algorithms also work well for particular sounds or instruments, but often do not generalize across the sonic spectrum of instruments easily. This is particularly true for pitched instruments, as demonstrated in the Music Information Retrieval Evaluation eXchange (MIREX) evaluations in recent years (Anon 2006; Anon 2007; Anon 2009). Added complexity also arises when trying to segment and correlate individual instruments from a single audio source or recording. These scenarios and others can be addressed by utilizing multimodal techniques that exploit the physical dimensionalities of direct sensors on the instruments or performers. In section 6.2 we discuss the strengths and weaknesses of performing onset detection on acoustic and sensor signals. An overview of our system and fusion algorithm is provided in sections 6.3 - 6.3.3, and finally we show how multimodal fusion can be used to integrate the strengths of both for superior results in section 6.4.

6.2 Audio vs. Sensor Onset Detection: Strengths and Weaknesses

There are many strengths and weaknesses that contribute to the overall success of audio and sensor based onset detection algorithms. The first strength of audio-based onset detection is that it is non-invasive for the performer. It is also very common to bring audio (either from a microphone or direct line input) into the computer and many machines provide built-in microphones and line inputs. This makes audio-based approaches applicable to a wide audience without the need of special hardware. In contrast, sensors have often added wires that obstruct performance, they can alter the feel and playability of the instrument, or restrict normal body movement. In the past, putting sensors on the frog of a bow could change its weight, hindering performance. In recent years however, sensors have not only become much more affordable, but also significantly smaller (and lightweight). Through engaging in communication with musicians during the experimental trials, the invasiveness of instrumental sensor systems was minimized enough for musicians not to notice that they were there at all. In fact, embeddable sensors like accelerometers and gyroscopes are already finding their way into consumer products beyond cellphones, as demonstrated by the emerging field in wearable technology. The technologies are also beginning to appear into commercial musical instruments, and wireless sensing instrument bows already exist such as the K-Bow from Keith McMillen instruments¹⁷.

Another consideration between audio onsets and sensor onsets has to do with what information the onsets are actually providing. In the acoustic domain researchers have not only explored the physical onset times but the closely related perceptual onset (when the listener first hears the sound), as well as the perceptual attack time (when the sounds rhythmic emphasis is heard) (Collins 2006; Wright 2008). These distinctions are very important to make depending on the task, and when dealing with non-percussive notes, such as a slow-bowed stroke on a cello or violin (where the rhythmically perceived onset may be much

¹⁷ <http://www.keithmcmillen.com/>

later than the initial onset). This exposes a weakness in audio-based onset detection—which has trouble with non-percussive, slow, or smeared attacks. This section shows how this weakness can sometimes be addressed by sensor onset detection that can detect slow, non-percussive onsets very well. This does not come without certain considerations, as described in greater detail later in this chapter.

In the sensor domain, the onset and surrounding data is often providing a trajectory of physical motion, which can vary from than the acoustic output, and can sometimes even be correlated with the perceptual attack time. Sometimes however, the physical onset from a sensor might not directly align with the acoustic output or perceptual attack time, and so careful co-operation between onset-fusion is necessary. In learning contexts, this trajectory can provide a highly nuanced view into information about the player's physical performance. The data can directly correlate with style, skill level, the physical attributes of the performance, and ultimately the acoustic sound produced.

As shown later in this section, the differences in the information provided from separate modalities can actually be used to strengthen our beliefs in the information from either modality individually. This helps overcome weaknesses in the modalities, such as the fact that a sensor by itself may not have any musical context (e.g. gesturing a bow in-air without actually playing on the strings). Combining information from both modalities can be used to provide the musical or other missing context from one modality for the other.

Additionally, while audio onset detection has proven to work very well for non-pitched and percussive sounds, they have increased difficulty with complex and pitched sounds. This can often be addressed with sensor onset detection that is not affected by (and does not necessarily have any concept of) pitch.

Lastly, many musical recordings and performances are outside of the practice room, and contain multiple instruments. This reality makes onset detection increasingly difficult as there is the additional task of segmenting instruments from either a single stream, or from bleed in an individual stream, as well as ambient noise and interference (e.g. clapping, coughing, door shutting, etc.). As there is a great deal of overlap in the typical ranges of sounds produced by traditional instruments, polyphonic sound separation is an extremely difficult

task. Physical sensors however are naive to other instruments and sensors other than themselves, and are typically not affected by other factors in the ambient environment. Thus, they provide (in some ways) an ideal homogenous signal from which to determine, or strengthen onset predictions.

Strengths	Weaknesses
<ul style="list-style-type: none"> • Non-invasive • Onset time can be close to perceptual attack time • No special hardware 	<p style="text-align: center;">Audio</p> <ul style="list-style-type: none"> • Algorithms have trouble with pitched and complex sounds • Algorithms have trouble with slow / smeared attacks • Ambient noise / interference • Source segmentation / non-homogenous recording
<ul style="list-style-type: none"> • Very sensitive physical measurements and trajectories • Can detect onsets from slow / smeared attacks • Not negatively affected by pitched or complex sounds • Resistant to factors in the environment / ambience • Typically mono-sources, no separation necessary 	<p style="text-align: center;">Sensor</p> <ul style="list-style-type: none"> • Can sometimes be invasive • No musical context • Onsets may or may not be related to the acoustic / auditory onsets

Figure 40: Strengths and Weaknesses of Audio and Sensor Onset Detection

6.3 System Design and Implementation

In designing this system, a primary goal was to create a fusion algorithm that could operate independently of any one particular onset detection algorithm. In this way, the system was designed such that it is given with two onset streams (one for onsets detected from the acoustic or audio stream of the instrument, and one from the sensors), without bias or dependence on a particular algorithm. The onset algorithms can be tuned both to the task and individual modalities (perhaps one onset detection function works best for a particular sensor stream vs. another sensor stream vs. the audio stream), while enabling compatibility with future onset functions that do not currently exist. Fusion happens as a post-processing step (late-fusion) that does not replace, but rather improves, the robustness and accuracy of the chosen onset detection algorithm(s).

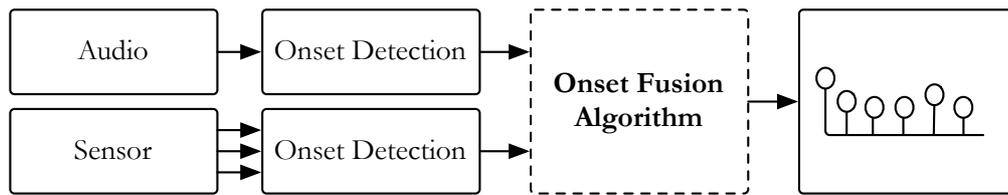


Figure 41: General Overview of Multimodal Onset Fusion

6.3.1 ONSET DETECTION FUNCTION

The onset detection used in these experiments works by computing the power-spectrogram of the waveform to extract its envelope. By default the frame size is 100ms, using a Hanning window and a hop factor of 10% although the parameters are adjustable depending on the task and need. The spectrogram is summed along the frequencies (frame-by-frame), resulting in a final onset curve. The onset curve is peak-picked at local maxima to determine the location (onsets) of the notes. The peak-picking function can be specified to use local minima as the onset positions, which corresponds more directly with the typical onset definition as described in the introduction. More information on the onset detection function and peak-picking algorithm can be found in (Lartillot 2011).

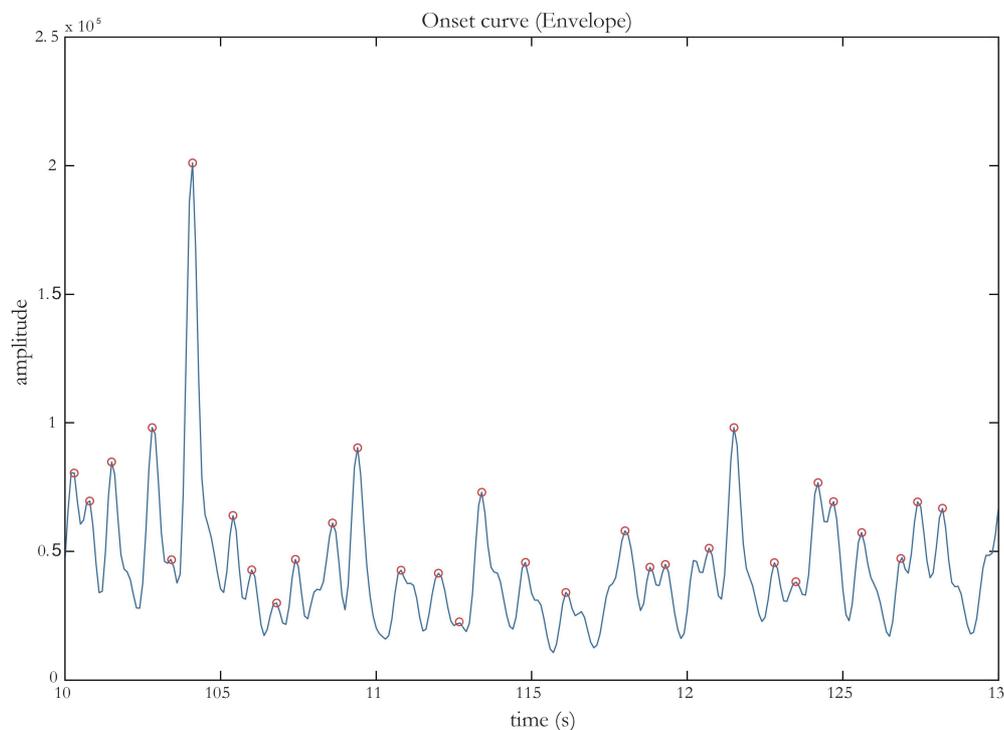
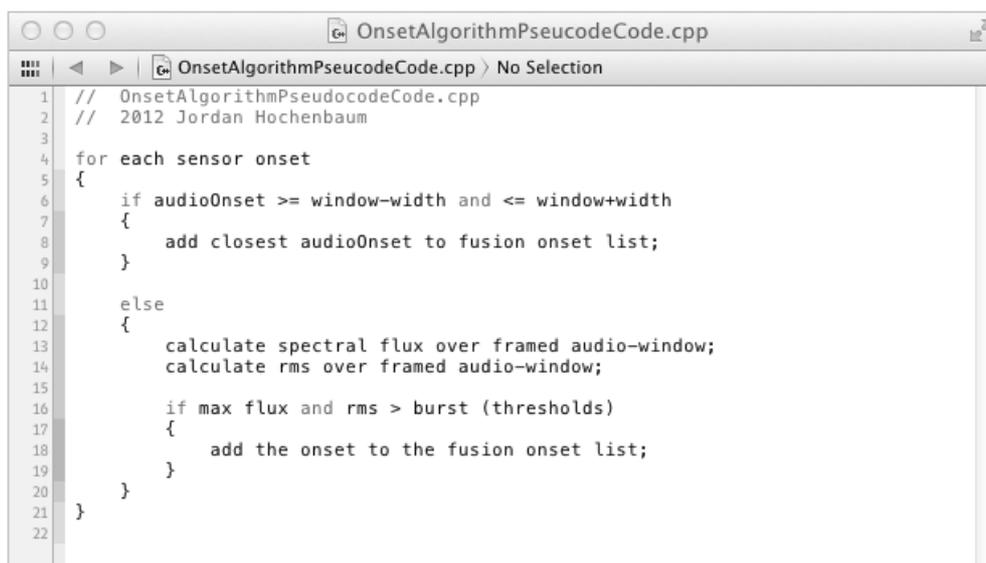


Figure 42: Onset Curve (envelope) and peak-picked onsets (circles) for a short window of audio

6.3.2 FUSION ALGORITHM

The fusion function is provided with two onset streams, one from the audio output and one for the sensor(s). First the algorithm searches for audio onsets residing within a window (threshold) of each accelerometer onset. A typical window size is 30ms – 60ms and is an adjustable parameter called *width* which effects the sensitivity of the algorithm. If one or more audio onsets are detected within the window of a sensor onset, our belief increases; the best (closest in time) audio onset is considered a true onset; the onset is then added to the final output fusion onset list.

The image shows a screenshot of a code editor window titled "OnsetAlgorithmPseudoCode.cpp". The code is as follows:

```
1 // OnsetAlgorithmPseudoCode.cpp
2 // 2012 Jordan Hochenbaum
3
4 for each sensor onset
5 {
6     if audioOnset >= window-width and <= window+width
7     {
8         add closest audioOnset to fusion onset list;
9     }
10
11     else
12     {
13         calculate spectral flux over framed audio-window;
14         calculate rms over framed audio-window;
15
16         if max flux and rms > burst (thresholds)
17         {
18             add the onset to the fusion onset list;
19         }
20     }
21 }
22
```

Figure 43: Onset fusion algorithm pseudo-code

If a sensor onset is detected, however, no audio onset is found within the window (*width*), there is only a partial belief that the sensor onset is an actual note onset. To give a musical context to the sensor onset, the audio window is split into multiple frames and spectral-flux and RMS are calculated between successive frames. The max flux and RMS values are then evaluated against a threshold parameter called *burst* to determine if there is significant (relative) spectral and amplitude change in the framed audio-window. Because onsets are typically characterized by a sudden burst of energy, if there is enough novelty in the flux and RMS values (crosses the burst threshold), the belief in the onset

increases and the sensor onset time is added to the fusion onset list. The burst threshold is a dynamic value that is calculated as a percentage above the average spectral-flux and average RMS from the audio-window. By default, burst is set to equal 20% increase in the average flux value, and a 10% increase in the average RMS from the current audio window. Increasing or decreasing the burst threshold decreases or increases the sensitivity to change in the relative spectral flux and RMS, ultimately changing the algorithms sensitivity.

6.3.3 DATA COLLECTION

The data used to test the onset fusion algorithm was chosen from a subset of the improvisation recordings (data set D4) recorded and discussed later in chapter 7.3.1. It was while analyzing the player's improvisations on the Ezither that we began to notice the difficulty in detecting note events in certain circumstances. The most obvious case was when the performer bowed back and forth very quickly in a tremolo style, and sometimes when bowing long, slow notes, and so this study focuses on a subset of recordings examining these scenarios. Five excerpts were extracted from four recordings made over the period of a month from January 20th 2012 through February 24th 2012, as well as an additional recording where the performer practiced playing tremolos in late March of 2012. In total 3,697 bows-strokes were performed, the majority being in a tremolo or similar style.

6.4 Onset Detection and Fusion Results

This section compares and contrasts the performance of the onset detection algorithm on the audio recording, the sensor recording, and finally using the multimodal onset fusion algorithm. Specifically the onset detection's performance will be gauged by looking at the following common statistical performance and correctness measures:

1. True Positives (TP) – The number of detected onsets that have been validated as notes actually played

2. False Positives (FP) – The number of detected onsets that have been validated as notes that *were not* actually played
3. False Negatives (FN) – The number of onsets that were actually played but not detected by the algorithm

All recordings and bow-strokes were annotated by hand to validate the onset detection results.

6.4.1 DISCUSSION: AUDIO-ONLY ONSET DETECTION RESULTS

The need to look for alternative onset detection methodology became apparent when analyzing audio recordings from the Ezither performer. As soon as the performer played at faster speeds or tremolo, or sometimes very long and slowly, the onset detection algorithms being used had difficulty detecting the notes played. In Figure 44, the audio recording of a performance excerpt is shown as the purple waveform, along with a spectrogram above. Detected onsets (TP) are shown as the tall black vertical lines and the areas highlighted with grey rectangles estimate problem zones where one or more onsets were not detected (FN).

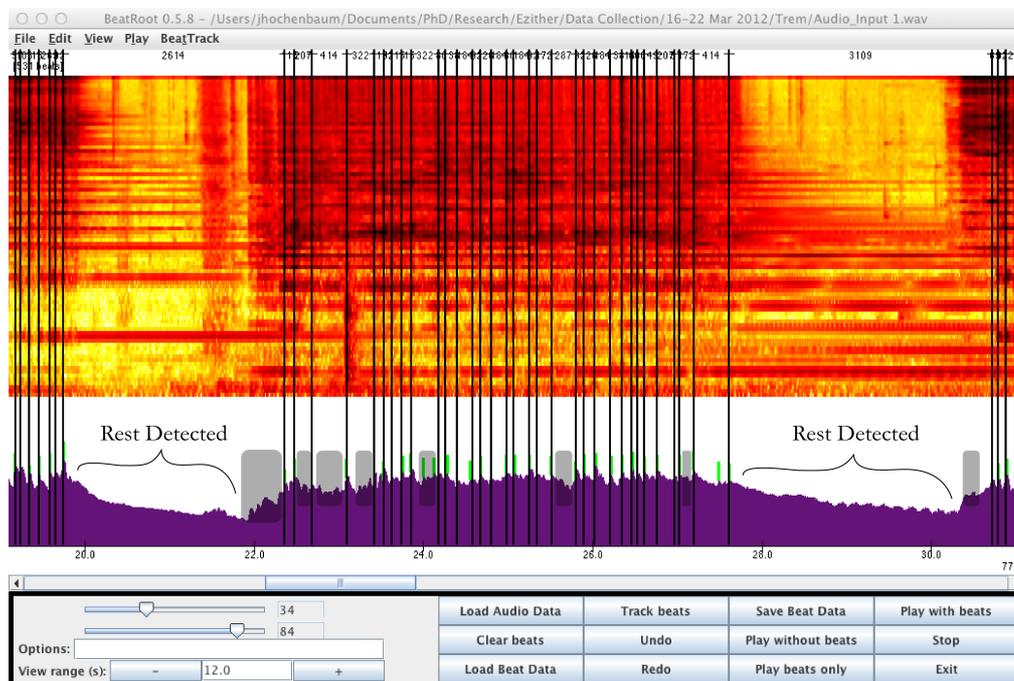


Figure 44: Audio onsets detected over an excerpt of mostly tremolo playing, TP (black vertical lines), FN (grey rectangles)

Table 15: Distribution of onsets detected from audio-only as either True Positive, False Positive, or False Negative

# Bows	True Positive	False Positive	False Negative
1445	1425	20	2172

In terms of the number of onsets detected, audio-only onset detection performed the poorest when compared to accelerometer onset detection and fusion detection. When analyzing the audio recording, the onset detector identified only 1445 of 3597 bow-strokes performed. This can be seen as an extremely high FN rate. It should be noted that we were interested in improving the onset detection algorithm when bowing within certain problem scenarios, and so it is to be expected that the number of detected onsets is low. Out of the 1445 strokes detected however, nearly all but 20 were actual onsets. As performing onset detection on the audio recording analyzes the actual musical output, the onset detection is robust to slow-moderate playing and musical rests, as marked in Figure 44.

6.4.2 DISCUSSION: SENSOR-ONLY ONSET DETECTION RESULTS

The first thing one might notice when comparing the sensor onsets detected in Figure 45 to the audio onsets detected in Figure 44 is the increased resolution of TP onsets. This heightened sensitivity however comes at the cost of detecting false positives when the player continues to move the bow between strikes. In Figure 45 the grey rectangles now represent FP (they represented FN previously in Figure 44) and one can see that 17 onsets are falsely identified during the rests previously detected in audio onset detection.

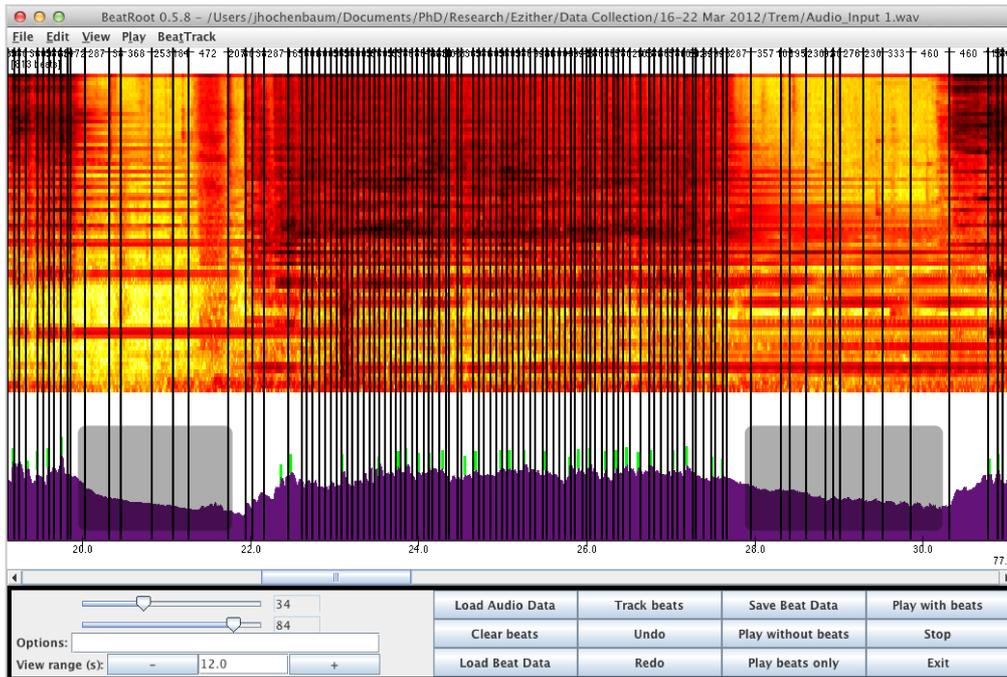


Figure 45: Sensor (accelerometer) onsets detected over an excerpt of mostly tremolo playing, TP (black vertical lines), FP (grey rectangles)

Whereas audio onset detection possessed the highest FN rate, sensor (accelerometer) onset detection missed the fewest notes, resulting in the highest TP and lowest FN rates. Sensor-only onset detection resulted in 3111 of 3597 notes played being detected, leaving 486 undetected and a total of 467 incorrectly detected. While the sensor onset detection does exhibit a higher FP rate than audio onset detection, comparatively, it performs much more accurately in this scenario as it has far fewer FN's (486 compared to 2172).

Table 16: Distribution of onsets detected from sensor-only (accelerometer) as either True Positive, False Positive, or False Negative

# Bows	True Positive	False Positive	False Negative
3578	3111	467	486

6.4.3 DISCUSSION: MULTIMODAL ONSET FUSION RESULTS

An excerpt of the fused onsets determined by running both audio and sensor (accelerometer) onsets through the fusion algorithm is shown in Figure 46. Compared to the sensor onsets shown over the same excerpt in Figure 45, most of the sensitivity and resolution is preserved while minimizing the amount of FP's. A few FN's do reappear and are shown as grey rectangles.

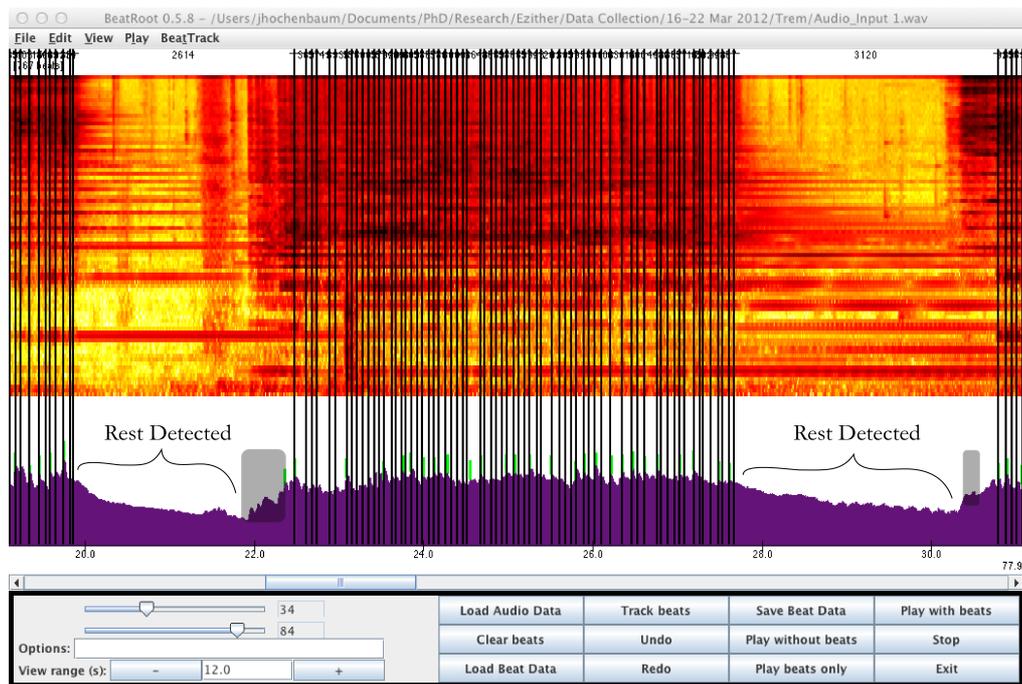


Figure 46: Multimodal fusion onsets detected over an excerpt of mostly tremolo playing, TP (black vertical lines), FN (grey rectangles)

Table 17: Distribution of onsets detected from multimodal onset fusion as True Positive, False Positive, or False Negative

# Bows	True Positive	False Positive	False Negative
3178	3022	156	575

Compared to sensor onsets, fusion onsets greatly lower the number of false onsets detected, reducing FP's from 467 to 156 bow strokes. The number of TP's stays quite high as well, retaining a total of 3022 correctly identified strokes out of the 3597 notes played in total.

6.4.4 DISCUSSION: PRECISION, RECALL, AND F_1 -MEASURE

Precision, *recall* and *F-Measure* are common evaluation measures for set-based analysis that are used to evaluate the quality of a retrieval or classification scenario. Take for example the case where a website search engine is given the task of returning a list of websites for a given search query. Precision would represent the portion of websites returned which are relevant to the actual search. In general this can be defined in the following equation

$$precision = \frac{|{\{relevant\ items\}} \cap {\{retrieved\ items\}}|}{|{\{retrieved\ items\}}|}$$

where relevant items are a subset of all items retrieved. Thus precision can be used to evaluate the performance of the onset detection algorithm in terms of the accuracy of the detected onsets by substituting

$$precision = \frac{TP}{TP + FP}$$

Examining the results in Figure 47 audio onsets returned the most precise results (98.62%), followed by the fusion onsets (95.09%), and lastly sensor onsets (86.95%). The number of notes detected by audio-only onset detection however was far fewer than both accelerometer and fusion onsets (1445 vs. 3578 vs. 3178 respectively), and so the precision of the audio-only onsets comes at a great cost when compared to the accelerometer precision. Using multimodal onset fusion however yields over 8% gain in precision over sensor onsets, leaving just about a 3.5% difference between fusion and audio-only onset detection—while also preserving the majority of TP's returned by sensor onset detection.

This leads to Recall which in the web search example would represent the portion of websites relevant to the search that are retrieved from the total number of relevant websites in the search database. It can be thought of as measuring the search engine's ability to present only those items that are in fact relevant to the query. In general terms recall can be defined as

$$recall = \frac{|{\{relevant\ items\}} \cap {\{retrieved\ items\}}|}{|{\{total\ relevant\ items\}}|}$$

where the numerator equals relevant items from the subset of retrieved items and the denominator is the total number of relevant items in the entire query space. Thus we can evaluate the onset detection performance in terms of how well it retrieves relevant documents (from the ground-truth) by substituting

$$recall = \frac{TP}{TP + FN}$$

As illustrated in Figure 47 audio-only onset detection performed the poorest in terms of recall rate, returning less than half of the total TP's at a mere 39.62%. Sensor onset detection scored the highest here with a recall rate of 86.49%, retrieving TP onsets nearly 47% better than audio onset detection. Onset fusion scored just under 2.5% lower than sensor-only onsets, which is to be expected as the fusion algorithm parameters were generalized across the data sets vs. being optimized to fit each data set as best as possible.

Fine-tuning the fusion algorithm parameters can help minimize the inherent trade-off between discarding TP's from the sensor onsets, while attempting to reduce the amount of FP's (by discarding).

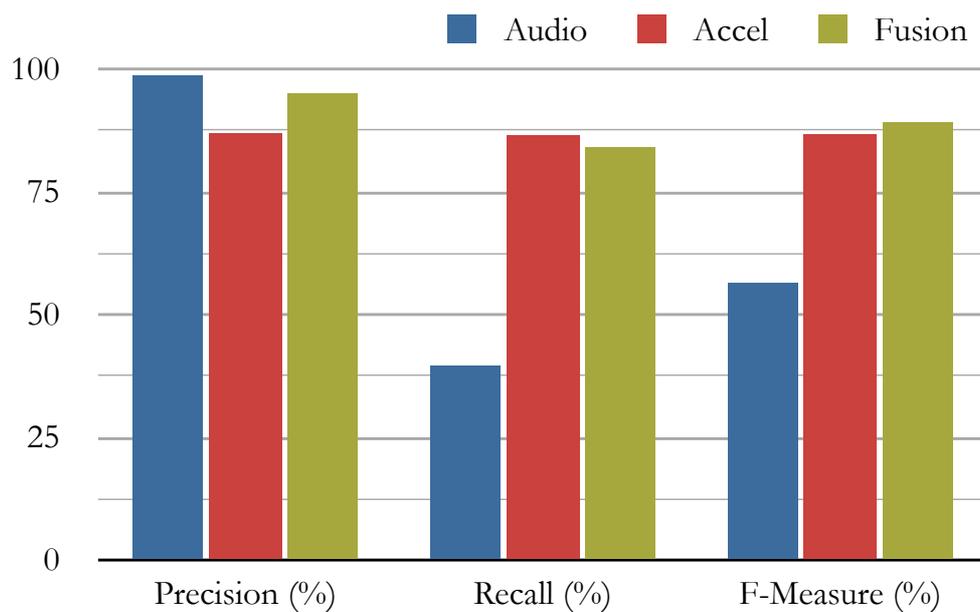


Figure 47: Comparison of precision, recall, and F₁-Measure for audio, sensor, and fusion onsets

Table 18: Comparison of precision, recall, and F₁-Measure for audio, sensor, and fusion onsets

Onsets	Precision (%)	Recall (%)	F ₁ -Measure (%)
Audio	98.62	39.62	56.53
Accel	86.95	86.49	86.72
Fusion	95.09	84.01	89.21

In a sense the fusion algorithm attempts to synergistically achieve the precision of audio onset detection, with the recall of the sensor onset detection. There often exists a trade-off between precision and recall, where greater precision yields poorer recall and vice versa—this can be seen in the audio-only onset results.

Both of these measures (precision and recall) can be taken into account by taking the harmonic mean, called the F-Measure. Because we weight both precision and recall equally, we refer to the measure as the F_1 -Measure, which is defined as

$$F = 2 \times \frac{\textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}}$$

Taking into account both precision and recall, multimodal onset fusion yields the best results with F_1 -Measure of 89.21%. Audio-only onset detection performed the poorest at 56.53%, and sensor-only onset detection at 86.72%. In musical performance contexts, as well as in analysis and information retrieval scenarios, a balance must be struck between minimizing false detection, while maximizing true detection. The F_1 -Measure score signifies this balance. The increase in F_1 -Measure between audio onsets, sensor onsets, and fusion onsets reveals even greater significance however when placed in real-world contexts.

6.5 Musical Contexts and Conclusions

In a musical context how can we interpret the 2.5% decrease in recall between accelerometer onsets and fusion onsets and the 8% increase in precision between accelerometer onsets and fusion onsets? Investigating Figure 48 can shed insight into these questions. In terms of actual numbers, we can see that this relates to many less FP's being detected by the fusion onsets (146-fusion vs. 467-sensor), while the amount of TP's remains very close proportionally (3022-fusion vs. 3111-sensor). Relating back to Figure 45 and Figure 46, it was observed that at the loss of just a few notes from being detected, we gained the ability to discard many more false positives, which musically speaking were moments such as rests

and silence. In a performance context, false positives could equate to unwanted notes being played, processes and effects triggered, etc. In an analysis or information retrieval scenario, analyzing and/or extracting features at false locations can very negatively effect the results.

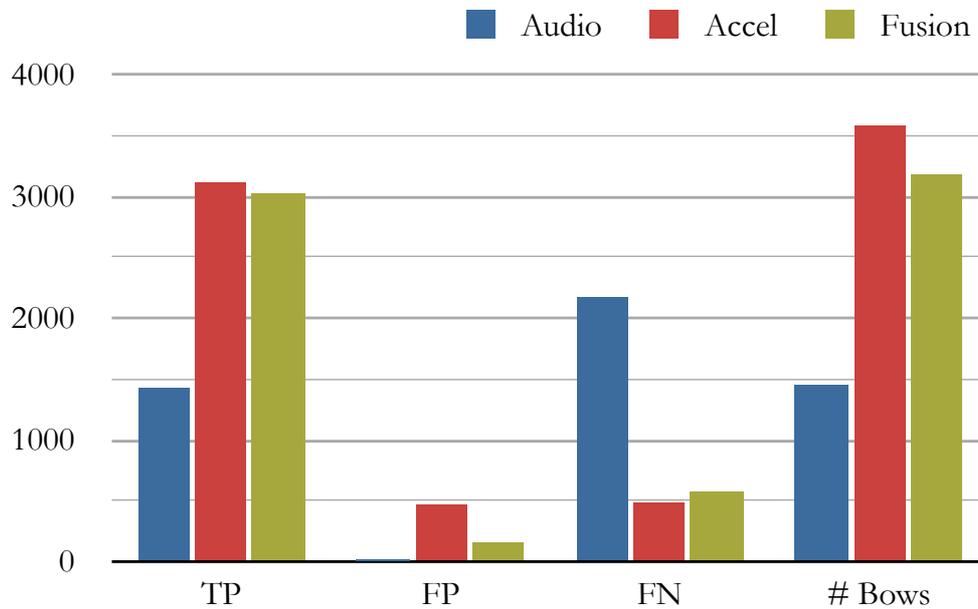


Figure 48: Comparison of TP, FP, FN and #Bows for Audio, Sensor, and Fusion Onsets

Table 19: Comparison of TP, FP, FN and #Bows for Audio, Sensor, and Fusion Onsets

Onsets	TP	FP	FN	# Bows
Audio	1425	20	2172	1445
Accel	3111	467	486	3578
Fusion	3022	156	575	3178

Results could be further improved by tailoring the fusion parameters more specifically to the data being analyzed. There are many ways to do this, both by hand and dynamically, and we hope to explore these in the future. For example, dynamic range compression (DRC) during a pre-processing phase could help generalize certain parameters by reducing the amount of variance in the dynamic range of the input data, which changes from day to day and recording to recording. Additionally, there is a considerable room to experiment with the onset detection function currently used, not only in terms of adjustable

parameters, but also in using different onset detection functions that are tailored to exhibit better performance for a specific modality.

This chapter showed both the power and promise of multimodal fusion for improving onset detection. Whether a performer, an audience member/listener, or a scientist, all humans possess the ability to deduce when musical events occur with great precision and recall. This ability to correctly differentiate musical events is at the core of all music related tasks, and in this chapter we have shown how the ability for computers to perform similar tasks can be greatly improved by the use of multimodal techniques.

Chapter 7

Rethinking How We Learn: Performance Metrics and Multimodality in the Practice Room

Investigating the Role of Multimodality in the Musical Practice Room

Music education is a rich subject with many approaches and methodologies that have developed over hundreds of years. More than ever, technology plays important roles at many levels of a musician's practice. This chapter begins to explore some of the ways in which technology, specifically with the help of multimodality, can inform a musician's daily practice, through short and long term metrics tracking and data visualization.

7.1 Background and Motivation

Pursuing a higher education degree in music today, one will observe first-hand the increasing prominence of technology in the life of practicing musicians. It is not uncommon to see musicians recording ensemble practices and private lessons with portable recorders or laptops. The field recorder is often thought of as one of the most important inventions for ethno-musicological purposes but its impact on western music *practice* is also very significant. Portable recording devices however are just one of the simplest ways in which technology permeates today's learning environments.

Many music programs (at the university and primary/secondary levels) now have "keyboard labs" where a group of students gather around computers with headphones and MIDI keyboards, engaging with interactive musicianship skills software. Computer-assisted learning has gained popularity in recent years and

most interactive software applications cover topics as diverse as aural identification/ear training, rhythm skills, scales, harmony, and other theory topics. Ear Conditioner¹⁸, Auralia¹⁹, Practica Musica²⁰, EarMaster²¹, are a few of the many applications being used in music schools around the world everyday, which enable musicians to be conducted through aural and theory exercises with a virtual guide. Most operate by receiving symbolic (MIDI) input from users playing a digital piano keyboard or via mouse/keyboard computer input. While these software applications have proven to be effective, there are many limitations as a result of the restricted input modalities.

Firstly, computer-assisted music training currently gauges a (non-pianist) musician's abilities via input other than their actual instrument. While basic keyboard skills are important for all musicians to acquire (at least in the Western tradition), it is important to engage and assess the student on their actual instrument or voice.

This leads to the second limitation, namely that the software is listening to the musician's input in a narrow manner. Input via a MIDI keyboard is a step in the right direction; however, it does not provide insight into the acoustical and physical dimensionalities, two elements that are crucial to musical performance. This is what brings learning musicians in front of instructors, tutors, gurus, every day—years of experience, knowledge, and human musicianship.

There are many other ways in which technology is influencing the environment in which musicians now learn. Universities such as McGill University in Montreal and others have been pushing the idea of “distance learning” in many of their disciplines (Bofinger and Whateley 2002; Bouillot and Cooperstock 2009; Jegede and Shive 2001; Lancaster 2007). In music education this enables educators (whether on tour, or music living in other countries) to administer lessons from afar via video conferencing technology. Anthropomorphic robotic music instructors that are capable of responding to human performers have even been explored (Petersen et al. 2008). Recently,

¹⁸ <http://www.michaelnorris.info/>

¹⁹ <http://www.sibelius.com/products/auralia/index.html>

²⁰ <http://www.ars-nova.com/practica6.html>

²¹ <http://www.earmaster.com/>

Percival presented an interesting approach to computer-assisted violin practice and a good overview of the current state of “Computer-Assisted Musical Instrument Tutoring” (CAMIT) musical in (Percival and Schloss 2008). In line with the goals of this chapter, Percival places a strong emphasis on creating systems that concentrate a musician’s interactive practice exercises on areas that need the most practice, rather than the (relatively naive) general theory based software approaches that currently exist.

Of course traditional musicianship training (sight-reading, ear training, rudiment training, chord identification, etc.) in the form of classroom activities, private practice, and from engaging with other musicians in performances will always be essential to the future musical learning environment. Invaluable feedback from professional musicians and educators will always play a needed role in a practicing musicians development. The musical classroom however is an ever-expanding environment, moving beyond its traditional latitude. Today, searching “guitar lesson” on YouTube yields nearly 1-million results, tomorrow who knows?

This research asks how can we advance technology to supplement ones musical practice, both inside and outside of scheduled class times and lessons? How can multimodal signal processing provide musicians and educators alike, focused insight into the acoustical and physical dimensionalities of a musicians practice? This research begins to parameterize and visualize this information in an attempt to inform musicians and educators, following musical pedagogy into new domains. To that end, this chapter explores multimodal metrics, tracking the day-to-day, and long-term evolutions, of a musicians practice. Specifically, this chapter investigates metrics pertaining to the player’s tempo performance and accuracy, as well as the performer’s practice of various bow strokes (articulations). In addition to metrics and statistical measures, various visualizations and statistical representations are proposed, which can provide musicians and instructors with nuanced information about the performers playing at a glance.

7.2 Overview of Metrics Experiments

These experiments investigate metrics from a musician learning a custom bowed string instrument called the Ezither (see section 3.1.2).

7.3 Data Collection

Nuance was used to record a variety of data sets for the Ezither performer, capturing the variability of the player's performances under scenarios ranging from typical practice routines to improvisation. The following section describes in greater detail the data sets collected spanning these grounds.

7.3.1 EZITHER DATA

For roughly seven months between August 12th 2011 and March 22nd 2012, the Ezither performer regularly recorded his practice. As this was a new, custom-built instrument, the performer was at a beginner level, and had no real prior experience playing a bowed stringed instrument (although he was a trained musician and composer on other instruments). The total data collected consisted of sixteen practice sessions over the seven-month period.

During each session the performer recorded four discrete data sets. The first data set (D1) targeted the practice of various bow strokes including *Detaché*, *Martelé*, and *Spicatto*. During a session each stroke was played for roughly 30 seconds in up-bow down-bow succession at 120 beats-per-minute. The player was restricted to playing on one string (the lowest string, C) of the instrument to limit the effect of string and position changes on stroke performance.

The second data set (D2) aimed to capture the performer's variability in tempo performance. As such the performer arpeggiated up and down the open-strings of the instruments at three tempi, *Andante* (80bpm), *Moderato* (110bpm), and *Allegro* (140bpm). The passage was recorded in up-bow down-bow succession for roughly two-minutes.

In data set 3 (D3) the performer repeated a melody for about two minutes. The melody was played at a fixed tempo (100 bpm) however the line was less-constrained than data sets D1 and D2 in that it was not confined to a single

string or moving up and down the strings linearly. The melody was a simple 4-measure long line as noted in Figure 49. As the performer was a beginner Ezither player, the melody line mostly moved in a scalar fashion, with one small intervallic leap in the last measure.



Figure 49: Melody Repeated in Data Set 3 (D3)

Lastly the final data set (D4) was purely improvisational. No instructions were given to the performer other than he should play whatever he liked. The performer was free to bow the notes, pluck the notes, and work his way through 2-minute long mini improvisations (while listening to a metronome at 120 bpm).

7.4 Tempo Metrics and Statistics

The ability for a musician to perform at various speeds is of utmost important in musical performance. The tempo of a piece of music or section influences the music on many levels, from its performability, to the music's affect and intent. This section focuses on examining the player's performance at various tempi (with the goal of playing as closely to the target tempo, with as little deviation as possible).

7.4.1 TEMPO ESTIMATION ALGORITHM

Tempo estimation follows the algorithm proposed in (Lartillot 2011). First a filterbank decomposes the signal into multiple auditory channels and the envelope of each channel is extracted. Each channel is half-wave rectified (as interest is in the increase of energy), the signal is differentiated to emphasize peaks, and each channel is summed together. Next the periodicity of the signals onset envelope is extrapolated through autocorrelation (at time lags corresponding to a range of tempi), and the autocorrelation coefficients are normalized to compensate for higher coefficients given for small lags. Finally tempo is estimated through a peak-picking function.

7.4.2 TEMPO: PERFORMANCE TIMING

As the tempo estimation algorithm determines the highest periodicity within the entire signal, the tempo estimated provides useful insight into the overall timing overture of the performance.

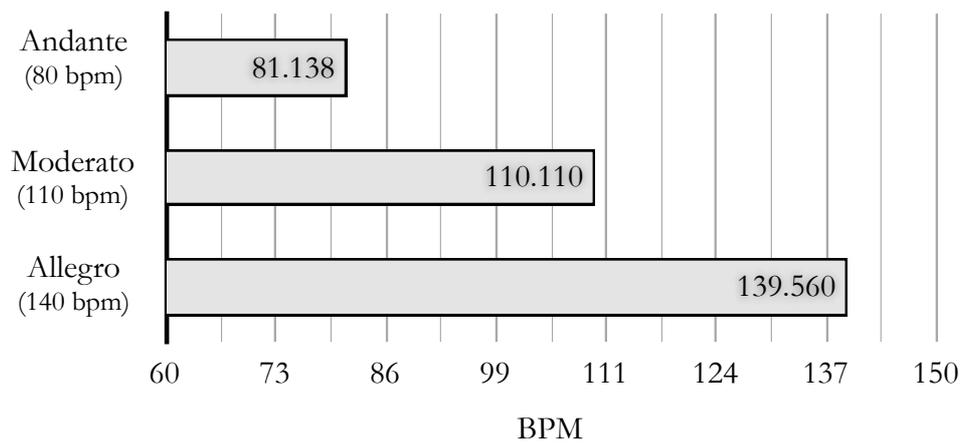


Figure 50: Tempo estimated for three recordings from Ezither recording #7, data set D2

Figure 50 shows the estimated tempos for three recordings from the Ezither recording session #7, data set D2. The first recording was played to a click track andante, with a target tempo of 80 bpm. The estimated tempo of the 2-minute performance was 81.138 bpm. This observation shows that the player was generally playing faster than the target tempo. Best performance was found at moderato speed (110 bpm), estimating an overall performance tempo of 110.110. The performer's allegro recording yielded 139.560 bpm, just under 0.5 bpm slower than the target tempo (140 bpm). In terms of practice this brings up two interesting points. Firstly, playing at a “faster” tempo does not necessarily yield less accurate performance. In this particular case the slowest bpm yielded the poorest results in terms of overall accuracy, while the best performance was achieved at the mid-speed tempo performed. While it seems plausible to say that a beginner musician may perform more accurately at slower tempi with a relatively low upper ceiling in playing speed, the relationship between tempo and accuracy is not necessarily linear, even for intermediate to advanced performers

with a wider range in acceptable speeds²². This leads to the second point, which is that in a pedagogical setting tempo estimation can be extremely helpful for practicing musicians when targeting or emphasizing areas of practice. Of course looking at one recording may not generalize about the overall trend of the performer, and so evolutionary and long-term analyses are very useful, as will be presented in sections 7.4.3 and 7.6 respectively.

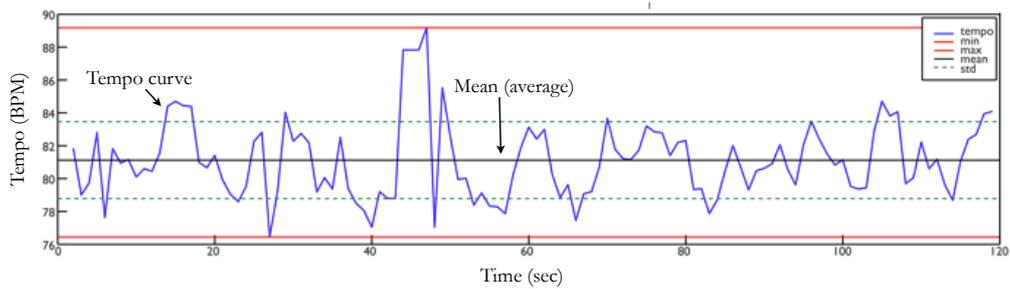
7.4.3 TEMPO: EVOLUTION OF TIMING OVER A PERFORMANCE

Tempo estimation over the entire recording is useful to inform one about the overall nature of a performance's tempo; however, it does not show the temporal evolution over the duration of a piece. To do so, the audio is framed (windowed) and tempo estimation is performed multiple times throughout the performance. In these examples, a frame-size of four seconds was used with a hop-size of twenty-five percent to ensure a minimum of four beats (one bar) was contained within a given frame. Visualizing the tempo evolution and discussing with performers, one to two bars was found to be an acceptable compromise between frequency of the tempo estimation and clarity of the tempo graph; displaying a finer frame-resolution (as used in 7.4.2) was often not macro enough to view overall trending. To further investigate the data discussed in 7.4.2, statistics are provided in Table 20, paired with visualizations and trending in Figure 51. Each graph in the figure represents a different tempo from the data set, and annotates various statistics (that are also presented in Table 20).

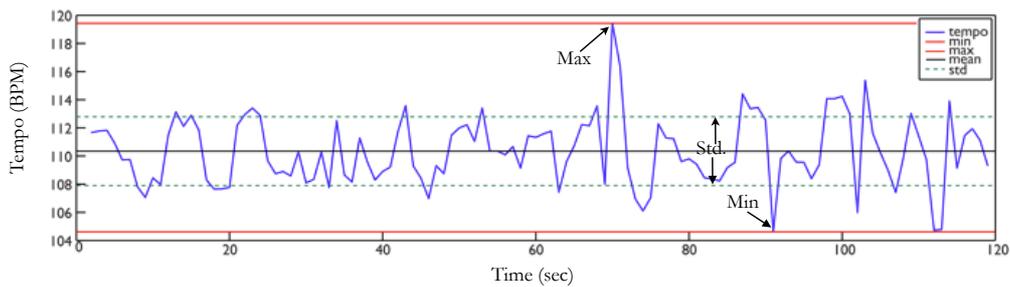
Table 20: Tempo evolution statistics (min, max, mean, standard deviation, and range) of Ezither recording #7, data set D2, andante (80 bpm), moderato (110 bpm), and allegro (140 bpm)

BPM	Min	Max	Mean	Std. Dev.	Range
Andante (80 bpm)	76.44	89.18	81.12	2.34	12.74
Moderato (110 bpm)	104.60	119.40	110.34	2.44	14.82
Allegro (140 bpm)	130.90	154.10	139.41	4.24	23.24

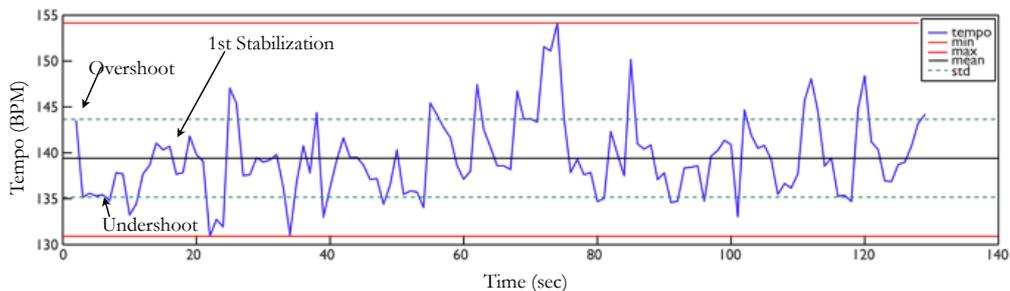
²² An “acceptable speed” is of course subjective to the performer/listener/observer, and for our purposes generally describes speeds in which the performer can play with relatively high confidence and accuracy



A – andante (80 bpm)



B – moderato (110 bpm)



C – allegro (140 bpm)

Figure 51: Tempo evolution of Ezither recording #7, data set D2, (a) andante, (b) moderato, (c) allegro

In the previous section this research showed that the best tempo performance (closest to the target tempo) was achieved when playing moderato, followed by allegro, and finally andante. This was interesting as the poorest performance was at the slowest tempo, and so it is useful to look at the statistics displayed in Figure 51 and Table 20 to gather more specifics concerning the actual performance. The first general observations of the visualizations relate to the blue tempo line detailing the estimated tempo curve over the duration of the performance (annotated in *A*). Even with a four beat (one-bar) count-in the performer had to “get into the groove” before the recording commenced, it is apparent that (usually) the performer initially overshoots in tempo (plays faster),

followed by undershooting (slowing down to compensate), before stabilizing into the tempo (see the annotations in Figure 51, C). This could inform the performer to practice an exercise where he/she might start specifically from a stopped state, targeting the accuracy of the player's initial timing.

The solid black horizontal line on each graph shows the arithmetic mean (average) tempo of the performance (annotated in A). As expected, the calculated means are very close to the overall tempi estimated in the previous section. The difference is expected as the periodicity and tempo estimated is calculated over a shorter context and then averaged, and thus will be slightly different than when considering the entire recording in the periodicity space. As confidence in the calculated tempo (from the tempo estimation algorithm used) is already quite high, the mean line can be swapped on the figure with the previously estimated tempo. However, for reference, the difference (in bpm) between the original tempo estimate and the average of the evolutionary tempo for each speed performed is: 0.018 (andante), 0.19 (moderato), and 0.16 (allegro).

The solid red horizontal lines at the extrema show the minima and maxima of each graph in the set (annotated in B). The difference between the min and max for each value determines its *range*, as shown in Table 20. The tightest range in values was achieved for andante (12.74), followed by moderato (14.82), and then allegro (23.24). In general the smaller the range, the more consistent or less variability in tempo there was over the duration of the performance. This is related to the standard deviation (shown in the graphs as two dashed green horizontal lines, annotated in B), which similarly measures the amount of variation or dispersion from the average. In Table 20, the smallest standard deviation was achieved in the andante performance (2.34), followed by moderato (2.44), and lastly allegro (4.24).

Taking all of this information into consideration, the initial observation that the overall best tempo performance was achieved for moderato, followed by allegro, and finally andante, can be revisited. While this is true in particular for overall tempo of the performance, this does not hold true in other regards. Whereas in overall tempo andante performance was technically the poorest, in terms of consistency andante performance achieved the highest rates (lowest standard deviation and range).

It should be noted that although andante's calculated standard deviation was technically lower than moderato's, hypothesis/statistical significance testing shows that while both andante's and moderato's standard deviations are statistically significantly different than allegro, they are not statistically significantly different from one another. This is supported using a Fisher-Snedecor Distribution (also called the F-Distribution) "F-test". The F-test assesses whether the values of a quantitative variable within different groups actually differ from each other. In this case, the null hypothesis would be that there is no significant difference in standard deviation of tempo when playing the instrument at andante speed vs. moderato speed. Thus, the two-tailed null (H_0) and alternative (H_A) hypotheses are defined as

$$H_0: \sigma_1^2 = \sigma_2^2 \quad \text{and} \quad H_A: \sigma_1^2 \neq \sigma_2^2$$

where in this case, σ_1^2 is moderato's variance, and σ_2^2 is andante's variance. The F-test statistic (F ratio) is thus defined

$$F = \frac{S_1^2}{S_2^2}$$

where F is an F-value determined by dividing S_1^2 (one sets variance) by S_2^2 (the other sets variance). As such moderato's variance substitutes the numerator and andante's variance substitutes the denominator, by taking the square of their standard deviations previously reported in Table 20.

$$F = \frac{2.44^2}{2.34^2} = \mathbf{1.0873}$$

Lastly, an F-table is used to compare F to the critical value, which is a value that describes the possibility of getting a particular value for F , at a specified level of significance. In this case, a confidence interval of 95% was chosen, also called a significance level of $\alpha = 0.05$, and the F-table at that significance level

was consulted. Because the critical value is dependent on sample size, the F-table is organized by degrees of freedom in the numerator and denominator of the F-ratio. Both andante and moderato for the data in Table 20 consisted of 118 tempo estimations ($n_1 = 118$, $n_2 = 118$), and as such, the degrees of freedom of the numerator, DFn , and the degrees of freedom of the denominator, DFd are

$$DFn: n_1 - 1 = 118 - 1 = \mathbf{117} \quad \text{and} \quad DFd: n_2 - 1 = 118 - 1 = \mathbf{117}$$

Consulting the F-Table with $\alpha = 0.05$ and $F(117, 117)$, the critical value $F_{crit} = 1.3571$ and p value $p = 0.3257$. F_{crit} is greater than F , and the p value is greater than the significance level of 0.05, and so the null hypothesis must be accepted. As such it is determined that in this case, the standard deviation for andante performance is not seemingly statistically significantly different from moderato's standard deviation. When comparing standard deviation for andante and moderato to allegro however, $F = 3.2832$ and $F = 3.0196$ (respectively). The critical value obtained from the F-Table using $F(127, 117)$ (because moderato is a slightly larger sample) is $F_{crit} = 1.3503$, and in both cases F is greater than the critical value, yielding $p = .0035$ for andante, $p = 0.0057$ for moderato. As such, the null hypothesis is rejected and it is determined that the standard deviations for both andante and moderato are extremely statistically significantly different than allegro's standard deviation.

Allegro which was the median performer in overall tempo accuracy, has almost two-times the standard deviation and range in tempo when compared to andante and moderato. In a sense these metrics can be associated with accuracy vs. precision of performance at each tempo, where accuracy is defined as the proximity of the overall (or frame-averaged) tempo to the target tempo, and the precision is defined as the repeatability of the tempo performed (investigated through standard deviation). In these terms, the performer or instructor would learn that for this example, the performer was the least accurate andante, equally as accurate andante and moderato, and least precise allegro. These would be important metrics to continuously record to track the performer's progress in tempo and timing over a longer period of time.

7.5 Bow Articulation Technique Metrics and Statistics

The previous section evaluated a performer's ability to play at various tempi, and how the information could be visualized to inform the performer's practice. This section focuses on various aspects of how the performer plays various bow-strokes. Building off acoustic studies in (Askenfelt 1989), recent work in the field concerning bowing technique has focused largely on utilizing low and mid-level features to automatically classify bow strokes or articulations (Fiebrink 2011; Rasamimanana, Fléty, and Bevilacqua 2006; Young 2002; Peiper, Warden, and Garnett 2003). This research instead looks at high-level features extracted from different bow articulations and how they relate to the overall performance technique and the abilities of the performer.

7.5.1 DEFINITION OF BOW ARTICULATIONS

The three bow strokes played in D1 included *detaché*, *martelé*, and *spiccato*. While definitions may vary slightly, *detaché* is a stroke in which only one bow is performed per note, with equal weight (pressure) in between strokes. *Detaché* appears to mean detached, however, it does not mean detached in the typical sense (that the bow leaves the string), and some refer to it being detached in that there are no slurs between notes.

Martelé is a hammered stroke with a strong crisp bite at the beginning of the stroke, which is immediately relaxed through the remainder of the stroke.

Spiccato is a bounced stroke where the bow leaves the string. It is lighter than *detaché* and *martelé* and is often played at the balance point (center) of the bow.

7.5.2 BOW ARTICULATION: TEMPO ACCURACY

Building off of the previous section on tempo estimation, this section looks at the player's tempo performance when playing the different strokes in D1. This section first compares statistics from each stroke side-by-side in a box and whisker plot (or simply the boxplot). The boxplot is traditionally used in descriptive analysis and statistics to show a general distribution of a data set. It

does not necessarily show as detailed a distribution as other plots (e.g. the histogram), however, it is extremely useful in showing the tendencies of data sets, and when comparing multiple sets. The boxplot visualizes a five-number summary of the data set including the median, the quartiles, and the smallest and greatest values in the distribution (more on these shortly). In this way the boxplot is useful in summarizing the shape of a data set's distribution, its central value, and the variability in the set. It is useful for detailing outliers, or data that fall outside the normal range of the set, and so many general conclusions about the data can be drawn from a boxplot.

A box plot for data set D1 from recording #4 from the Ezither performer is shown in Figure 52. Each audio recording was roughly thirty seconds long and was windowed into three-second frames, with a twenty-five percent overlap between frames. The tempo was estimated once per frame. A three-second window was chosen to split the recording into roughly ten divisions, and to ensure three to six beats were included per tempo calculation. Two and four-second windows were also explored and showed similar relationships between stroke statistics.

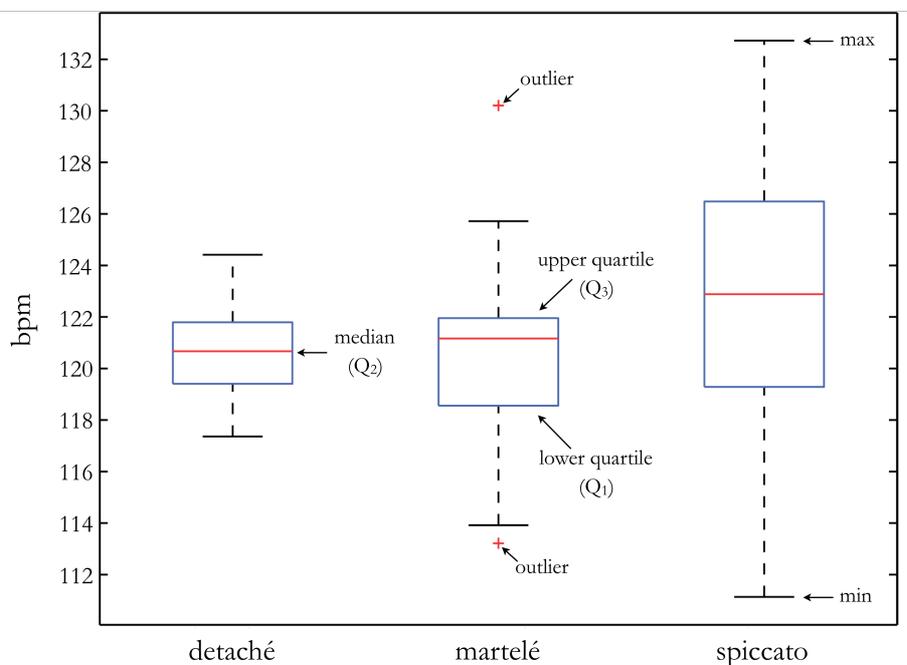


Figure 52: Box and whisker plot for Ezither recording #4, data set D1 showing bowing statistics of three bow strokes (detaché, martelé, and spiccato) when playing at the target tempo of 120bpm

The five-number summary displayed in the boxplot (Figure 52) provides insight into the data spread at a glance, making it useful for side-by-side comparison. Even without looking specifically at hard numbers, one can make a number of observations about the performances. Starting from the bottom of the box up, the *lower quartile* (Q_1) shows the point at which 25% of the data resides below. The *median* (Q_2) shows the point that splits the data set into halves (50%), and the *upper quartile* (Q_3) splits the highest 25% of the data, or the point at which 75% of the data rests below. These are also commonly referred to as the 25th percentile, the 50th percentile, and the 75th percentile (respectively), with the entire area spanning Q_1 to Q_3 called the *interquartile range*. Formally, the interquartile range is defined as

$$IQR = Q_3 - Q_1$$

and is a very useful measure which shows how spread-out the values are. In particular it shows how spread out the “middle” values are (where most of the data clusters), and is not heavily influenced by extreme values.

Interpreting the figure, *detaché* had the tightest (smallest) interquartile range, which also happens to fall within shortest distance of the target tempo (120 bpm). *Detaché*'s median was also closer to the target tempo than both *martelé* and *spiccato*. These results suggest that the performer's ability to play at the target tempo is greatest when playing *detaché*. *Martelé* strokes' interquartile range is smaller than *spiccato*'s, with the median falling closer to the target tempo, suggesting that in terms of tempo, the performer played *martelé* more accurately than *spiccato*.

Next the *whiskers* show the min and max values for each data set, and are marked accordingly on the figure. The closer the whiskers fall from the target tempo, the less the performer deviated below or above the target tempo. Again *detaché* was the most precise in terms of min and max tempi, followed by *martelé*, and then *spiccato*.

The last main component on the box and whisker plot are the red ‘+’ symbols which mark the outliers (if present). An outlier is a value that is one and a half times the length of either end of the box, essentially a value that does not seem to fit the data set. There were no outliers in either detaché or spiccato, however there were two outliers that were determined in the martelé recording. For reference, the evaluations made in this section can be verified by examining the actual box plot data provided in Table 21. From left to right each row in the table shows the “five-number summary” for the respective bow strokes tempo.

Table 21: Five number summary for each bow stroke in Ezither recording #4, data set D1 (target tempo = 120 bpm)

	Min	Lower Quartile (Q ₁)	Median (Q ₂)	Upper Quartile (Q ₃)	Max
Detaché	117.36	119.40	120.67	121.79	124.41
Martelé	113.22	118.56	121.16	121.96	130.21
Spiccato	111.14	119.29	122.88	126.48	132.72

Of course, the boxplot is not the only visualization that can be used to describe the player’s tempo performance under various bow strokes. The statistics (mean, standard deviation, and range), as well as the tempo curve that were evaluated earlier are still very useful in understanding the bow performance. The boxplot allows one to quickly visualize the spread of the data to draw conclusions about the relationships and skew of the data on a somewhat macro level, but what was the actual average tempo for each bow stroke?

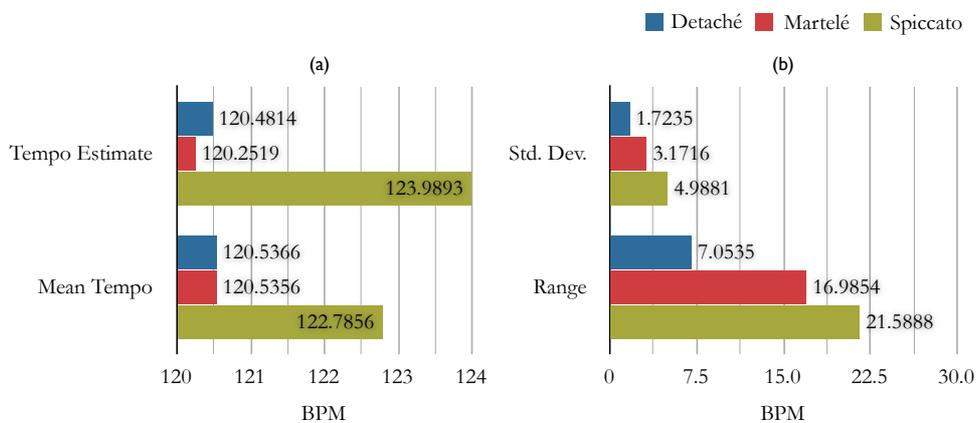


Figure 53: Tempo estimate and statistics (mean, standard deviation, and range) for detaché, martelé, and spiccato bow strokes for the Ezither recording #4 data set D1 (target tempo = 120 bpm)

Analyzing Figure 53, a number of conclusions can be drawn about the three different bow recordings. Average tempo is shown in (a) as both the estimated tempo calculated over the entirety of each performance, as well as the mean (average) tempo of multiple windowed tempo estimations. While the tempi estimated vary by only a small amount, the relationships between bow strokes are consistent. Firstly though, to test if the tempo performances for each bow stroke are statistically significantly different, a one-way ANOVA (Analysis of Variance) test can be used. Similar to the F -test used earlier, the one-way ANOVA tests if the means (of three or more groups) are equal (or statistically significantly different), by taking into account their variances. That is, the one-way ANOVA tests the null hypothesis

$$H_0 = \mu_1 = \mu_2 = \mu_3 = \dots = \mu_k$$

where μ is a group population mean and k is the number of groups. The alternative hypothesis (H_A) is that there are at least two group means that are significantly different from each other. The formula for the one-way ANOVA F -test statistic is defined

$$F = \frac{\textit{between group variability}}{\textit{within group variability}}$$

The “between group variability” is

$$\sum_i n_i (\bar{Y}_i - \bar{Y})^2 / (K - 1)$$

where \bar{Y}_i is the sample mean in the i^{th} group, n_i is the number of observations in the i^{th} group, \bar{Y} is the overall mean of the data, and K is the number of groups.

The “within group variability” is

$$\sum_{ij} (Y_{ij} - \bar{Y}_{i.})^2 / (N - K)$$

where Y_{ij} is the j^{th} observation in the i^{th} out of K groups, and N is the overall sample size.

As such, the F -statistic is

$$F = \frac{\text{between group variability}}{\text{within group variability}} = \frac{67.978}{13.727} = \mathbf{4.952}$$

The F -statistic is then compared with the critical value from the F -table, with between-group degrees of freedom (DFb) in the numerator, and within-group degrees of freedom (DFw) in the denominator, such that

$$DFb = K - 1 = 3 - 1 = \mathbf{2} \quad \text{and} \quad DFw = N - K = 113 - 3 = \mathbf{110}$$

Consulting the F -Table with $\alpha = 0.05$ and $F(2, 110)$, the critical value $F_{crit} = 4.192$, and the p value is $p = 0.009$. In summary, the null hypothesis is rejected and with strong confidence, at least one or more of the bow strokes tempos are statistically significantly different than the others.

The performer's best tempo performance was achieved almost identically playing martelé (120.25/120.53 bpm) and détaché (120.48/120.53 bpm), followed by spiccato (123.98/122.78 bpm) (see Figure 53a). Interestingly the technique required for both martelé and détaché are the most similar in the set, and when asked the performer said that he most commonly played détaché and martelé (or similar) in his repertoire, very rarely playing spiccato.

The one-way ANOVA and average tempo suggest stronger timing performance when playing détaché and martelé over spiccato; however, looking at just their mean tempi does not guarantee statistically significant difference between the performance means. The one-way ANOVA test performed is an *omnibus* test, that is, it does not say which groups are significantly different from

each other, only that at least two groups were. To do so a *post-hoc* test must be performed, such as the Tukey Test, also called the Tukey honestly significant difference (HSD) comparison. Similar to the *t*-test, the HSD is typically used in conjunction with the one-way ANOVA to find if the means between multiple groups are statistically significantly different. The test applies simultaneously to the set of all pairwise comparisons, comparing each group's mean to every other group's mean, $\mu_i - \mu_j$. The test statistic q is thus defined

$$q = \frac{Y_A - Y_B}{\sqrt{\frac{MS}{N_c}}}$$

where Y_A is the larger of the two means being compared, Y_B is the smaller of the two means being compared, MS is the within-group mean-squared, and n_c is the number of values in the sample (in instances such as this where the samples have different sizes, the harmonic mean is used). The results of the Tukey test are summarized in Table 22.

Table 22: One-way ANOVA multiple comparisons (Tukey HSD) for each bow stroke, Ezither recording #4 data set D1 (dependent variable = tempo)

(I) Stroke	(J) Stroke	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Detaché	Martelé	.00105	.89214	1.000	-2.1185	2.1206
	Spiccato	-2.24902*	.84598	.024	-4.2590	-.2391
Martelé	Detaché	-.00105	.89214	1.000	-2.1206	2.1185
	Spiccato	-2.25007*	.83914	.023	-4.2437	-.2564
Spiccato	Detaché	2.24902*	.84598	.024	.2391	4.2590
	Martelé	2.25007*	.83914	.023	.2564	4.2437

*. The mean difference is significant at the 0.05 level.

For the data in question (Ezither recording #4 data set D1), a Tukey post-hoc test revealed that the tempo performance between detaché and martelé bow

strokes were not statistically significantly different ($p > .05$), however the tempo performance between *detaché* and *spicatto* ($p = .024$), and *martelé* and *spicatto* ($p = .023$), were indeed statistically significantly different. This corroborates the earlier observation that the performer's best tempo performance, in terms of average tempo, was achieved almost identically playing *martelé* and *detaché*, followed by *spicatto*.

Looking at standard deviation adds more to story, and shows that in fact *martelé* varied from the average tempo greater than *detaché*. This can again be supported using the F -test, given the null hypothesis that there is no statistically significant difference between the standard deviations of the performers *detaché* and *martelé* bow strokes in this instance. Here $F = 3.3864$ (*martelé*'s variance divided by *detaché*'s variance). At the significance level $\alpha = 0.05$ and $F(35, 34)$, the critical value $F_{crit} = 1.7669$. F is greater than the critical value ($p = 0.00029$) and the null hypothesis is rejected. This reconfirms our previous observation in the boxplot (Figure 52), which showed a wider IQR (dispersion of values around the median). These measures of performance are further reinforced looking at the range (max – min), which was seen earlier in the boxplot by visualizing the spread of the whiskers. *Detaché* had a tempo range of 7.05 whereas *martelé*'s range was 16.98, and *spicatto*'s 21.58. In the earlier definition where performed tempo related to accuracy and standard deviation and range related to precision, for this particular recording *detaché* and *martelé* had nearly the same accuracy, however *detaché* had greater precision. Both were more accurate and precise (than *spicatto*) across the board and the performer could benefit from extra emphasis on *spicatto* bowing in his practice and performance repertoire.

7.5.3 BOW ARTICULATION: ONSET DIFFERENCE TIME (ODT)

The Onset Difference Time (ODT) is a feature explored earlier for drum performance (see 5.4.5) which compares the note onset times between sensors on the performer/instrument with the note onset time from the resulting acoustic output. The ODT is also a useful metric in bow stroke analysis as it captures a characteristic of the performer's performance with a particular bow

articulation, making it useful in both pedagogical situations as well as other contexts (e.g. bow stroke identification). This section looks at the former, detailing the ODT for different bow strokes performed by the Ezither performer and how the ODT may inform a player's practice.

Generally speaking, the accelerometer placed on the frog of the bow will detect a sudden jerk at the beginning of a stroke from stand still, or when the performer twists their wrist at the start of the succeeding note. The acoustic sound produced is determined by a number of factors, ranging from the weight placed on the strings, the location of the bow on the strings, and sometimes the speed (although a skilled performer can play fast or slow while maintaining control of dynamics). Before the sound is produced, the performer gestures the start of the bow stroke, and this section compares the onset of the gesture to the acoustic output as a characteristic feature of the performer's bow-stroke performance.

By subtracting the sensor onset time from the audio onset time it is possible to determine which onset preceded the other. A negative (-) ODT would mean that the sensor onset arrived earlier than the acoustic onset (rush), whereas a positive ODT would mean that the sensor onset was detected later than the acoustic onset (lag). The lag and rush times for *detaché*, *martelé*, and *spiccato* for recording #9 data set D1 are visualized in Figure 54, alongside the mean and standard deviation of the ODTs.

Overall the average ODT for each bow stroke was below zero (rush), meaning the sensor (accelerometer) onset was detected earlier than the acoustic onset. This seems likely when taking the twist of the performer's wrist between notes into consideration, and the fact that the performer sets the stroke in motion, and then pressure and other dynamic/timbre control are applied. Earliest rush was detected for the performer's *martelé* stroke, perhaps due to the fact that the performer must apply more pressure to the strings with the bow, affecting the gesture's velocity curve.

When the accelerometer onset lags the acoustic onset, the performer continued the head of the note past the note's start. Similar lag was detected for *detaché* and *martelé* strokes and both were greater than *spiccato*. Of the three strokes, *detaché* and *martelé* are the most similar, with *martelé* requiring the

easing up of pressure after the head of the note. This may account for the slightly lower maximum lag time vs. *detaché*, and the slightly earlier (earliest) rush time resulting from the sudden direction and pressure change between strokes.

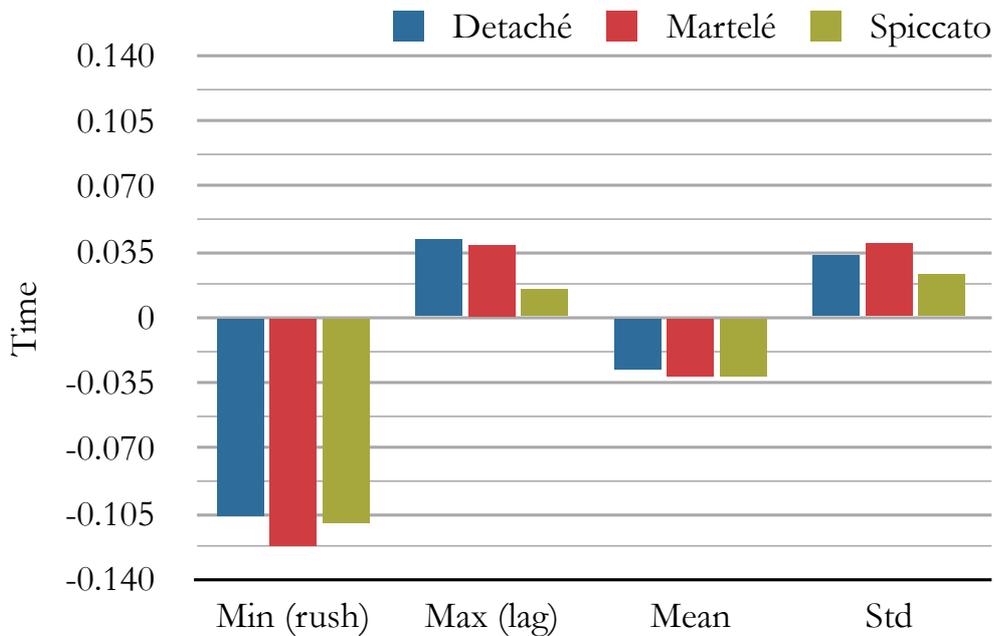


Figure 54: Onset difference time (ODT) statistics for recording #9 data set D1

Overall *spiccato* performance was the most regular in ODT when compared to the other two strokes. *Detaché* was slightly more regular than *martelé* and the performer could use these results to focus his practice to minimize the ODT or standard deviation through practice. In the future, further analysis into the ODT of expert performers would be useful to understand how the ODT contributes to the expressive qualities and acoustic output of skilled performers, and as a useful feature in other tasks such as bow stroke recognition.

7.5.4 BOW ARTICULATION: ARTICULATION ATTACK SLOPE

In addition to the onset difference time, another useful bow gesture metric is the (attack) slope of the bow articulation acceleration curve. Previous work by (Rasamimanana, Flety, and Bevilacqua 2006) parameterized min/max velocity

and acceleration for bow stroke classification using accelerometers, and demonstrated a strong bond between gesture bow articulations and velocity/acceleration. The work in this section parameterizes the slope of the curve leading up to the accelerometer note onsets, which we call the Articulation Attack Slope (AAS).

Following the audio attack slope detection strategy in (Lartillot 2011) the AAS is computed as a ratio between the magnitude difference between the start (local minima) and ending (local maxima/onset) of the attack phase, and the corresponding time difference. Figure 55 displays the entire attack phases of the detected AASs for a single *detaché* recording as the red lines in between onsets and their preceding local minima (valleys). The top of the figure shows the attack phase for AASs detected for the entire recording and the bottom of the figure displays a six-note excerpt between 10.0-seconds and 12.8-seconds.

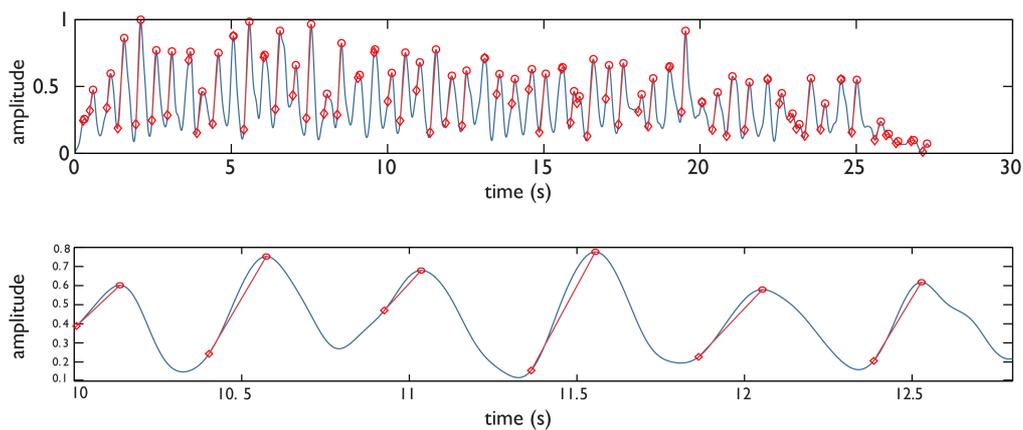


Figure 55: Note attack slope for Ezither recording #9 data set D1, *Detaché* entire recording (top), 2.8 second window from 10sec – 12.8sec (bottom)

The actual AAS value as previously described is the ratio between the valley-onset magnitude difference, and the corresponding time difference. Figure 56 provides the average and standard deviations of the AAS values for each bow articulation.

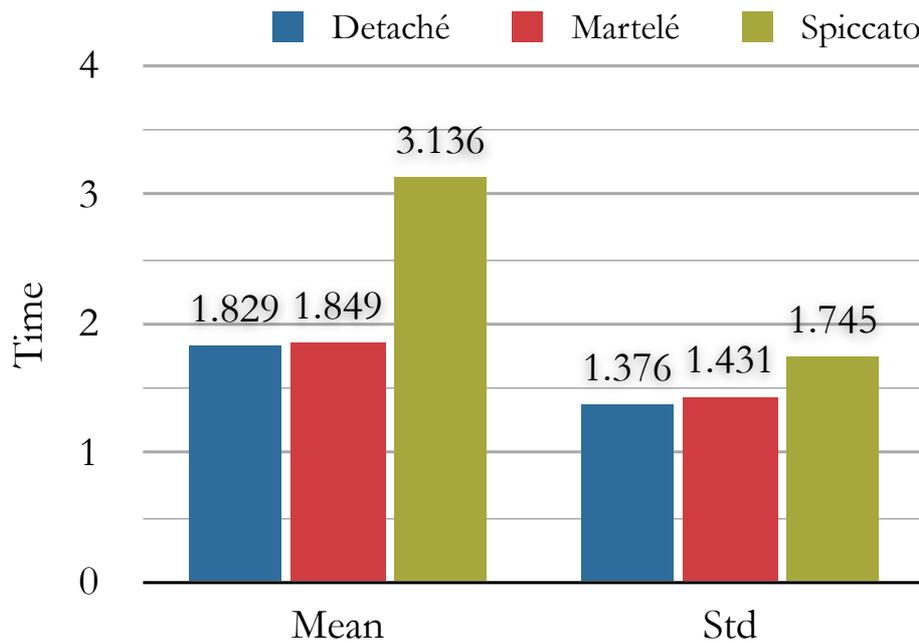


Figure 56: Mean and standard deviation of bow stroke attack slopes for Ezither recording #9 data set D1 detaché, martelé, and spiccato

7.6 Long-term Metrics Analysis

In the previous sections various bow performance measures were explored across multiple modalities; the ultimate goal was to capture performance metrics (and their differences when compared to the ideal target performance) that could be used to help focus the performer's practice. Timing metrics tracked how accurate the performer's timing was at three tempi (andante, moderato, and allegro), as well as the performer's tempo accuracy evolution over the length of a recording. Bow articulation metrics were also explored, including the performer's tempo accuracy for three bow articulations (detaché, martelé, and spiccato), bow-stroke and acoustic onset difference time, as well as the articulation attack slope. All of the metrics and derived statistics were visualized in various ways to inform the performer about their playing over the individual performances and recordings. In this section, the development of the performer's playing is observed by examining similar metrics and statistical measures over the course of seven months.

7.6.1 LONG-TERM TEMPO METRICS: AVERAGE

In looking at the performer's progress over the seven months in which he recorded his practice, his average tempo (for all three tempi performed in a practice session) naturally deviated from the goal target tempo. Looking at the performer's tempo averages for each pass over time it can be concluded that the performer tended to play slightly faster than the target tempo. This can be seen in Figure 57, which shows the average tempo for all data set D2 tempo recordings (over the entire corpus of practice sessions). The average tempo for all andante recording sessions is 81.25 bpm, 110.61 bpm moderato, and 140.21 bpm allegro.

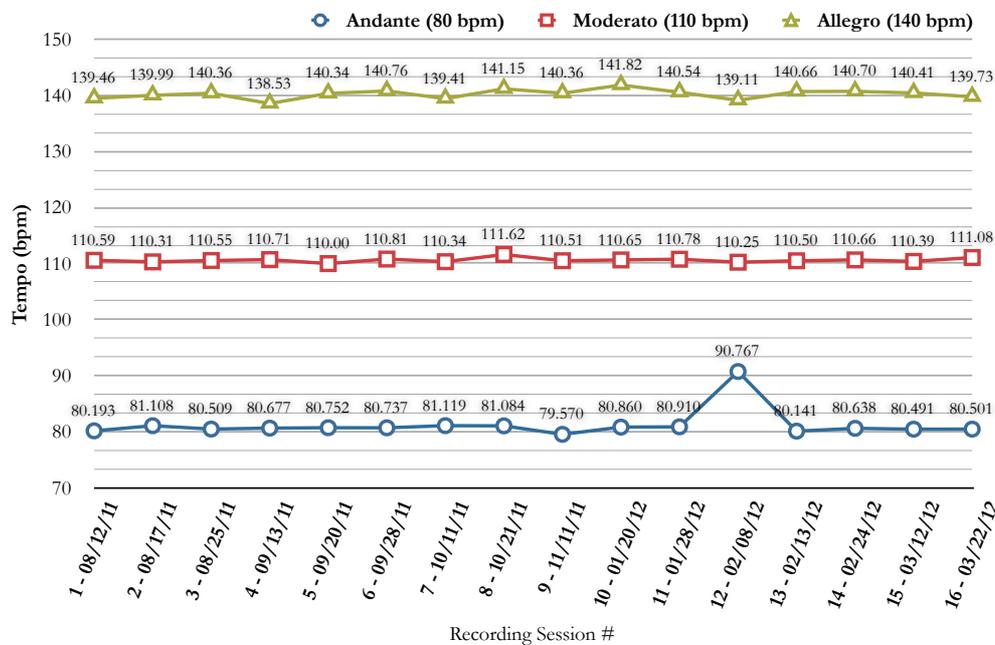


Figure 57: Average (mean) tempo of each D2 tempo recording (andante – bottom, moderato (middle), allegro (top), from the entire data corpus 1-16

7.6.2 LONG-TERM TEMPO METRICS: STANDARD DEVIATION

Evaluating the performer's strongest or average tempo over a performance reveals the performer's general ability to play at the desired tempo without knowing about how consistently the performer actually performed at the tempo. In fact the average tempo achieved across all three tempi from the earliest

recording sessions through the later sessions are actually very close in value. Merely looking at average tempo in this way would suggest very little progress (if any) was made in terms of tempo improvement over the course of the performer's practice. Thus in order to gain more meaningful insight into the performances, it is necessary to add a temporal component to the analysis. This is similar to section 7.4.3 in which tempo graphs illustrated the tempo evolution of the performance, as well as various statistics determined from the evolution. In this section we will revisit the standard deviation of tempo over a given performance, while placing the singular performance's standard deviation within the context of the entire collection of D2 recordings. In this way it is possible to not only capture the average amount of dispersion from the average tempo over a singular performance, but also how the performer's ability to perform at a consistent tempo changes with time and practice.

Figure 58 shows the standard deviation of tempo in data set D2 for all practice sessions recorded by the Ezither performer. In general, smaller standard deviation means that the performer played with a higher consistency or less variation in tempo; earlier we loosely referred to this as "precision" or the ability to steadily play at a given tempo (whereas accuracy is defined in this context as the ability to play as close as possible to the target tempo). The blue line shows the standard deviation achieved for each practice session for *andante* tempo, red for *moderato*, and green for *allegro*.

As visualized in the graph, the performer almost always exhibited the smallest standard deviation and most consistent tempo when playing *andante*, followed by *moderato*, and finally *allegro* (exceptions include practice sessions #6, #9, #11, #15). The average standard deviation for *andante* over all practice sessions was 2.32, 2.40 *moderato*, and 3.53 *allegro*. This suggests that the performer exhibited less consistency in speed and timing the faster he played and was the most precise when playing *andante*. This is in accord with the previous results discussed in section 7.4.3.

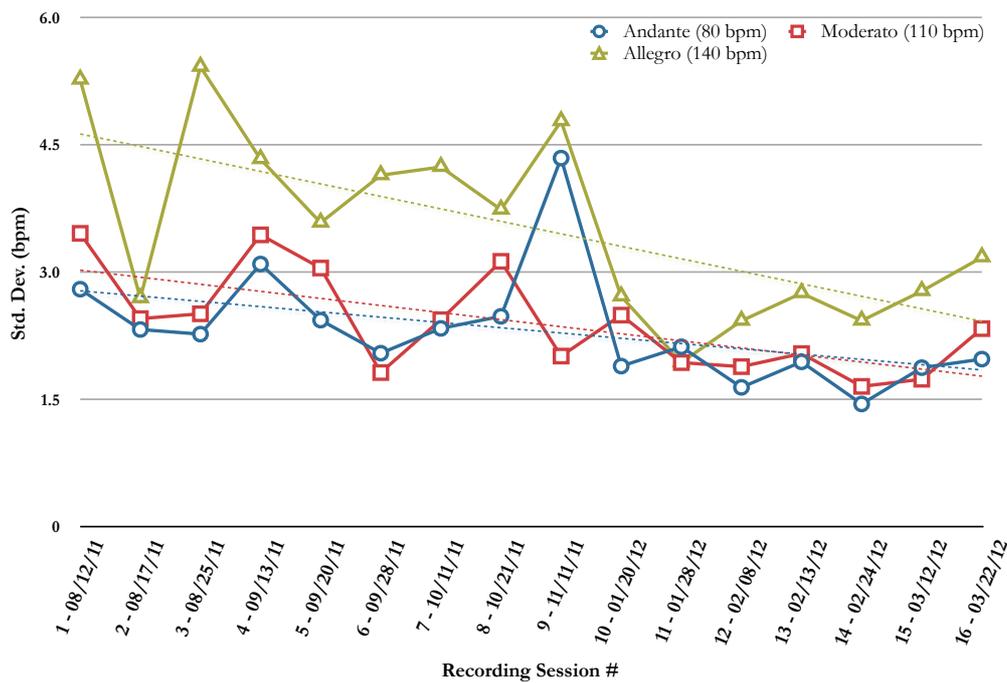


Figure 58: *Standard deviation* for every session data set D2 tempo (solid) and linear trend lines (dashed)

To verify this observation, again the F-test was used to test the significance of andante's average standard deviation vs. allegro's average standard deviation. Similar to 7.4.3, the two-tailed null hypothesis is that there is no significant difference in standard deviation when playing at andante speed vs. allegro speed. Thus, the F-value is calculated using andante's and allegro's variance

$$F = \frac{3.53^2}{2.32^2} = 2.3151$$

At a significance level of $\alpha = 0.1$, F is greater than the critical value of 1.9280 ($p = 0.051$), and the null hypothesis is rejected. To summarize, the average standard deviation of the player's performance at andante speed (across all 16 recordings sessions) may be regarded as statistically significantly different than the average standard deviation when playing allegro. Note, given the limited sample size and the nature of the test, a significance level of $\alpha = 0.1$ was

deemed acceptable, although the p value is almost significant at the more stringent $\alpha = 0.05$.

An interesting observation when looking at Figure 58 is that with a few exceptions, all three tempi exhibit similar change in standard deviation metrics from practice session to practice session. Between sessions one and two all three tempi lower in standard deviation and then rise from two to three. Andante and allegro both continue to raise between sessions three and four, and all decrease from four to five. Andante and moderato both continue to decrease between sessions five and six, and all increase in standard deviation between sessions six and seven. Again andante and moderato move similarly between sessions seven and eight, and then andante and allegro follow the same trend in standard deviation from session eight to ten. Moderato and allegro follow similar movement from ten to twelve, and from twelve through sixteen all exhibit similar movement in standard deviation—increasing from sessions twelve to thirteen, decreasing between thirteen and fourteen, and slightly rising again between fourteen and sixteen.

The tight inter-tempo standard deviation relationships between practice sessions may suggest the potency of various factors on practice metrics such as routine and consistency of practice (how many times the performer played and/or practiced between recorded practice sessions), physical parameters (not practicing enough to keep muscle memory active or practicing too much or with improper form), time constraints, mental focus, and other external factors, etc. Most importantly, the clear link in session-to-session metrics shows that the change in the performer's metrics are consistent across all three tempi, showing a progression in performance metrics.

The progression in standard deviation was analyzed from session-to-session; however, one of the most insightful observations emerges when viewing the session-to-session standard deviation within a more macro scope. The dashed line overlaid on top of each tempo in Figure 58 shows the linear trend line for the tempo's standard deviation over time. As shown in the figure, over the seven months in which the performer started playing and practicing the Ezither, his tempo precision (as measured by standard deviation) had steadily improved. The

steepest slope (or highest improvement) in terms of sheer magnitude of improvement was achieved for allegro tempo, which was previously concluded to be the least precise in the example in 7.4.3 (which still remains true). Andante and moderato has increased at a similar pace, however at the current rate, moderato might actually surpass andante in terms of precision. This information is extremely useful for a practicing musician or educator to visualize and understand, in order to tailor practice-to-practice sessions specifically to the performer's needs.

7.6.3 LONG-TERM TEMPO METRICS: RANGE

Closely related to standard deviation is another statistic examined previously called range. Here range describes the distance between the min and max tempos estimated for each framed performance (recording session), showing the overall width of tempi (highest tempo subtracted by the lowest tempo) estimated for each pass. Whereas standard deviation measures the average variation per performance, range measures the total variation per performance. Like standard deviation, smaller range means less dispersion and more precise performance (in terms of tempo).

As shown in Figure 59 range doesn't exactly follow the same curve as standard deviation, however it's values are closely related and so the curves between the two over time are reminiscent of one another. Again, the link between particular recording sessions and practice outcomes can be seen where various tempi exhibit similar changes in range between recording sessions. Between sessions one and two all three tempi decrease in range, and from sessions two through eight andante and moderato follow similar changes. Between sessions eight and ten andante and allegro change similarly (first increasing and then decreasing in range), and from ten to eleven all three decrease. From eleven to thirteen andante and moderato increase and decrease in the same directions, and from thirteen through sixteen moderato and allegro change in range together. All three tempi change in range similarly between practice sessions fifteen and sixteen.

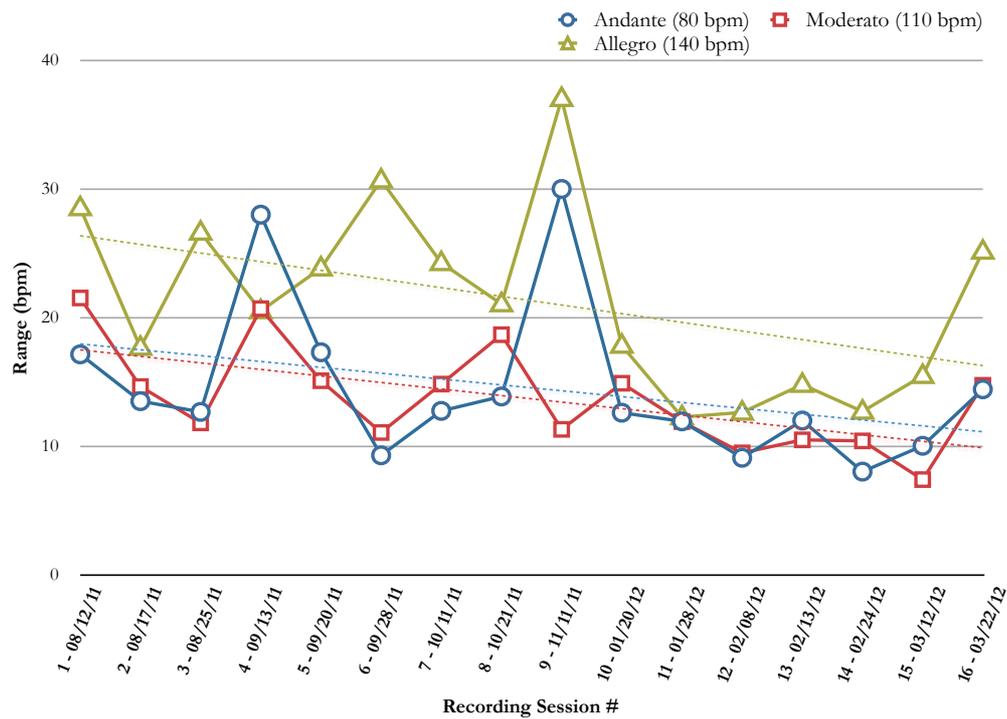


Figure 59: *Range* for every session data set D2 tempo (solid) and linear trend lines (dashed)

Again the gradual increase and decrease in range becomes meaningful when stepping back, exposing that the range steadily decreases as the performer continues to practice. This means that over time the performer improved at all tempi, reducing the average variation in tempos during practice. And while standard deviation showed that the performer was the most precise when performing andante, one can see from the figure that moderato actually has a slightly tighter range than andante, at the expense of a slightly less consistency (deviation) over time per performance. As a general measurement, the average range is shown in Table 23, showing that moderato had the smallest range (13.68), followed by andante (14.53), and lastly allegro (21.30).

Table 23: Average Range of tempos from D2 for all data collected

	Andante (80 bpm)	Moderato (110bpm)	Allegro (140 bpm)
Range	14.53	13.68	21.30

Becoming a proficient musician requires the ability to accurately and precisely perform over a wide range of tempi. The ability to further subvert or “push and pull” tempo and speed at will is an important characteristic of nuanced performance and is a trait most musicians spend years honing (consciously and subconsciously). During practice it would be useful for musicians and educators alike to have a window into one’s tempo performance, and its evolution over time in various time scales. Combining the performer’s average, standard deviation, and range in tempo over a wide window of time begins to paint a detailed map of the player’s tempo performance, and can inform and help focus ones understanding of their performance, style, and practice.

7.6.4 LONG-TERM BOW ARTICULATION METRICS: TEMPO ACCURACY

Also useful is the performer’s ability to perform various bow stroke articulations. Bow articulation performance was explored earlier in this research; first in section 7.5.2 to gain insight into the Ezither performers tempo performance when playing different bow articulations over a single recording. These statistics will be revisited in this section, over the entire corpus of the musician’s practice sessions recorded. In this way, it is possible to investigate how the performer’s timing had progressed for the three articulations practiced (detaché, martelé, and spiccato) over the course of his training. For all statistics, the estimated tempo was calculated by windowing the articulation practice session recording every three seconds with 25% overlap between frames.

Table 24: Tempo, range, and standard deviation averages over all practice sessions

	Detaché	Martelé	Spiccato
Tempo	120.67	120.60	120.61
Std. Dev.	2.35	2.22	2.64
Range	11.28	9.61	12.24

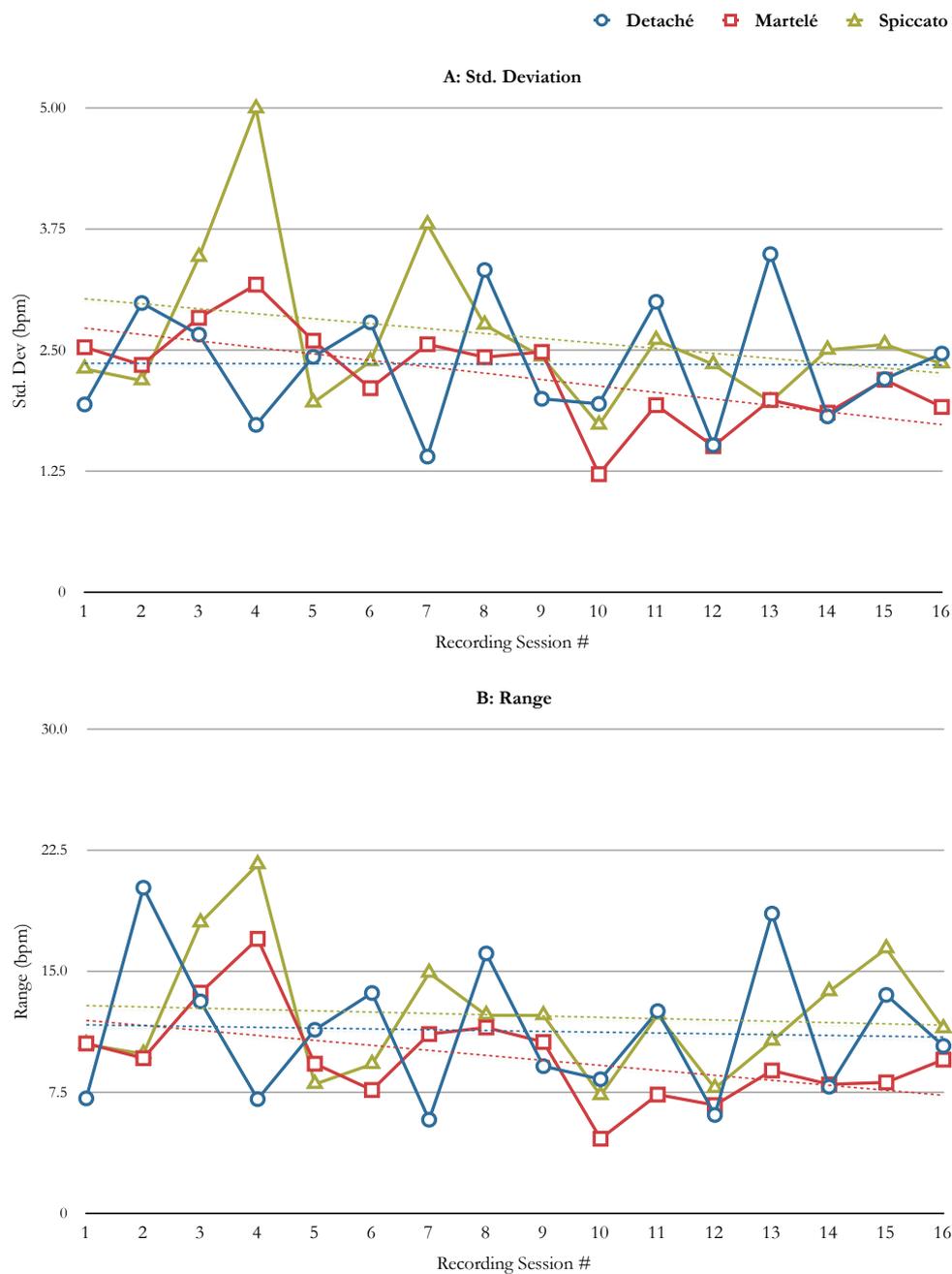


Figure 60: Standard deviation (A – top) and range (B – bottom) of bow articulation tempo across all D1 data sets collected

The statistics are displayed in Figure 60 and show the standard deviation (top) and range of estimated tempos (bottom) over the sixteen practice sessions for each bow stroke articulation. Also shown are linear trend lines indicating the overall slope of the articulation tempo's standard deviation and range. For the performer's detaché stroke there seems to be little improvement (blue horizontal

dashed line); the performer hovers around the standard deviation average of 2.35 bpm and average range of 11.28 bpm.

Previously from recording #4 alone this research concluded that the best tempo performance was achieved when playing martelé, although détaché was slightly more consistent (smaller standard deviation and range). This was confirmed by the distribution shown in the box and whisker plot (Figure 52) and then in Figure 53. While the analysis is true for recording #4 on its own, in looking at the statistics over time we see a slightly different picture painted. As détaché's timing remains mostly consistent, the performer's tempo for both martelé and spiccato continue to increase in tightness over time.

By practice session #16 the performer's average standard deviation and range of his martelé stroke is now generally smaller than his détaché articulation (which was not the case earlier). His tempo performance while playing spiccato (previously far behind the other two articulations tempo accuracy) has gotten much closer to his tempo performance when bowing détaché and martelé (around session #8). Clear improvement has been made for both martelé and spiccato bow articulations, and by session #16, the performer's tempo for all three articulations have become more consistent and similar to one another.

The trend lines in Figure 60 suggest that by session #16 the performer has the strongest (most consistent or precise as defined earlier) tempo performance when playing martelé. Looking at the average tempo, standard deviation, and range calculated over all sessions confirms that across the board, martelé is the strongest articulation in terms of tempo performance. However, all three articulations exhibit extremely similar characteristics, especially in average tempo and standard deviation; this shows that the Ezither performer has improved all three articulations such that he has near equal (tempo) performance for all of them.

7.6.5 LONG-TERM BOW ARTICULATION METRICS: ONSET DIFFERENCE TIME

In addition to reviewing bow articulation tempo performance over time, it is useful to investigate the Onset Difference Time explored previously in 7.5.3 as a

characteristic metric of each bow articulation. Observing statistics of a particular bow stroke's ODT, the research showed the average ODT and its variability for a given articulation and performance. The ODT also showed how much the performer may have lagged or rushed the beat for the particular articulation and practice session recording. Thus it is useful to evaluate the ODT for each bow articulation over time, in order to evaluate the usefulness of the measure and how it can inform the performer's practice.

Figure 61 shows the session-to-session difference between the average ODT for each practice session, for all (three) bow articulations performed. A smaller delta between sessions means that the ODT remained more consistent between sessions.

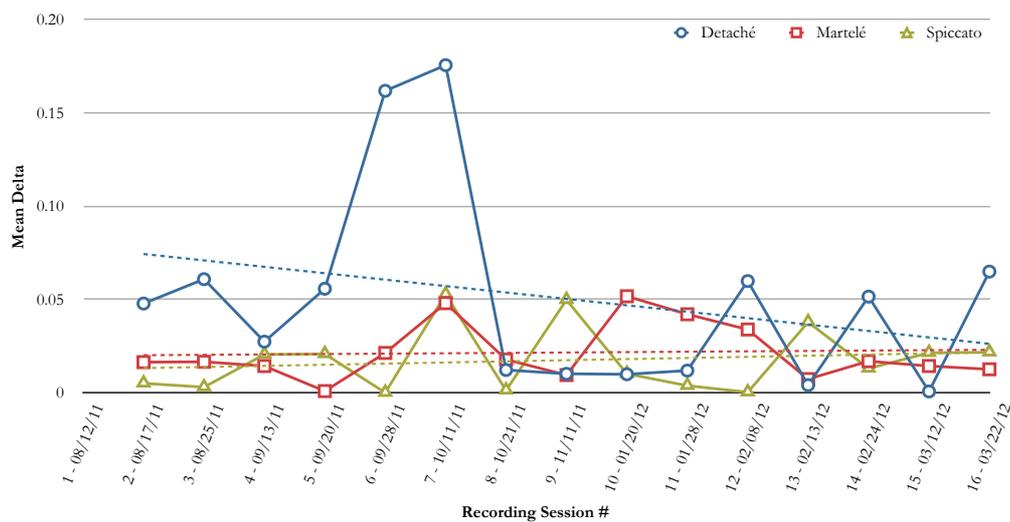


Figure 61: Session-to-session change in Ezither articulation Onset Difference Time

As illustrated in the figure the difference between average ODTs from session to session was very close for both martelé and spiccato strokes. This can infer that (from the start) the particular onset properties of the performer's physical and acoustic actions remained regular. This is also true for the performer's detaché stroke the majority of the time, except between practice sessions five and eight. If the performer was aware of this at the time of practice, for example during practice session #6, he may have placed more focus or emphasis on his detaché stroke, to target the consistency of his detaché playing.

7.6.6 LONG-TERM BOW ARTICULATION METRICS: ARTICULATION ATTACK SLOPE

In this section we revisit the Articulation Attack Slope, a metric that measures the acceleration slope of the bow articulation gesture. As the nature of the physical gesture's attack slope may change slightly between performer and/or playing style, this research does not compare the performer's AAS against a target attack slope for the particular bow articulation; rather it investigates the consistency of the performer's gesture over time.

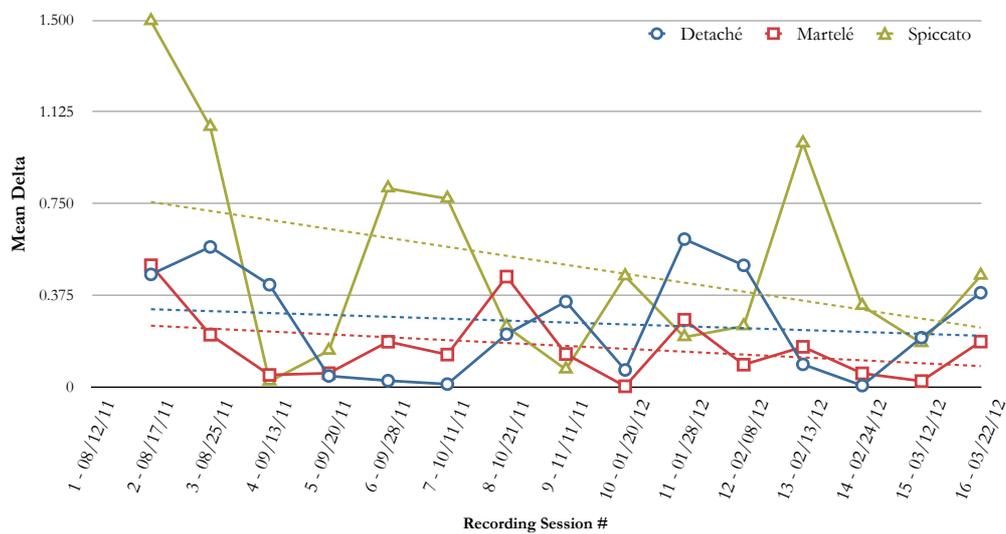


Figure 62: Ezither average articulation attack slope difference over time for (AAS difference – solid, trend lines dashed)

Theoretically as the performer's technique improves the average AAS for each articulation recording should homogenize. Essentially the performer's technique should become more consistent, leading to a regular AAS when performing a particular bow articulation. To investigate this relationship the delta in average AAS between successive practice sessions is examined and displayed in Figure 62. As expected, the difference in the average AAS in the earlier practice sessions is generally greater than in later practice sessions. The dashed trend lines show that for each of the three articulations practiced, the performer's technique improved in terms of consistency of the gesture's AAS.

The performer's best stroke (in terms of AAS regularity) was martelé, followed by détaché, and then spiccato. Greatest improvement over the sixteen practice sessions was achieved for spiccato, as illustrated by the steepest slope of the three trend lines. Martelé was the best stroke and also improved slightly greater than détaché (as illustrated by its steeper trend line). Interestingly, these characteristics mirror some of the characteristics discovered previously in the bow articulation tempo studies concluded in section 7.6.4. As in the previous tempo studies, spiccato was the weakest articulation, albeit showing much improvement, martelé was the strongest performer overall, and détaché was a strong stroke for the performer but showed the least amount of improvement over time.

Inevitably there will be variation in the AAS every time a performer plays a particular bow articulation. To further measure the consistency of the performer's (physical) technique, one can also look at the change in standard deviation and range of the AAS between practice sessions. Just as the delta in average AAS regularize more over time if the performer's technique improves (Figure 62), the range and standard deviation of an articulation's AAS may also become more regular over time (hopefully decreasing).

As illustrated in Figure 63, this is the true for the Ezither performer. In terms of standard deviation of AAS, the performer's AAS standard deviation for both martelé and spiccato regularize over time. When performing détaché, the performer's AAS standard deviation remains fairly consistent, which also resembles earlier results in both change in average AAS for détaché (Figure 62), as well as the particular articulation's standard deviation and range of tempo performance (Figure 60). Martelé and spiccato however become more regular over time in terms of AAS standard deviation, and all three articulations regularize in AAS range between sessions. These measurements are useful signifiers to the musician and his instructor about his overall progress and uniformity of the physical motion of his bow stroke articulations.

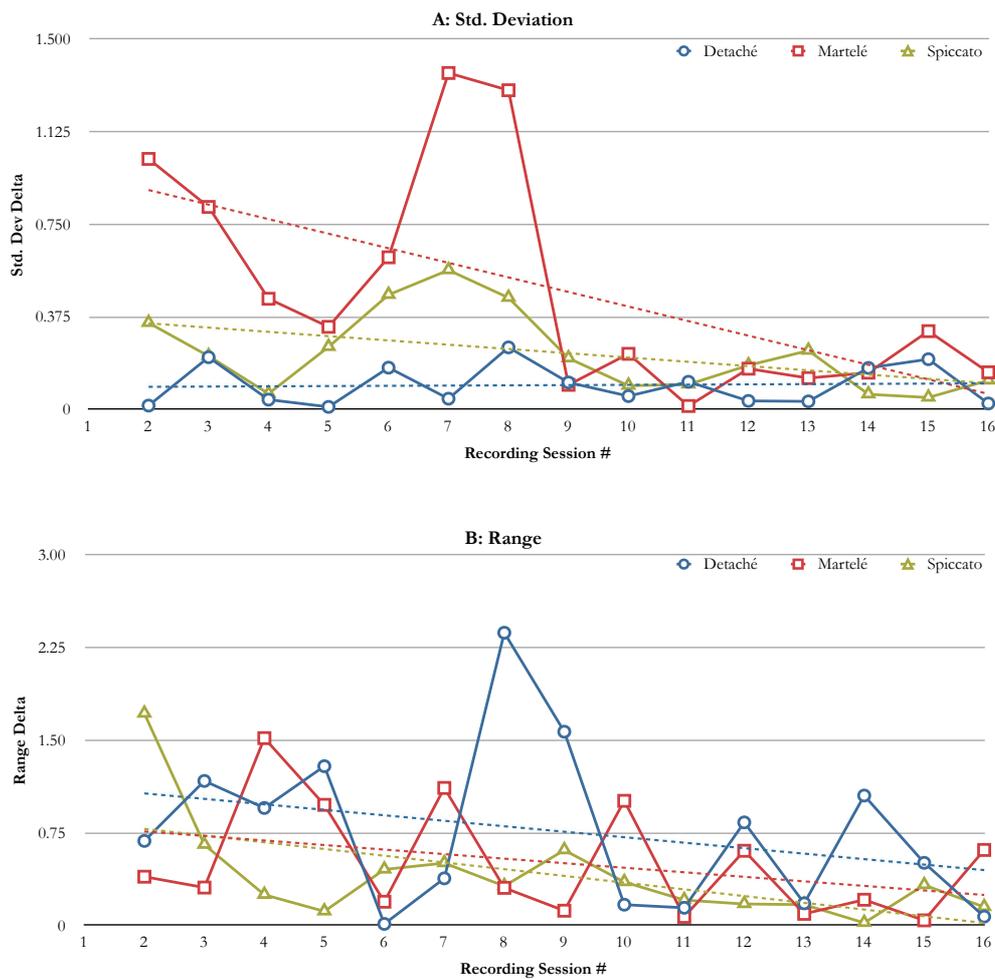


Figure 63: Ezither articulation attack slope standard deviation (top) and range (bottom) practice session-to-session difference over time for (AAS difference – solid, trend lines dashed)

7.7 Summary

There is no doubt that the role of technology in the practice room will continue to permeate the ways in which musicians and musical educators learn and teach. Interactive systems and computer assisted musical development have already been integrated into the everyday curriculum of music schools the world over. While current systems work satisfactorily for certain aspects of musical training, no readily available or widely-used system currently specializes to the individual needs of the performer, or the musical semantics of their particular instrument.

Musical performance is highly individualized in nature, and traditionally a musician learns to play in a formalized contract between the teacher (mentor, guru, master musician, etc.) and the student. Commonly the amount of time a

musician spends practicing alone compared to the amount of time they spend practicing within the guidance and presence of their instructor is often less than ideal. Thus, computer assisted practice offers great potential in helping musicians practice with greater understanding and focus, especially when practicing independently.

In order to enable effective and nuanced channels of understanding between musicians and the computer, this research argues that analysis in a number of modalities is necessary. In particular this chapter focuses on string performance, and some of the possibilities when combining analysis of the acoustic signal of the input with gesture data from an accelerometer in the bow of the instrument. By exploiting the information present in both modalities, this research attempts to highlight useful characteristics from the performers practice, including timing and tempo metrics and statistics, as well as information about the performers ability to play various bow articulations. The metrics and statistics are evaluated at various time-scales, obtaining useful performance metrics not only in individual practice sessions, but also over a seven-month period in which the performer learned to play his instrument, having never played the instrument before.

In analyzing the Ezither performer's practice a concise set of statistical measurements and visualizations are presented. There are many other features, statistical measures, and visualization techniques that can be observed and provide useful information about performance. However, this research chose to focus on the following selection of common statistical tools for a number of reasons: (1) Statistics, (1a) Min, (1b) Max, (1c) Range, (1d) Standard Deviation, (2) Visualizations, (2a) Bar graph, (2b) Line plot, (2c) Box and whisker plot.

Firstly, the experiments and analysis tools proposed should not require trained mathematicians or scientists to be used and understood. As the ultimate goal is to eventually support these metrics in the regular practice room or bedroom of practicing musicians, there was a strong desire to keep the metrics and visualizations as simple and straightforward as possible. Relational observations are also desired, so many of the visualizations presented were chosen as they highlight certain musical performance relationships, for example

the performance differences between multiple tempi, or between various bow articulations.

While much of the discussion thus far has been under the scope of informing the musician about their practice, analysis tools as described in this chapter would also greatly benefit the educator. In observing the contract between teacher and student, the teacher's role is to guide and nurture the student into honing their skills. By effectively identifying the strong and weak areas of the student's practice, educators can best target and focus their limited time with their students. One exciting area to explore in the future would be combining multimodal musical performance metrics with practice content generation, an interesting concept being investigated by (Percival and Schloss 2008).

Lastly, this chapter presents results calculated after recordings were collected; however, these techniques are feasible in real-time. The statistical measurements and visualizations presented are computationally lightweight, and could very well run on today's computer, laptops, and other mobile devices such as the Apple iPad or iPhone. One benefit of a future real-time system would be that it creates a useful feedback loop—musicians and educators won't only be able to see performance data in between lessons, but also *during* lessons, providing dynamic information to influence practice, in real time.

There is of course lots of additional room to continue exploring multimodality in the practice room, and many more families of instruments to reach. This research has also explored multimodal drum performance metrics previously in this dissertation (refer to section 5.5), and so this section has focused on string instruments as to cover a broad body of traditional western instruments. While multimodality in the practice room is still very new, this research hopes to show a glimpse of exciting possibilities of applying multimodal signal analysis to the everyday routine of practicing musicians.

Chapter 8

Conclusion

Summary and Conclusions of the Presented Research

The overarching goal of this dissertation has been to explore the musical affordances of multimodal HCI. Early on it became evident that exploring this goal would require at least three vital steps in the process. The first was identifying valuable aspects of musical interaction in which to capture, and which motivate the crosspollination of multimodal techniques and musical scenarios. Secondly, capturing the heterogeneous information must also be efficient, in order to make the benefits of multimodality applicable to a large audience. Lastly, it was important to evaluate the outcomes of the techniques. To that end, this research has presented a variety of work that outlines our process, and the affordances garnered from applied multimodality in the face of musical interaction.

8.1 Summary

This research identified two specific areas of musical interaction that can benefit greatly from multimodality: musical practice and pedagogy; and secondly, live performance. Examples were provided that demonstrated the viability of multimodal techniques in these scenarios, and established its significance in future musical interactions. To facilitate multimodal communication, this work developed a valuable software tool that made it possible to easily acquire data from heterogeneous sensor systems and musical instruments. This tool was used to capture multimodal data in a number of different musical scenarios, and enabled an assortment of research investigations in which multimodal analysis was applied to the domains of machine musicianship, machine learning, and real-

world performance. As a result, this research enables many new possibilities in musical interactions, including, the empowerment of musicians and educators to better understand ones playing; and secondly, facilitating new modes of musical expression in performance.

8.2 Primary Contributions

The remainder of this section provides a summary of the primary contributions presented in this work—specific research examples that we believe will significantly add to future musical interactions, both in the practice room, and on the stage. The primary contributions of this research include:

1. A new software tool that enables anyone to easily access and record heterogeneous data from multimodal instruments and sensor systems.
2. Two performer recognition examples that establish the importance of multimodal analysis for understanding the intricacies of musical performance.
3. Investigations into multimodal percussion analysis including:
 - a. Enabling the computer to automatically detect left and right hand hits from drummers.
 - b. Detailing useful multimodal performance metrics (e.g. Onset Difference Time).
4. A study of how multimodal techniques, such as multimodal fusion, can help improve machine musicianship tasks when unimodal techniques fail, exemplified through the core task of onset detection.
5. Investigations into the future of multimodal musical practice including:
 - a. Designing multimodal systems that can be used daily by practicing musicians.
 - b. Conceiving useful multimodal metrics to help inform musicians and musical educators about progress and performance (e.g. Onset Difference Time).
 - c. Demonstrating useful multimodal data visualizations.

8.2.1 ENABLING MULTIMODAL MUSICAL ANALYSIS WITH NUANCE

Nuance is a software tool used throughout this research to facilitate multimodal analysis of musical performance. A fundamental strength of multimodal interaction is that the systems can be highly specific to the instrument, and sympathetic to technique and task. Unfortunately, this results in the need for tailored solutions to capture the data from individual instruments and sensor systems. This significantly inhibits the applicability of multimodal analysis in real-world scenarios, and significantly slows down the research process. Nuance begins to address these issues, and is the first software program of its kind that enables one to tap into multimodal musical systems with little to no programming or patching. Nuance can be used by anyone, whether musician or researcher, and supports nearly any musical input through a combination of audio, MIDI, Open-Sound-Control, and direct serial I/O.

8.2.2 TEACHING THE COMPUTER TO KNOW WHO YOU ARE

Using Nuance, another primary contribution of this dissertation presented the first work in multimodal performer recognition. Research into multimodal performer recognition served two main purposes. The first was concerned specifically with the task of performer recognition itself—having the computer automatically detect who a performer is from learned performance data. The second was concerned with the holistic belief of this research, in that important detail of a player's performance is not only in the acoustic output (or purely time and velocity based information from MIDI or symbolic data sources), but also within the physical actuation of performance (and the relationships between both domains). Thus, performer recognition was presented for both string (sitar) and percussion (snare drum) performers, and used a set of low-level features from audio and various sensors. This work not only showed improved performer recognition rates when a multimodal approach (vs. a unimodal approach) was employed, but also the strong bond between the physical and acoustical domain of musical performance. Investigating this link has been a tremendous motivating force behind this research in whole.

8.2.3 NEGOTIATING NOVEL UNDERSTANDINGS AND INTERACTIONS IN DRUM PERFORMANCE

To further explore the links between the physical and acoustic domains, this dissertation presented work in multimodal drum-stroke computing. Drummers spend years training to achieve dynamic control over both of their hands; the ways in which drummers orchestrate their hands while playing, and control the dynamics of their strikes, is crucial to drum performance and practice. Unfortunately, from the perspective of an audio input alone, the computer has no way to tell which hand a performer is striking a drum with. This severely stunts applications and research into drum performance, as manually labeling data is often too time consuming, and is not robust to errors within the collected data set. To overcome this, this research examined multimodal surrogate data training. This technique used gestural data from accelerometers to automatically label acoustical onsets (note strikes). Next, this research presented the first work in automatic drum hand recognition, showing that that the machine can be trained to accurately recognize strike hand from a player pool of ten drummers ranging from beginner to advanced skill levels. Further investigating the strong associations between the physical and acoustical domains of drum performance, this work also introduced the first explorations in multimodal Onset Difference Time—a simple statistical measure which compares the onset times between audio and sensor onsets, and provides insight into the performance of the strike.

8.2.4 ADVANCING MACHINE MUSICIANSHIP THROUGH MULTIMODAL FUSION

All listening organisms have the ability to differentiate when sound events occur. Musically speaking, humans are exceptional at detecting when a musical event begins, when it ends, segmenting which instrument in the group or body it emanated from, etc. Thus, at the core of most machine musicianship and machine learning scenarios is onset detection—the task of detecting when musical events actually occur. When sounds have strong transients, such as in the drum-stroke computing experiments, onset detection algorithms work very well.

However, in instruments with slower or smeared attacks, such as bowed string instruments, there are many situations where this task becomes increasingly difficult (e.g. tremolo playing). Because much of this research and real-world applications are interested in also investigating the performance of bowed instruments, a solution for robust onset detection is needed. To this end, this dissertation presented the first work in multimodal onset fusion for bowed instruments. The onset detection fusion algorithm presented is a late-fusion process, meaning it is algorithm independent, and can fuse audio and sensor onsets from any (current or future) onset detection algorithm. Using the fusion algorithm, the high accuracy of the sensor onset detection was achieved, while maintaining the missing musical context that is normally only present in audio-based onset detection.

8.2.5 REFINING THE WAY MUSICIANS LEARN: MULTIMODAL PERFORMANCE METRICS AND MUSICAL PEDAGOGY

Lastly, bowed string performance was further explored using a hyperinstrument called the Ezither. In this instance, the goal of the work was to show the benefits of multimodal techniques in musical practice and pedagogy. The metrics and visualizations investigated ranged from various tempo measurements, to metrics concerning the player's performance of various bow stroke articulations. Specifically, the metrics included: accuracy of tempo performance; accuracy of tempo when playing with different articulations; and lastly, metrics concerning the physical properties of performance. Physical performance metrics contained the Onset Difference Time of difference articulations as well as the Articulation Attack Slope of the various bow strokes. The research investigated the performer's practice at various time-scales, from looking at overall observations of a particular performance, to analyzing the player's performance and progress over a seven-month period. This is the first work of its kind that not only shows the affordances of multimodality for metrics tracking and machine musicianship, but also takes the task outside of the research lab, and directly applies it to the everyday practice of a training musician.

8.3 Principles and Considerations on the Design of Multimodal Musical Instruments and Sensor Systems

This section forms general guidelines stemming from the experiences and lessons gleaned from working with a variety of musicians, across many musical contexts and styles, in the research presented in this work. In order to effectively capture the nuances of musical performance, extreme care must be taken while designing multimodal sensor systems for a given instrument or input device. It is possible that different multimodal instruments and sensor systems can share similar input modalities and core sensing technologies. Thus systems must be extremely sympathetic to the individualized needs of the given instrument and or performance context. As a result, the responsibilities and the possibilities afforded by a multimodal system can change merely by the musical context for which they are being used. For example, the physical association of a simple force-sensing resistor dramatically changes when being used to measure characteristics of a percussionist's playing, versus that of a North Indian sitar player. This section discusses a simple set of design principles for designing multimodal systems for musical performance.

8.3.1 WHAT IS THE MUSICAL CONTEXT?

The first questions that may be useful to ask when designing a multimodal system should aim to identify and understand the musical context for which there is a need or desire. Is the primary goal of the multimodal system for musicological, pedagogical, or performance scenarios? A combination? Once the musical ecology for which there is a need exists (and has been identified), it is possible to break down the instrument and its requisite performance attributes. In this way it is possible to identify how it is best to implement a sensor system. For example, in a controlled environment such as the research laboratory, it may be possible to use video camera systems to track a performer's physical action. Implementing the same modality for a touring project may not be appropriate. There may be too many environmental variables that can negatively affect the sensing system (ambient lighting, busy RFI channels, unknowns about the

performance space, setup times, calibration, etc.). In this scenario it would be useful to explore other modalities or sensing techniques. The key is striking a balance between need and context.

8.3.2 EXPLOITATION

Depending on the desired outputs within the defined musical context, an effective multimodal sensor system may include modifications to an instrument (or other interface) and/or the performer's body. This does not account for certain input channels (such as vision based sensing), which can potentially work in both areas at the same time. When modifying instruments, consideration should be made to exploit existing parameters of the instrument and its performance techniques. Doing so can provide the benefit of the additional modality without significantly increasing the learning curve or developing new techniques. Examples in this research include the thumb sensor on the Esitar, which captures the normal plucking activity from the performer, while turning the stream into control data and an analysis entry point (depending on the application).

Other examples include the accelerometers on the modified bow used in the experiments in Chapter 7 and performances in Appendix A. Embedding the sensor within the bow exploited the normal activity of the performer, while providing additional gestural dimensions to be explored on top of the traditional technique. This opened up a plethora of applications, both in the performing-research domains, as well as in the creative.

It is the belief of this research that when identifying a scenario which truly calls for multimodal techniques, that it is possible to implement them in a way which exploits the informational channels already present. This is true especially for multimodal hyperinstruments, but can also be applied to NIMEs, and other sensor systems. This is in accord with Cook's "Re-designing Principles for Computer Music Controllers", in which two of his design principles say that, "Existing instruments suggest new controllers" and "Copying an instrument is dumb, leveraging expert technique is smart" (P. Cook 2001; P. R. Cook 2009). In these situations the multimodal system should work to exploit and extend the

natural mechanisms provided by the performer, unless sound reason is provided otherwise.

8.3.3 TRANSPARENCY

One of the most important factors in multimodal musical interaction design is transparency. This is related to the ideas in exploitation, with a strong emphasis on the playability, complexity, and effects on established techniques. At all stages of design it is paramount to ask the following questions. How intrusive is the system you are designing for the performer? Does it have a negative effect on the instruments playability? Does the new system have a steep learning curve or has the learning curve of the modified instrument increased?

The physicalities of an instrument are crucial to its playability by the performer, down to the weight and diameter of a drummer's stick. The physical constraints the multimodal system places on the player and the instrument can dramatically influence the performance itself, the success, and usefulness of the system—therefore non-invasive solutions are highly desired.

8.3.4 APPLYING MULTIMODALITY TO A MUSICAL TASK VS. APPLYING MULTIMODALITY INTO A MUSICAL TASK

“Programmers who program “in” a language limit their thoughts to constructs that the language directly supports. If the language tools are primitive, the programmer’s thoughts will also be primitive.

Programmers who program “into” a language first decide what thoughts they want to express, and then they determine how to express those thoughts using the tools provided by their specific language.” —McConnell (McConnell 2004)

In the highly regarded book on computer programming, *Code Complete*, author Steve McConnell describes the difference between programming *in* a language, vs. programming *into* a language.

The idea of programming in a language vs. into a language as McConnell describes is an extremely powerful idea that can be applied to multimodal musical interaction, and the broader field of musical HCI. At the core of

multimodal interaction is the facility of multimodal integration, both as complimentary modalities and multimodal fusion. As such, approaching a musical scenario (whether research analysis or performance), one should first ask what the requirements of the task are. If the task is able to be satisfactorily addressed unimodally, it should be done as so, rather than forcing it unnecessarily into the multimodal domain. On the other hand, it is possible to unimodally implement a system that is more complex, in which case it may be useful to consider multimodal options.

As such, it is important to always ask, “am I applying multimodality *to the musical task*, or am I applying multimodality *into the musical task*?” If you are applying multimodality to a musical task, the benefits of a multimodal approach will be few, if not superficial. If you are applying multimodality into the musical task, the task will itself propose the need for a multimodal approach, and the affordances or outcomes will be greater. Multimodality is a means to an end, so it is always important to figure out the goal first, and to express the goal using a multimodal approach when necessary.

8.3.5 ON CONTINUOUS CONTROLS AND ‘LEAKY FAUCETS’

Working with multimodal systems often involves various continuous controls, and working with continuous parameters means taming them in a number of ways. The first and most obvious is what I like to call “plugging the leaky faucet.” While continuous controls can be extremely nuanced channels of data, there are simple considerations that are often overlooked when designing multimodal systems. A standard knob (potentiometer) for example normally facilitates continuous control only while the user is actuating the parameter. It is the equivalent of turning on a faucet, and increasing and decreasing the water flow by turning the handles. The user has control over the temperature and flow of the water coming out of the faucet. The faucet can be left open, or shut before finishing.

Traditional knobs however are usually consistent data streams. They can be set to a value, and stay there consistently. Continuous gesture controls such as accelerometers on the other hand can be leaky faucets. The valves are often

always open, and they continue to stream water (data) incessantly, and with fluctuation. For musical control, this can be extremely unruly (and admittedly, sometimes interesting). When working with accelerometers in this research, a number of simple methods were used to help deal with constantly streaming (and fluctuating) data. In this work, “plugging the leaky faucet” included, (1) filtering the data source (e.g. low-pass filtering) to smooth the data; and (2) bypassing the data stream with a toggle switch (button). Using another button to toggle between receiving and discarding is reliable, at the cost of requiring additional buttons. Another variation on this method is having a momentary button/switch that is active only when one wishes to engage (or disengage) the parameter. This requires additional coordination from the performer. Another useful method is (3) thresholding the input data. While this can slightly limit the active-range of the data, it can provide an effortless means of control where a sensor’s data stream is discarded when the sensor has not exceeded the given threshold.

8.4 Mapping Multimodal Musical Systems

To paraphrase (Hunt, Wanderley, and Paradis 2002), in traditional acoustic instruments, the sound source is inherently bound to the physical performance interface. A guitarist may pluck a string, which is both the actuator, and the sound source at the same time. Working with digital musical instruments (DMIs) and NIMEs presents a vastly different scenario, as the physical interface is normally separated from the actual sound source (the BoSSA (Trueman and Cook 2000) and Overtone Fiddle (Overholt 2011) are examples of exceptions). Thus, the relationships between the interface and sound parameters, or *mappings*, must be designed (Hunt, Wanderley, and Paradis 2002).

Mappings can be considered the connective tissues between a performer’s actions with a DMI/NIME, and the resulting musical output. They are a translating layer connecting a performer’s gesture to the sound. The openness in possible mapping relationships makes the applications of DMIs and NIMEs extremely flexible, but can also make the task of defining and learning the instrument, difficult or lack direction. Approaches to instrument mapping and

parameterizations have been proposed. Hunt and Wanderley offer principles for mappings in gestural music in (Hunt and Wanderley 2002); these include the mapping parameterizations into four categories: as *one-to-one*, *one-to-many* (divergent), *many-to-one* (convergent), and *many-to-many*. Wessel and Wright propose specific metaphors for guiding computer-based musical interactions in (Wessel and Wright 2002). These metaphors include *drag and drop*, *scrubbing* (and variants), and *dipping*. While the above examples propose mapping techniques in specific instrument and gestural examples, Tanaka discusses higher level mapping strategies to enable the articulation of musical phrases (beyond specific instrument cases and designs) in (Tanaka 2010).

The scope of this section is not to provide an overview of all recent mapping principles and models. For a review on the current literature on mapping and computer music, please refer to (Hunt and Wanderley 2002), (Miranda and Wanderley 2006), and the previously cited sources. Rather, this section will highlight particular musical mapping principles. Many build off previous work as mentioned; however, here their definitions are extended to the context of multimodal musical interaction.

8.4.1 ONE-TO-ONE, COMPLIMENTARY MODALITIES, AND MULTI-DIMENSIONAL CONTROL

Hunt and Wanderley discuss one-to-one mappings as mappings “where one synthesis parameter is driven by one performance parameter” (Hunt and Wanderley 2002). This dissertation has iterated that one of the primary goals of multimodal interaction is to enable new sonic situations through complimentary modalities. The idea that harnessing the affordances of two disparate modalities can lead to a more meaningful experience—a synergy between the independent modalities and modes. In working with one-to-one mappings in live performance in this research, the principle of one-to-one mappings has been found to lead to interesting performance affordances when under the guise of complimentary modalities. In the traditional definition, one-to-one mappings are independent channels. For example, a knob may be tied to the cut-off frequency of a low-pass filter on a synthesizer, and a slider to the attack of the synthesized sound. Both

parameters alter the output sound, but interaction wise, are independent acting agents. Under the definition of complimentary modalities, the sensing inputs are technically still independent agents, however, their interactions serve to complement one another. Complimentary modalities aim to encourage one-to-one mappings that are both physically and psychologically more enticing. As an example, the Turbine application described in A.4.4 utilizes one-to-one mappings of the SmartFiducial’s multimodal sensors, to control various parameters of a wavetable synthesizer. Complimentarily, the individual sensing modalities afford a new multi-dimensional level of gestural control. The user is able to move and rotate the fiducial object (x, y, and rotation dimensions), while simultaneously gesturing above the object with their free hand. Using the vision tracking alone enables x, y (position), and rotation parameterizations, while a fourth parameter is enabled through the distance sensor embedded within the face of the fiducial object. The particular gesture enabled by the complimentary modalities, however, creates a new experience altogether—one that results in a more physically and psychologically nuanced interaction.

8.4.2 MANY-TO-ONE AS A SPACE FOR MULTIMODAL INTEGRATION

Hunt and Wanderley also discuss many-to-one mappings, where multiple performance parameters control just one sound parameter. Many-to-one mappings present an interesting space where multiple modalities can be democratized to facilitate a particular parameter, or gesture. Multimodal fusion is characteristically many-to-one by definition—data sensed from multiple modalities combine to form one final [musical] output. Tanaka discusses these “compound mappings” as effective means of articulating music phrases or single events, under his mapping model proposed in (Tanaka 2010). The idea of multimodal integration for many-to-one mappings in live performance is extremely powerful. In section 8.3.5, simply mapping a button to function as a momentary or toggle gate was discussed as an effective method of regulating unruly continuous parameters. This sort of many-to-one mapping, when combining information from multiple modalities, is a powerful method for manipulating and generating musical events.

8.4.3 DEFINING A SET OF PARAMETERIZATIONS

As discussed in the performance reports in Appendix A, the definition of a finite set of parameterizations is key to successfully using multimodal instruments and sensor systems. With acoustic instruments, musicians spend years learning the fundamental techniques over a finite set of musical parameters. Even when considering extended techniques, the amount of possible parameterizations of DMIs and NIMEs far outweighs those possible with traditional acoustic instruments. The space then only becomes more dense in the case of multimodal musical instruments, as the cross-modal interactions add additional complexity to control parameters. Thus, it is important to revisit the original principles outlined in 8.3 to define a finite set of parameterizations for the piece or instrument. It is then, that non pre-determined parameterizations can arise, and be added, and non-useful mappings discarded.

8.5 Future Work

There are many areas that the work presented in this research can continue to develop. Nuance (or a sister-application) for example could develop as a system that not only captures multimodal data from musical systems, but also provides analysis and visualizations in one package. It would be useful to deploy a system like this within an active music curriculum, and to evaluate on a large scale, how training musicians and educators benefit from multimodal musical practice.

This work has shown some of the unique affordances in machine musicianship and machine learning scenarios. It can be argued that the moment the first intentional musical sound was ever produced, so was the need for multimodal integration. The nature of producing sound is intrinsically multimodal, in the sense that on a granular level, sound is a combination of physical actions, physics, and acoustics. To that end, there exist fervent possibilities for applied multimodal techniques for machine musicianship and machine learning tasks. These fields already enable many possibilities in both musical analysis and performance, and multimodal techniques can provide a unique vantage, which has been largely unexplored. Building on the particular

research presented in this work, one area to explore is in rudiment recognition. This research has already presented a framework for drum-hand recognition, and extending this work into pattern (rudiment) recognition could be useful for practicing drummers, and for score-level control in live performance (e.g. automatically enabling sound processes based on particular patterns played).

8.6 Conclusion

Over the past decade, the movement towards expanding established modes of musical interaction using HCI has been guided into many directions. In this research, we have shown the importance of providing a multimodal vocabulary to musical HCI. Using multimodality as a powerful and promising light, the future of nuanced musical HCI can continue to evolve. This research has shown that multimodal systems can enable many unique musical interactions when given the opportunity. To this end, we have proposed software, approaches to instruments and sensor systems, and other techniques that make it possible to tap into multimodal musical HCI.

Establishing multimodal communication channels, this research has shown that novel affordances in musical HCI are within grasp, and are very promising. The breadth of possibilities is far reaching, and as this work has demonstrated, can empower many new sonic engagements in creative and performance contexts, as well as in practice and pedagogical scenarios. Our investigations into multimodality have been inspired by, and applied to active areas of musical research, namely machine musicianship and machine learning. Exploiting the physical and acoustical dimensionalities of multimodality in these areas, this work hopes to inspire others in the field to begin asking, how else can multimodality help achieve our musical tasks and desires? Throughout time, music has often taken twisted and turning paths in response to individual and cultural needs. Like Russolo and many others before him, now more than ever, there is a widespread movement to negotiate new musical interactions, through the development of new instruments and techniques. In this work we have shown that leveraging the affordances of multimodal techniques can make this possible. As such this work asks, how else can multimodal techniques facilitate your musical needs?

Section IV

Appendix

Appendix A

Live Performances and Applications

Additional Explorations in multimodal live performance and HCI

Chapter 3 through Chapter 7 have shown how multimodal techniques can be used in the laboratory or practice room for analysis. This appendix serves to show how some of these techniques can also be used for modern artistic endeavors. Selected performances and projects in which multimodal techniques were used to shape live musical performances and multimedia interaction are presented. These include musical performances in A.1 and A.2, and a multimedia performance in A.3. Lastly, a case study on applying multimodal techniques to alternative musical interfaces (tabletop surfaces) is explored in A.4.

A.1 Minim Performance at the New Zealand School of Music Sonic Arts Exhibition Concert, October 9th, 2010

On October 9th 2010, Owen Vallis and I performed an untitled piece from one of our collaborative music projects called *Minim* in the NZSM Sonic Arts Exhibition. In this section I will document a few elements from both the compositional and performance experiences, which relate specifically to the themes and research presented in this dissertation.

Initially we approached the piece thinking about previous *Minim* compositions, which were primarily long structure or slowly evolving ambient works. A large interest in previous works explored somewhat microevolutions of synthetic sounds and timbres. These could be as simple as the slow and controlled pendulum swing of synthesized parameters; paying close attention to the resulting perceived periodicities and movements produced in the sound.

Brainstorming and composing our piece for this concert however, we chose to explore a few other key ideas that influenced my conceptions of music performance systems and sensors. Ultimately this concert very directly impacted my goals and necessities in future performances at the NZSM (detailed in A.2).

The first clear differentiation from other *Minim* pieces was that we decided not to compose for any synthesized sounds or generators. The piece was written specifically for piano and guitar (acoustic), and explored various themes in controlled feedback, excitation, and impulse. This was divergent from previous pieces that cycled around long, evolving timbre spaces.

A.1.1 Excitation, Impulse, and Probability Machines

Using acoustic instrumentation, we began thinking about a number of various interaction ideas and principles. How could we facilitate multiple modes of user or autonomous interaction? How could different sound processes embellish, or expose, various properties of the acoustic sounds. Specifically, we were intrigued with the idea of controlled feedback, inputting short musical events into the machine, and eliciting a response; essentially creating a situation in which the machine feedbacks musical events by exposing and manipulating parameters of the original input such as harmonics. I remember Owen bringing up a piece by David Tudor in which delays could stabilize feedback systems and reinforce overtones.

We ultimately wrote two software plugins for the piece, one a granular signal processor, and the other a probabilistic re-sequencing sampler. Owen wrote the granular effect in *Reaktor*, which acted as a harmonizing feedback machine. The input could initiate an impulse by triggering a reset button, which would pitch and overlap the grains with various probabilities and in relation to the input. The effect would expose various partials and harmonies of the input signal, in an overlapping sequence of pitched grains. The output would be resampled when retriggering, resulting in a controlled feedback-based granular system.

The second plugin was a probabilistic sampler we co-wrote in C++ called *Audio Carwash*. Upon triggering the impulse button, the incoming signal was sampled for the length of one bar. The sampled bar's material was then

automatically divided into k divisions (steps) and pseudo-randomly re-sequenced. Each step in the sequence was assigned two parameters by the user; one parameter was the probability or likeliness of playing (on/off), and the other parameter was the amplitude or loudness of the step (if played).

We decided not to apply a smoothing ramp to the start and end of the steps, leaving a desired “clickiness” and attack to each step when playing back in the sliced order. The resulting effect was a highly rhythmical chopped and re-sequenced one-bar phrase, and offered many levels of musical interaction. As an example, the performer and effect could initiate a game of cat and mouse, where the phrase would trail the performer’s input by one-to- n bars, by retriggering the sampling every n bars. The probability based sequencing was also free running; providing an autonomous degree of variation and dynamics to the piece. This in turn freed the performer to continue working on other aspects and interaction levels in the piece.

A.1.2 Composing by Improvisation

Before and during the compositional process we discussed and re-evaluated the macro structure of the piece. We established a form for the piece’s arc and development, but left the body of composition to structured improvisation. At first I found it useful to improvise with one another using our new effects, without worrying about form. I learned subtle ways to control and interact with the probabilistic nature of Audio Carwash. Eventually we saved particular probability sets (presets) we liked so that we could go through the piece and effect the input in an organized, yet still probabilistic manner. This facilitated a particular level of control for the performer, where one could influence the note density in the produced sound by cycling through a pre-defined sequence of probability sets. Still the element of uncertainty, and controlled randomness, gave the rhythmic portions of the piece a fresh, cyclical evolution that was exciting for me.

Ultimately the task of composing the piece (initially) improvisatorially worked quite well in this case. We were working with instruments we’ve played for many years, and gave ourselves a limited set of additional controls.

A.1.3 Performer Interaction

For the performance we built simple foot control pedals with two buttons to control the plugins. One button reset the grain plugin, and the other triggered Audio Carwash to resample the input for one whole bar. We both played our instruments (Owen on piano, myself on guitar) mostly traditionally, with a few parts requiring other techniques, for example picking behind the bridge or at the headstock of the guitar.

As mentioned, working with probability-based effects was an interesting space to work in. It enabled the ability to set something in motion, with a refined state, and to simultaneously move into other areas or direct mappings. The uncertainty in the grain plugin, how it would sometimes latch onto the input and suddenly emerge harmoniously was particularly exciting. As the performer, I never knew exactly how the note would feedback and sound when I retriggered the grain plugin, although I could influence and initiate the event with a comfortable amount of confidence and control. The uncertainty however was electrifying, ultimately allowing me to focus and listen, and very much give over to the piece.

At the same time, giving over a lot of direct, continuous physical control left me desiring more in terms of interaction. Often I would pluck a single note as an impulse to the grain plugin, hear the response, but have no means of engaging with the sound further, other than re-inputting into the grain or Audio Carwash plugins. As a first performance I was ultimately happy with the general outcome, however, I desired the ability to interact with various parameters and manipulations with greater control. This was a primary concern I aimed to address in my next performance at the NZSM, which is detailed in A.2.

A.2 III: Performance at the New Zealand School of Music Sonic Arts Exhibition Concert, October 9th, 2011

III was composed and performed collaboratively between Jason Erskine, Blake Johnston, and myself. We got together with the goal of composing and

performing on hyperinstruments, investigating some of the possibilities of hybrid acoustic instruments with gestural controls. While I had played in groups in the past that incorporated hyperinstruments and other hybrid instruments, this was my first experience performing on a hybrid acoustic-electronic instrument myself, which led to a number of discoveries shared in this section.

A.2.1 Hyperinstruments and Gesture

PREVIOUS EXPERIENCE AND INSPIRATION

In the previous years Sonic Arts Exhibition in which I performed with Owen Vallis, I had consciously constrained the amount of real-time parametric control I was able to input into the performance system. There was a certain level of user input and feedback as discussed, however my own personal constraints led to binary or impulse (trigger) input actions. The reasons for limiting the amount of control were two-fold. A major factor were physical limitations of picking and fretting a guitar. Although perhaps possible with extended practice, it was imprecise (physically) to control continuous parameters with my feet. The foot pedals we built to trigger our effects were large, containing only two buttons; this helped overcome the low precision of our feet for parametric control (similarly, foot pedals like wah-wahs use a large footprint with 1-D control). The other reason for limiting the parametric control was conceptual. The piece itself was concerned with setting up a situation of auditory feedback and probabilistic automata, and thus, impacted the kinds of performer interactions in the performance.

When approaching the piece this time, I was particularly inspired by Curtis Bahn's EDilruba (Sensor Esraj) implementation from a (at the time) recent performance in our group, *the KarmetiK Machine Orchestra* (Ajay Kapur et al. 2011). The EDilruba is a traditional dilruba (esraj), a Hindustani bowed instrument found in North Indian classical music. The frets are played in a *siding* style, achieving portamento or *meend*, which is a common trait of Indian music. Bahn's EDilruba uses a simple biaxial accelerometer at the frog of the bow, providing two axis of continuous parametric control of the sound. While there are other

examples of bow-controllers and performances, it was performing with Bahn earlier that year which really caught my attention as something I not only was interested in researching, but in using and performing with myself.

With two axes of continuous control, Bahn was able to negotiate many different sonic engagements with performer intent. Often he would play the instrument traditionally, followed by an in-air gesture of the bow, after lifting the bow off the strings at the end of a note. In this way he was able to continuously modulate the performed note with the gesture of the bow (as the note continued to decay naturally). The motion was both seamless and elegant, and by switching “modes” with a few buttons in front of him, he was able to exert a high level of parametric variety and control over the processing of his sound—without hindering his ability to play regularly.



Figure 64: Curtis Bahn and the EDilruba (Sensor Esraj)

Other times he would use the bow independently of the acoustic instrument itself. For example, during a noise interlude, he would set off a free running noise process, controlling multiple parameters of the sounds evolution by subtly gesturing the bow in the air. The bow could be used and appropriated in many ways, from affecting the acoustic sound of the instrument itself, to becoming a musical wand in purely synthetic circumstances. There are many other elements of this that I felt worked very well, from the immediacy of the correlation between input and output for the audience, to the subtle and dynamic levels of control. Thinking about these components, I began to wonder how I could apply

some of these principles to guitar, freeing myself to explore similar ideas as the piece I performed the year prior with Owen, with the added level of interaction I desired.

FORMING THE “GROUP”

For almost a year I had been working independently with two students at the New Zealand School of Music on designing and building their own custom hyperinstruments. Jason Erskine and I had been working on refining his Esuling (see 3.1.3), a modified Balinese suling (ring flute), and Blake Johnston on his custom instrument called the Ezither (see 3.1.2). We quickly realized we shared a lot of similar musical interests, and that we had unintentionally created a situation in which we could collaboratively explore our modified instruments together.

A.2.2 Composition, Improvisation, and Iteration

PREPARATION IS KEY

Our compositional process was collaborative; we would get together over the course of a few weeks and play out the sounds we were currently getting out of our instruments. It was an iterative process that involved preparing signal chains and effect processes at home, and then playing in the context of the group. During rehearsal we would stop, assess the role and functions of the sounds we produced, parameterizations, mappings, and performative roles within the group. We would often stop to adjust our interfaces as necessary, adding particular functionalities as required.

WORKFLOW AND CHALLENGES

Our two initial rehearsals exposed some of the challenges in workflow when working with newly designed hyperinstruments. We had yet to develop systems to easily allow us to map, and remap our controls to various parameters, as rehearsal sparked ideas. Jason had been performing for quite some time with an

earlier version of his Esuling, and so he was most prepared to switch between interaction modes and mapping of his instrument. After the second rehearsal, Blake and I both decided to come up with a framework to allow us to map our controllers more easily. This resulted in modifying our software that translated the sensor data into MIDI and OSC messages, allowing us to map parameters with the click of a button in the programs user interface. It was also useful to come up with two or three ideas and ways to use our instruments at home, and then to show those ideas to the group during rehearsal. From there we pieced elements together, and started forming the structure of the composition. One of the most important things that made rehearsals productive was preparation and streamlining of the possible interactions, and defining a set of practices that we could easily try out at any moment.

We did this until we were satisfied with our parameterizations, and could focus solely on performing the piece. Working with hyperinstruments, there were many levels of control, mappings, and other interactions possible. Ultimately, we knew that we could live within a state of defining (and redefining) our interactions forever, and decided to set a composition deadline for ourselves. As such, after the first couple of weeks we made an effort to solidify our technological structure and instrument mappings, so that we could focus wholly on the performative aspects of the piece (both the music and learning our newly mapped instruments). Of course the two (system design and performance) were not mutually exclusive, and influenced each other greatly during this process. Mappings and parameterizing came directly out of improvising and composing, and vice versa.

The composition itself was set into place, defining a macro-structure with space to improvise within if desired. We memorized the structure of the piece during rehearsals, which greatly afforded us the freedom to explore the music and the instruments when performing.



Figure 65: Performance of *III*, group (left), and close-up of modified 12-string acoustic guitar and bow (right)

A.2.3 Performer-Interaction

The main modes of interaction on all three instruments involved gestural control from triple-axis accelerometers. Jason's Esuling included an accelerometer at the top of the instrument, and Blake and I both performed using modified bows (with accelerometers at the frog). Additionally, my accelerometer system described in 3.1.4 provided a secondary accelerometer that I attached to the body of the guitar. In this way I could also tilt and twist the guitar (played vertically), controlling six dimensions of parameters simultaneously. Excluding myself, additional control was facilitated by the use of knobs, pressure sensors, and buttons on the performers' respective instruments. I however, controlled a MIDI device with my feet. Again because of the limited precision, I only mapped a few parameters (volume slider, and a few buttons), and placed them far apart on the controller to require less precise physical action.

A number of considerations emerged from interacting primarily with accelerometers, related to those discussed in 8.3.5. While some of these may seem obvious, they are useful guidelines when preparing, and arose through actual interaction. In order to effectively parameterize the accelerometers, and repurpose the data stream in real time, the following methodologies were practiced.

Firstly, buttons were assigned on the foot controller to toggle (bypass) each of the effects I was using. In this way I could easily switch between which effects were currently being controlled by the bow gestures. This was a simple yet effective way to repurpose the bow's parameters in real-time, essentially giving the bow banks or modes in which I could switch between at any given moment.

Parameters were also often considered and practiced in pairs. During rehearsal I experimented with pairing different parameters on the x-axis and y-axis, and examining the intuitiveness of the mapping. Some pairs worked extremely well, while others did not. For example, one mapping-pair that worked well for this piece was mapping the x-axis as a wet/dry control, and the y-axis as an effect parameter. In one use case, this was achieved by creating a “return” channel with a delay effect and a hi-pass filter. The x-axis of the accelerometer was mapped to the return “send amount” on the guitar input track, while the y-axis was mapped to the filter cut-off frequency. As I played I could slowly rotate the bow while bowing to send more or less to the return track—effectively adding in more or less delay. Tilting the bow forward or back, I also controlled the spectral shape of the delay, which gave me a high level of simultaneous control of the sculpted sound. I had previously experimented mapping those parameters individually, and paired with other parameters, which didn’t quite feel as natural. However, when I mapped them in this particular pair, the inter-relationship between side-to-side roll, and front-to-back tilt of the bow opened up a more physically and musically intuitive path for me to engage.

There were also times where our instruments were excited with non-traditional actions, and further manipulated by our gestures. For the intro of the piece, we used the resonating qualities of our instruments, without their actual excitation mechanisms. By tapping the instruments, we initiated impulses that were amplified by the microphones on the instruments. We then used the gestural sensors to sculpt the knocks and taps into rhythmic material. This foreshadowed a section of the piece later where the roles of impulse and interaction were reversed. Later in the piece I sampled and re-arranged one bar of a bowed line, using the Audio Carwash plugin described previously in A.1. The recorded phrase was post-effect processing, and so it included the bowed note (impulse), after being processed by the bow gestures and effects. In turn, the probability-based re-sequencing now affected the input. Whereas before I would input a knock, and affect it with the gesture (using the bow), in this case I would input the note post-bow processing, and the machine would probabilistically affect the gesture.

A.3 Transformations: Integrating Multimodal Music, Dance, Visuals, and Wearable Technology, May 11 – 17, 2012

In October of 2011, a friend, Leila Navon, approached me about an installation and performance project of hers titled *Transformations*. I helped advise the project with both technical and artistic direction, realizing her performance in three shows taking place on May 11th, 2012 and May 17th, 2012 at California Institute of the Arts.

Upon discussing her ideas, she wanted to develop an integrated multimedia piece that incorporated live computer music, dance, and visual projections. Ultimately she envisioned the dancer interacting with both the auditory and visual mediums, with the ability to guide the movements and interactions between modalities. This project was particularly interesting to me, as it was highly interdisciplinary and would enable the exploration of multimodal techniques not just in music, but in larger multimedia and art-based contexts.

A.3.1 Designing the System

I advised Leila in determining a list of interaction requirements in order to assess the technological needs of the project. After lengthy discussions the primary interaction requirements included:

- Continuous control from the dancer to:
 - Manipulate the audio in real-time
 - Facilitate real-time interaction between the dancer and visualizations
- Audio/Visual interaction from Leila herself

Initially we planned to use a Microsoft Kinect²³ camera for all dancer-centered interaction. The Kinect is a motion-sensing camera designed by Microsoft that enables hands-free user interaction. Essentially the Kinect

²³ <http://www.xbox.com/en-US/kinect>

employs a 3D light scanning system that enables motion and gesture tracking in three dimensions (x, y, and depth). Very shortly after the Kinect was introduced to the public in 2010, the open-source online community developed drivers and entry points into the motion capturing data for most programming platforms.

CHALLENGES WITH THE MICROSOFT KINECT

The Microsoft Kinect was a great solution in that it was readily available and affordable, and facilitated many of the interactions desired. However upon hands-on experimentation, a number of challenges arose that were eventually addressed with other modalities. The first limitation was an issue of frame rate. Working with the dancer, often the choreography demanded movements from the dancer that surpassed the camera's frame rate of 30 Hz (frames-per-second). This led to the undesirable quantizing or sluggish response of audio/visual parameters.

While filtering, smoothing, and interpolation were possible solutions to help minimize the issue for musical parameters, other considerations arose. One such consideration was the camera's limited field of view in the open plan performance space designated for the piece. We discussed focusing the camera on a particular zone of the room, and having all interactions take place in that zone; however, Leila expressed strong convictions in having the dancer interact with the musical parameters at any given moment or location within the space.

At this point it was clear another solution might be necessary for musical-dancer interactions. However, an inspiration for the piece was a video Leila saw online in which projections were mapped (masked) onto the body of a dancer in real-time. As such the Kinect would still play a key role in the piece.

Working with the dancer and a real-time image masking solution called the *Mad_KinectMasker*²⁴, the camera still exhibited less than ideal frame rates, and a limited active area in which the dancer could be reliably tracked. Other tracking frameworks exist, which exhibit a wider active tracking area, however the project

²⁴ *Mad_KinectMasker* renders an image mask of a persons shape in real-time which can be used to mask the projection output, available at <http://www.madmapper.com/madlab/>

time frame did not permit the design of a custom image masking solution utilizing these frameworks.

As such the two major limitations were carefully addressed in the implementation of the piece in the following ways. Firstly, the piece began without any visible projections. The dancer began on the ground, and slowly worked through the introduction choreography.

At a certain point in the piece, the dancer moved within the active tracking region of the camera. The projections then automatically appeared and masked to the body of the dancer. The music and movements of the dancer at this point were composed and choreographed more slowly, to help minimize the effects of a limited frame rate. The effects of frame-rate were most noticeable with the projections, and moving too fast would result in the projections lagging behind, essentially losing the mask and mapping onto the wall behind the performer. With practice and slower musical tempo selection, we were able to achieve satisfactory results with the dancer.

To deal with the limited tracking area, the dancer's interaction with the Kinect happened within a limited footprint in the space. All choreography at this point happened low to the ground, enabling a tightly controlled space to be explored. In addition to mapping the projection onto the dancer, the dancer's contour was tracked, and projected on the wall behind her. In a sense, the wall (projection space) behind the dancer became an interactive canvas. Leila improvised the projections, controlling the particular images and image processing in real time. Additionally, the depth sensing from the Kinect was used to manipulate the wall projection, making the dancer's shadow on the wall appear larger or smaller depending on her proximity to the camera. With some of the projections on the wall, the dancer's contour and movements were shadowed, while other times the wall acted as an interactive surface, responding to the choreography.

UTILIZING THE XXL ACCELEROMETER SYSTEM

In my other research and performances, I had utilized a general-purpose gesture sensing system I designed called XXL (see 3.1.4). XXL is a simple wireless

accelerometer based sensing device that I proposed could be combined with the Kinect tracking. While the accelerometer could not replace the Kinect in that it could not meet the real-time tracking requirements of the project, it could support the Kinect interactions in a number of ways as a complimentary modality and sensor.

Firstly the accelerometer system can achieve quite long communication range (depending on the chip used, it can communicate up to a few kilometers or miles in an open-field). This overcame the Kinect's limitation of only being able to sense in a limited area of the performance space. Secondly, the system was designed for musical control, achieving frame rates up to 100Hz and a resolution of 10-bits.

As such I helped Leila build her own version of the XXL system for the piece, using the original designs. The two accelerometers were placed on the hands of the performer (see Figure 66) and were used throughout the piece to control various musical and visual parameters. Leila controlled the mapping of the accelerometers to musical parameters in real-time, allowing the dancer to focus on their movements, while paying acute attention to the musical elements they were controlling. The circular dependency between the dancer and audio/visual environment led to very high synchronization between elements in the composition, visuals, and choreography.

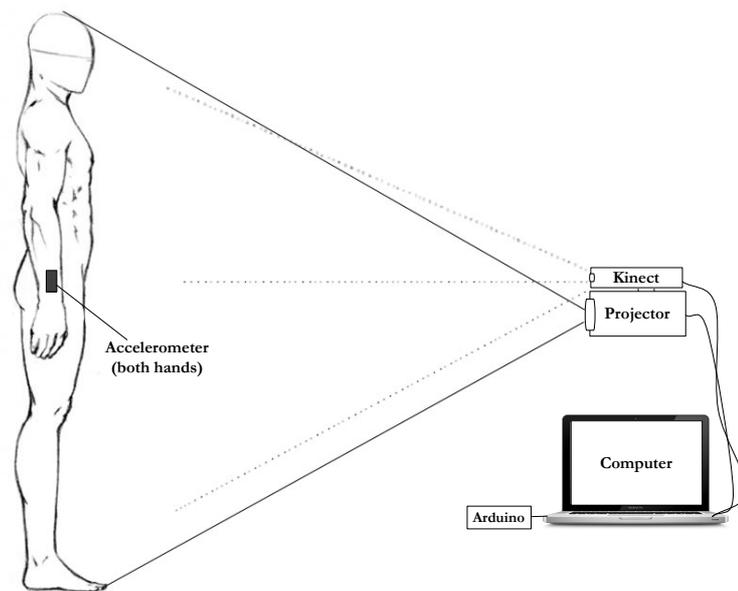


Figure 66: Overview of dance technology, XXL accelerometers on hands, and Microsoft Kinect for real-time projection mapping (masking) onto the dancer

A.3.2 Discussion

Overseeing this project was a beneficial experience in which we were able to work through many challenges by using complimentary modalities. Utilizing both vision tracking and accelerometer control, we mapped projections onto a dancer in real-time, enabled interaction with an interactive surface on the wall behind the dancer, and enabled precise control over various musical and visual parameters of the piece with the XXL system. It was a valuable exercise in working within the constraints of the particular modalities, while satisfactorily meeting the initial artistic endeavors. For the Microsoft Kinect the limitations included limited frame rate and field of view. These were addressed by the high precision and speed of the accelerometer system. Subsequently the accelerometer system failed to meet the projection mapping requirements, which was achieved by the Kinect. Utilizing both modalities, a synergy was found and the artistic requirements and desires for the piece were faithfully met.

It was also particular interesting to observe both performers interacting with a shared social instrument. As the dancer moved and improvised with the musical parameters, Leila adjusted the dancer's sensor mappings in real-time. There was a great deal of exploration and listening, and it was a particularly

interesting experience to see both of them sharing a changing multimedia environment. While there was a structured form to the piece, as well as choreographed movements, the mappings possible (between the dancer, the technology, and the audio-visual output) were rehearsed, but improvisatorially arranged each performance. Ultimately, this led to many beautiful realizations and interpretations of the piece.

A.4 SmartFiducial

Affording new interactive musical experiences using “tangible surfaces”

While the previous sections have focused on particular applications of multimodality in live performance contexts, early on I was also particularly interested in extending new performance interface capabilities using multimodal approaches. Multimodal performance metrics tracking and machine learning naturally lend themselves to using hyperinstruments and sensor augmented musical instruments; effectively capturing the performance characteristics of players on their traditional instruments. However, interfaces described in 2.2.1 as diverging from traditional performance paradigms, can also benefit greatly from multimodal techniques. In this section, we present the SmartFiducial, a wireless tangible object that facilitates additional modes of expressivity for vision-based tabletop surfaces. Using infrared proximity sensing and resistive based force-sensors, the SmartFiducial affords users unique, and highly gestural inputs. Furthermore, the SmartFiducial incorporates additional customizable pushbutton switches. Using XBee radio frequency (RF) wireless transmission, the SmartFiducial establishes bipolar communication with a host computer. This section describes the design and implementation of the SmartFiducial, as well as an exploratory use in a musical context.

A.4.1 Background and Motivation

Musicians have long been intrigued by gestural interfaces since the invention of the Theremin in the early 20th century (Glinsky 2000). This has led to the exploration of pressure-based input sensing for expressive musical interaction. Realizing the potential expressivity of gestural interaction in musical contexts, researchers have developed a number of hands-free and pressure based interfaces, exploring several sensing technologies. These include laser controllers such as Hasan, Yu, and Paradiso’s work on the Termenova (Hasan, Yu, and Paradiso 2002), Wiley’s Multi-Laser Gestural Interface (Wiley and Kapur 2009), Murphy’s force-sensing resistor based controller the Helio (Murphy, Kapur, and Burgin 2010), and countless others.

Concurrently, the last few years has seen an explosion of interest in musical tangible interaction including the Reactable (Jordà et al. 2005), the Bricktable (Hochenbaum and Vallis 2009; Hochenbaum et al. 2009; Hochenbaum et al. 2010), Block Jam (Newton-Dunn, Nakano, and Gibson 2003), the Audiopad (Patten, Recht, and Ishii 2002), and other audio/visual interactions (Ferguson, Beilharz, and Calò 2012). The Microsoft Secondlight project (Izadi et al. 2008) is an interesting example of adding additional input freedom to tabletop surfaces by quickly alternating projection between two independent diffuse surfaces (the tabletop and ones above the tabletop). While Secondlight can track tangibles and gesture above the surface, it lacks distance tracking. Tangible surfaces can undoubtedly provide users with extremely dynamic interaction, however they lack the gestural qualities of non-contact and pressure based interfaces.

The SmartFiducial is an attempt to provide the best of both worlds—offering and expanding upon the traditional x, y, and rotational modes of interaction available on tabletop surfaces, while providing the gestural expressivity and sensory affordances experienced from hands free and pressure based interaction.

The remainder of this section is organized as follows: first the physical design and technology embedded within the SmartFiducial is described, followed by a discussion of the use of the SmartFiducial with an interactive musical application.

Lastly, a discussion is provided detailing the design considerations and affordances of the SmartFiducial.

A.4.2 Implementation

The SmartFiducial offers users multiple degrees of freedom and expressivity. In this section, we describe the hardware design of the SmartFiducial that enable these input freedoms, as well as our exploratory software implementation of using SmartFiducials in a musical setting. Figure 67 provides an overview of the SmartFiducial tracking system, which is further expounded upon in the following section.

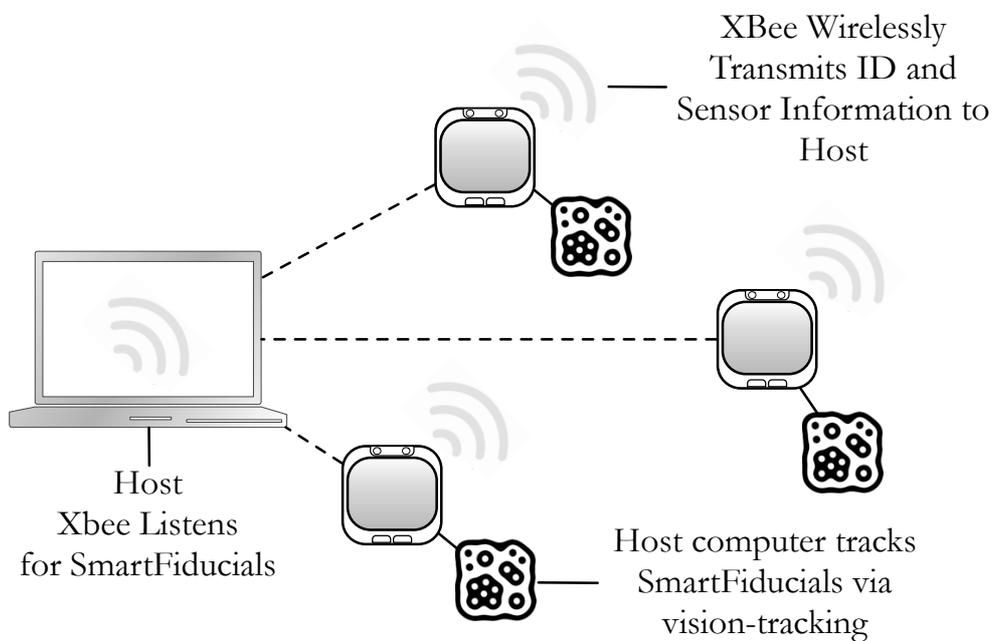


Figure 67: SmartFiducial System Overview Diagram

A.4.3 Hardware

VISION TRACKING

X, Y, and Rotation tracking is achieved using a custom version of the open-source vision tracking software CCV (Community Core Vision). CCV

implements the libfidtrack engine developed for the reacTIVision system (Kaltenbrunner and Bencina 2007).

Z-DEPTH

In addition to x, y, and rotation information captured by the vision tracking system, the SmartFiducial enhances the traditional 2D optical tracking system into a three-dimensional space. Z-depth input freedom is achieved by a short-range Sharp GP2D120XJ00F infrared (IR) proximity sensor embedded on the top face of the SmartFiducial (Figure 68, item C). The *GP2D120XJ00F* has an active sensing range of approximately 3cm – 40cm, providing users with a coverage area capable of highly expressive gesture sensing.

PRESSURE SENSITIVITY

The SmartFiducial also provides pressure-based gestural input via two force-sensing resistors (FSRs), on the sides of the SmartFiducial, as depicted in Figure 68, item B.

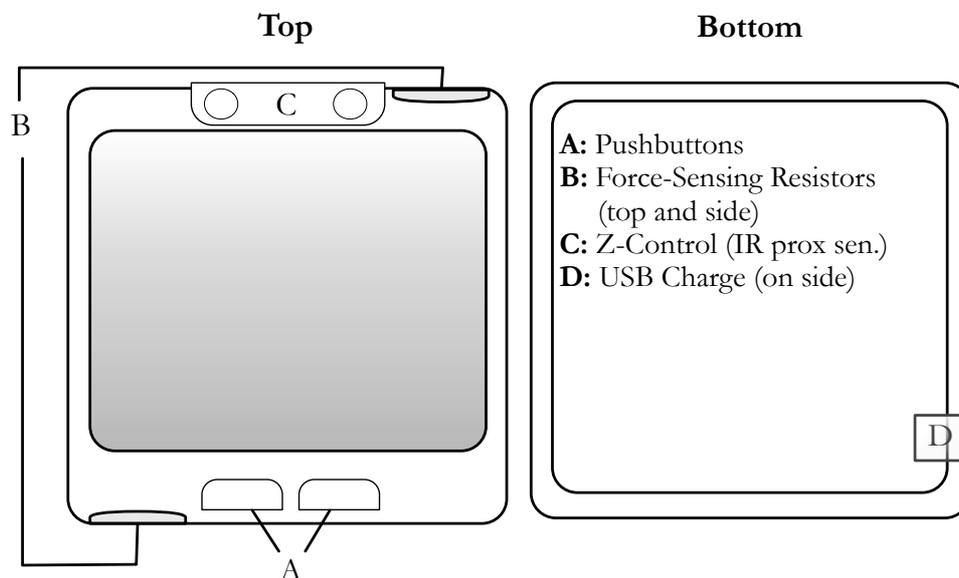


Figure 68: SmartFiducial hardware design and layout

WIRELESS TRANSMISSION

Embedded within the SmartFiducial is an Arduino Funnel IO²⁵ (Fio) equipped with an XBee wireless transmission module. XBee utilizes the ZigBee communication protocol, operates at 2.4GHz radio frequency, and exhibits extremely low power-consumption properties. This makes XBee an excellent candidate for wireless serial communication between the SmartFiducial and a host computer. Additionally, the XBee provides each SmartFiducial with a unique identifier, tied to its fiducial ID.

Data is received wirelessly via an XBee connected to the host machine, and is parsed by the custom version of CCV. CCV then sends out the ID and sensor data to other client applications using a custom implementation of the TUIO²⁶ protocol (Kaltenbrunner et al. 2005) that supports the additional data (Figure 69).

Additionally, a basic algorithm has been implemented in the SmartFiducial firmware to only broadcast new data when input is detected. This optimization helps to reduce the amount of data being transferred in larger system use-cases and scenarios, and can optionally be turned off in the firmware if constant streaming is preferred.

Once CCV receives new data bundles from the connected XBee, it first checks to make sure that the SmartFiducial's ID is present in the list of active fiducials being tracked by the vision system, before broadcasting a new TUIO message. This prevents the SmartFiducial's sensor data from being transmitted when not active on the tabletop surface, however, this can optionally be turned off if off-surface interaction is desired. Although the SmartFiducial messages include all information present in standard TUIO fiducial (“/2Dobj”) messages, CCV also broadcasts the SmartFiducial as part of its regular fiducial message broadcasting. Lastly, support for the SmartFiducial has been added into the standard C++ TUIO client implementation allowing easy integration into custom software applications. Support for the SmartFiducial in other TUIO client implementations (Java, Processing, openFrameworks, Max/MSP, Pure

²⁵ The Arduino Funnel IO is an Atmega based microprocessor designed by Shigeru Kobayash

²⁶ TUIO is a UDP based data-communication protocol, built around Open Sound Control (OSC)

Data, etc.) is planned for the future; however, the SmartFiducial data can still be accessed cross-platform via any OSC receiver application or library.

/tuio/smartFid set sId x y z a X Y A m r f F b B		
sId	Session ID	int32
id	Fiducial ID	int32
x, y, z	Position	float32
a	Angle	float32
X, Y	Velocity Vector (motion speed & direction)	float32
A	Rotation velocity vector (rotation speed & direction)	float32
m	Motion Acceleration	float32
r	Rotation Acceleration	float32
f, F	Pressure	float32
b, B	Button-state	int32

Figure 69: SmartFiducial TUIO Protocol Specification

SERIAL PROTOCOL

Figure 70 outlines the serial-protocol developed for SmartFiducial communication. All data is sent to the vision tracking software in 6-byte message bundles.

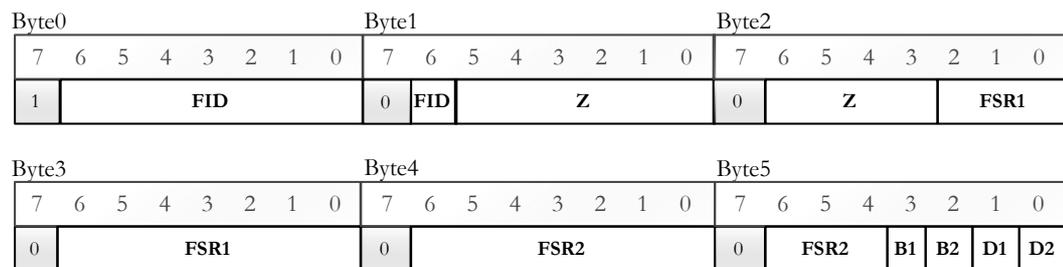


Figure 70: Overview of the SmartFiducial Serial Protocol

Fiducial ID has a resolution of 8-bits, yielding support for 255 unique fiducial IDs. All analog sensors (z-depth and pressure sensitivity) retain full 10-bit resolution, while digital inputs (buttons) use 1-bit respectively. An additional 2-bits (bits 0 and 1 in byte5) are reserved for two additional digital sensors in the future. Lastly, the most significant bit (MSB) in each of the six-bytes is reserved as a special

alignment bit, which is checked in CCV in order to ensure robustness and reliability of the wireless transmission.

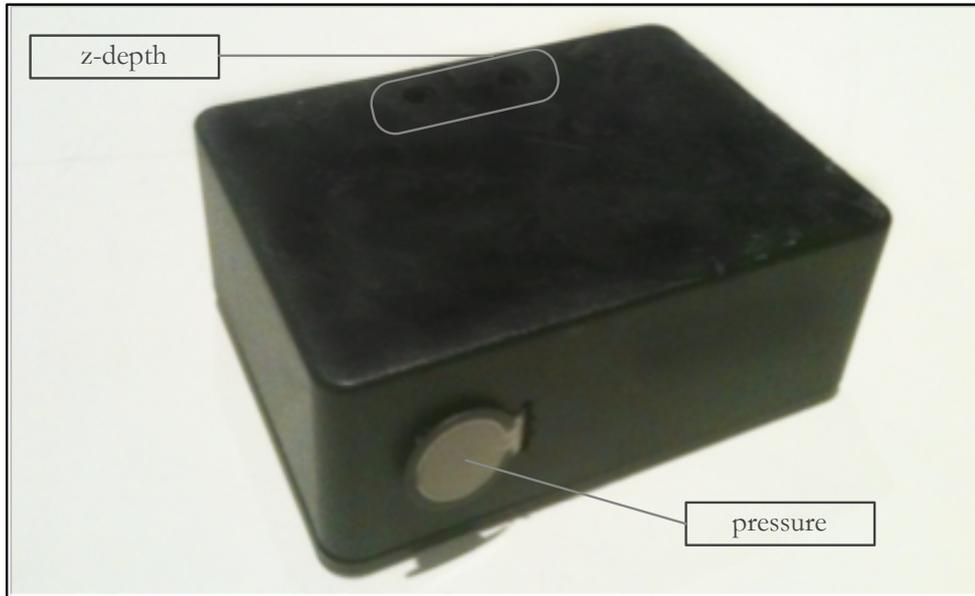


Figure 71: SmartFiducial Prototype (buttons 1 & 2 not pictured)

A.4.4 Software

In order to begin exploring the unique interactions afforded by the SmartFiducial, we have developed a basic wavetable synthesizer sequencer called *Turbine* (Figure 72). When a SmartFiducial is placed on the tabletop surface, sixteen “nodes” are created around the object. Each node represents a sixteenth note in a one-bar sequence, and dragging the node away from the SmartFiducial changes the pitch of the step. Using the z-depth sensing, the user is able to gesturally morph between the wavetable’s single-cycle waveforms, creating highly expressive, complex oscillations. Visual feedback is provided to the user via a soft Gaussian circle emitting from underneath the SmartFiducial. Currently the circle grows larger in size as the user nears the SmartFiducial’s proximity sensor, although the visual feedback may change as additional functionality is added to the application. In the future, we hope to expand *Turbine*’s functionality, including the interaction between multiple SmartFiducials, as well as enabling regular fiducial objects to act as sound modifiers, effects, and other types of intermediaries.

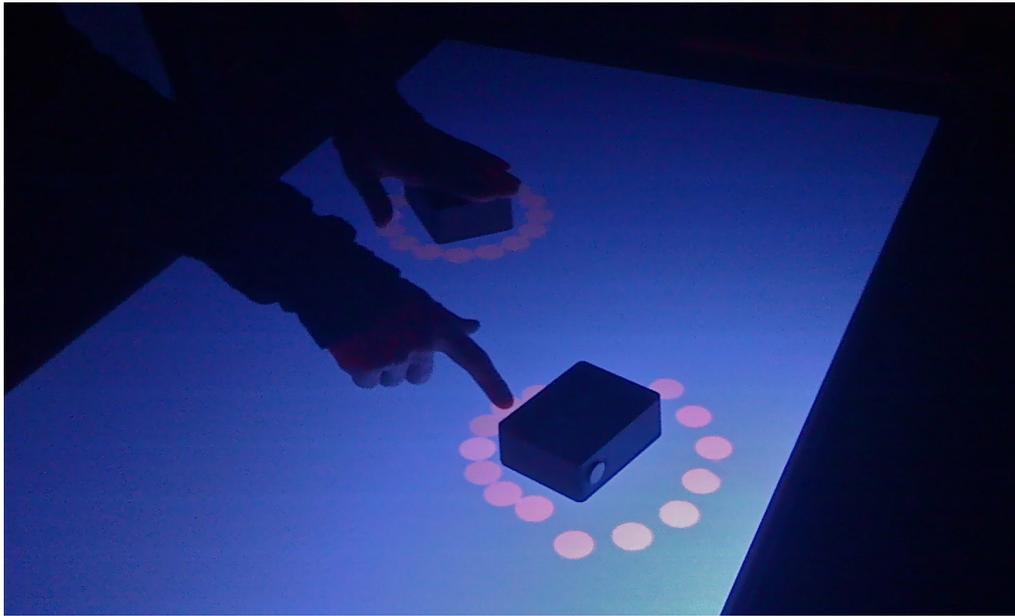


Figure 72: Two SmartFiducials being used with Turbine

8.6.2 DISCUSSION: SPATIAL RELATIONSHIPS AND TANGIBLE INTERFACES

Because tangible surface interaction happens along the x/y plane, (input) interaction is often a result of 2D spatial manipulation of the objects and the resulting relationships to both the tabletop surface and other objects. Once a tangible is placed within this location-dependent context however, e.g. when actions are tied to specific x, y coordinates on the surface, the x and y input freedoms are no longer useable (without changing the set relationships). In this situation, the only user-interaction possible is by rotating the object, or interacting with a virtual parameter displayed on the surface itself (assuming the surface is also touch-enabled). Although manipulating on screen parameters can often be effective for input, it poses many user interface (UI) challenges (clogging the UI, dealing with movable UI elements tied to the fiducials, etc.) and is often less than ideal. Additionally, proximity sensing and pressure based input offer a wide range of interaction affordances not possible by other means, as further discussed in the following section. Thus, the addition of z -depth proximity sensing and pressure sensitivity on the SmartFiducial allows tangible interaction to be more expressive in this situation, and other scenarios in the following ways:

- Adding complementary modes of input that can be utilized independently or simultaneously with traditional x, y, and rotational tangible interaction
- Beginning to address the loss of input modes in situations where the object must be placed in specific locations or when x,y spatial relationships and movement are primary means of surface interaction.

This greatly strengthens the ability of having dynamic relationships possible between tangible objects and the surface, and also between tangible objects and neighboring objects.

A.4.5 Discussion: New Affordances for Tabletop Interaction

Affordance theory, originally proposed by perceptual psychology pioneer J.J. Gibson introduces the idea that the potential utility of an object is based on the perceived qualities of the object by the subject (Gibson 1986). Whereas previous work in vision-based tangible tabletop surfaces has given users a set of interaction affordances defined by spatial relationships within a 2D environment, the SmartFiducial not only extends these affordances into the third dimension, but also offers additional affordances, governed by the unique cognitive notions of gesture based input. The following are a few of the interaction affordances that we have discovered through our initial experimentation with the SmartFiducial:

- Z-Depth proximity sensing may provide a more natural means of exploring 3D virtual environments on tabletop surfaces compared to traditional 2D interfaces.
- Both pressure sensitivity and proximity sensing offer the user means of highly gestural continuous control. These are very

different than common touch-based input gestures such as pinching, zooming, etc.

- Pressure sensitivity is not only gestural but may provide the user more tactile interaction and control over traditional tangible interaction, especially when employed in combination with other interaction techniques (for example, utilizing the pressure sensors simultaneously with moving and/or rotating the objects on the surface).
- Proximity and pressure sensors lend themselves particularly well to the application of a parameter modifier, non-dependent on the tabletop surface's GUI.

Additionally, the design of the SmartFiducial is influenced by Donald Norman's application of Affordance Theory to the field of Design, and Human Computer Interaction (Norman 1988). In accord with Norman's idea that the design of an object can be such that it suggests potential usage, our qualitative use of SmartFiducials has matured in its design in ways we believe optimize the SmartFiducial to be naturally used, without previous experience. This includes the interaction design decision to place the IR proximity sensor on the top of the SmartFiducial, and the pressure sensors on both sides of the SmartFiducial, typically where users grip the object. While of course there will always be a familiarization stage between the user and the software running on the tabletop surface, our initial exploratory testing showed that the users easily learned that there was a proximity sensor on the top of the SmartFiducial, and pressure sensors on the sides. As a result, they were able to very naturally exert a high-level of control and nuance in the use of the inputs.

A.4.6 Final Thoughts on Augmented Fiducial Objects

Building upon previous vision-based tangible surface interaction techniques (offering x, y, and rotational modes of input freedoms), the SmartFiducial is a novel tangible object that offers a new level of gesture and tactile affordances to

tangible tabletop interaction. While we present an initial exploratory application of these new input freedoms in the realm of music (Turbine), we believe the potentials enabled by the SmartFiducial can greatly enhance the user-experience when interacting with tangible tabletop surfaces across many disciplines and fields.

We are currently developing the Turbine synthesis engine to more thoroughly examine the affordances of the SmartFiducial in musical contexts. In the future we are particularly interested in conducting user-studies that explore our preliminary findings and experiences with the SmartFiducial, and will also hopefully illuminate new use cases and affordances of the SmartFiducial.

Additionally, we are excited to finally release the SmartFiducial and our branch of CCV into the community and see how others interpret and apply the new input freedoms.

A.5 Summary

While the research and analysis presented in this dissertation have looked at multimodal musical interaction in the laboratory and classroom, a core motivation has always been the application of such technologies in performance-based contexts. To that end, this appendix has presented various projects and experience reports in which multimodal techniques were used to enable live musical interactions. Initial motivations arose from an early concert in which additional modes of real-time interaction were desired (see A.1). This was later achieved in A.2, a piece that was composed and performed exclusively with three multimodal hyperinstruments. This performance was a particularly enlightening experience, and brought to light many challenges and affordances of working with multimodal systems for live computer music.

Multimodal techniques can also greatly benefit other multimedia scenarios, as investigated in the performance detailed in A.3, and the SmartFiducial in A.4. In both cases, multimodal approaches were used to heighten musical and multimedia interactions, overcoming limitations of particular sensing modalities. In *Transformations* (A.3), a synergy was created between the auditory modality, the vision tracking modality, and sensors on a dancer's body. Once combined, the

entire performance space became an interactive environment for the dancer and musician to engage. In the *SmartFiducial*, multidimensional levels of control and gesture were added to tabletop surfaces, enabling a more dynamic and expressive platform for collaborative musical interactions.

An in depth review of the particular affordances, implementations, challenges, design and interaction principles, and other experiences from the projects described can be found in their respective sections. In a broader scope, the projects and experiences detailed have created a number of situations in which the application of multimodality has been applied to real-time interactive contexts with musicians (and other artists). The experiences have helped solidify the overarching “why” of this research. Through multimodal interaction, it was possible to fulfill many desired interaction needs as a composer, performer, and interaction designer in the pieces described. Multimodal techniques enabled additional levels of control over musical parameters, resulting in more heightened musical experiences. This of course was not without its challenges and considerations. Most importantly, the projects presented are not without many potential possibilities—promising powerful, and more heightened musical experiences and interactions in the future.

Appendix B

Sensors

Overview of sensor technologies used in research

The research presented in this dissertation is multimodal in nature—combining the acoustical analysis of musical performance, with the analysis of various sensors on the instrument and performer. This section serves as a reference for the sensing technologies used in the research, and other sensors that are commonly used in musical HCI and physical computing.

B.1 Transducers

Nearly all sensors used in this research are a type of *transducer*. A transducer is simply a device that converts one form of energy (mechanical, electrical, magnetic, etc.) into another. Generally speaking, transducers output a variable electrical signal from some physical variable.

The most common transducers used in musical scenarios are microphones and loudspeakers. A microphone is essentially an acoustic-to-electrical transducer that converts sound (vibrations) to an electrical signal. Depending on the type of microphone (dynamic/moving-coil, condenser/capacitive, etc.) this is implemented slightly differently, although they operate under the same general principle. When a sound enters the microphone, the sound waves vibrate a diaphragm. In a dynamic or moving-coil microphone, the vibrating diaphragm moves a coil positioned in a magnetic field (created by a fixed magnet) which produces a varying electrical current in the coil via electromagnetic induction. A dynamic microphone essentially works like a normal loudspeaker, only in reverse.

Similarly, a condenser microphone also vibrates the diaphragm, however, the diaphragm instead acts as one plate of a capacitor. The distance between the diaphragm and another plate retains the voltage; when the sound vibrates the

diaphragm, the distance between the diaphragm and the plate varies, changing the voltage between the plates.

There are other types of microphone variations (electret, ribbon, etc.), and all are transducers. Transducers however are not just microphones and loudspeakers, they come in many forms of energy as illustrated in Figure 73.

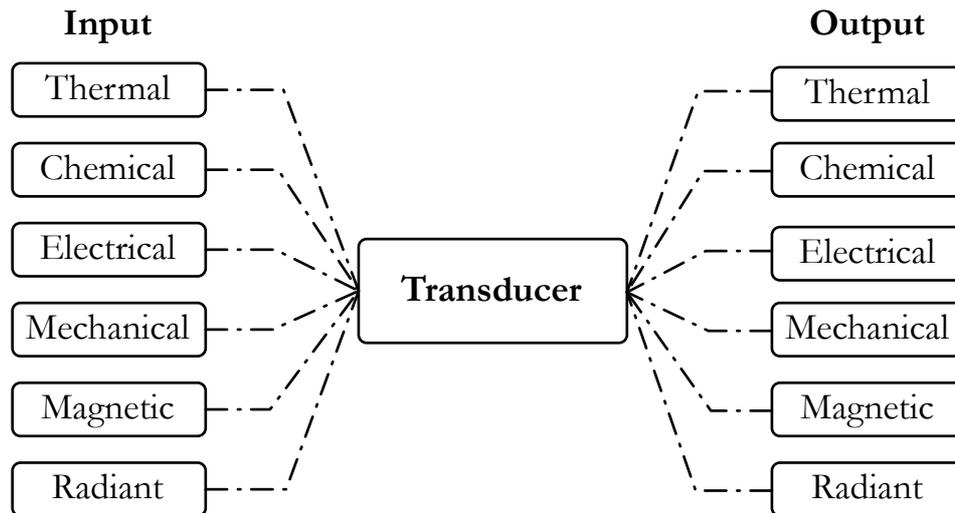


Figure 73: Overview of common forms of energy that transducers convert

B.2 Piezoelectric Sensors

Made from piezoelectric ceramics and single crystal materials, piezoelectric sensors are a type of transducer that converts mechanical measurements such as pressure, acceleration, strain, and force into electrical signals. Piezoelectric vibration sensors convert (wasted) energy from mechanical strain into electrical energy. Because piezoelectric sensors have a very high frequency response, and can sense and convert very small amounts of mechanical changes, they are extremely useful in musical applications and research. Common piezoelectric sensors include “contact” microphones, and the ceramic material in a record players stylus (the crystal flexes in the records grooves, resulting in a voltage).

B.3 Force-Sensing Resistors

Force-Sensing resistors (FSRs) are made from a conductive polymer film that decreases in resistance under the application of pressure and force onto its surface. Although FSR resolution is quite high ($\pm 0.5\%$ of full use force for most common FSR's), accuracy can vary depending on the setup consistency and can range anywhere from $\pm 5\%$ to $\pm 20\%$.

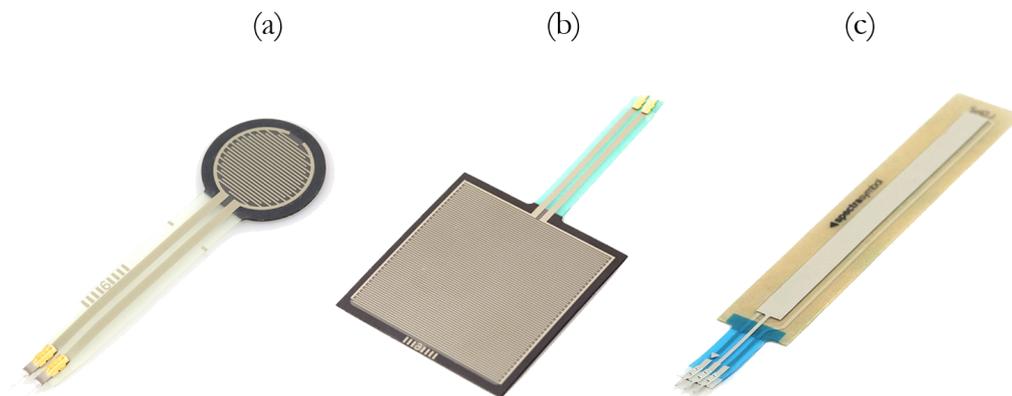


Figure 74: Common FSR shapes and configurations (force a and b, position c)

An advantage of FSRs, especially in musical contexts, is the ability to come in various sizes and configurations. Additionally, they are very flat, typically with thicknesses less than 0.5mm. Figure 74 shows three common FSR types; (a) is used as a thumb pressure sensor on the Esitar used in the performer recognition experiments, and (b) is commonly used for velocity sensitive drum triggers. The last FSR in the figure (c) is a special type of FSR called a linear softpot membrane potentiometer. These work similarly to regular FSRs, however, instead of changing resistance based on force applied anywhere on the surface, the membrane potentiometer changes the resistance based on the forces location along the strip.

B.4 Accelerometers

Accelerometers are a sensor with many applications, and have become very common in everyday consumer electronics. For example, they allow a cellphones screen to orient itself properly when the phone tilted on its side; to help combat data corruption, they direct a laptops hard-drive to lock when sensing that the computer may have been dropped; they are also used to convert a users motion into gestural control, for example, in the Nintendo Wii gaming console. As such accelerometers are powerful sensors that can capture the physical motion of musical performance.

Accelerometers come from a family of motion sensors (including gyroscopes), and specifically they measure *acceleration*. There are generally two types of acceleration, static or tilt (due to gravity), and dynamic or movement. There are many different ways accelerometers are constructed, including piezoelectric and capacitive techniques. In a piezoelectric approach, tiny crystal structures are stressed by accelerative forces, which cause a voltage to be generated. In the capacitive approach, capacitance is created between two microstructures. An accelerative force causes one to move, changing the capacitance, which then gets converted into a voltage.

Accelerometers come in a number of configurations and sensitivities. Acceleration is measured in meters per second squared (m/s^2), or g-force. Accelerometers are tuned to provide accurate acceleration measurements at either one or a configurable g-force. In general, the greater the g-force the accelerometer is capable of detecting, the lower the precision across the full-scale range of the accelerometer (upper and lower limits of detection), so it is important to select an accelerometer appropriately for the particular application. Additionally, accelerometers can measure acceleration in single, biaxial, and triaxial configurations (essentially one, two, or three 1D accelerometers in a single package). In musical applications and other gestural situations, biaxial (two) or triaxial (three) axis of acceleration is often desired.

Accelerometers also come in both analog and digital packages. Analog accelerometers produce a voltage output (for each axis), which is directly

proportional to the sensed acceleration. Digital accelerometers are normally more configurable, and communicate over a serial interface such as SPI or I²C.

Examples of accelerometers in the research can be found in the four sensor systems described in 3.1 and in the research found in Chapter 4, Chapter 5, Chapter 6, and Chapter 7. Additionally, accelerometers we used in the live performance scenarios documented in A.2, and A.3.

Appendix C

Communication Systems and Protocols

Information sharing through common languages

At the center of computer music are communication systems and protocols—information channels in which applications and hardware devices can exchange data with one another. Without protocols there would be no common language for electronic instruments or software applications to converse, and so establishing common languages is essential for computer music.

C.1 MIDI

The MIDI (Musical Instrument Digital Interface) protocol specification was created and adopted in the early 1980's to standardize communication between musical devices. Still in use today, the protocol was designed to communicate information such as pitch, velocity, control parameters such as vibrato, panning, as well as clock signals for synchronization and tempo information. Prior to the MIDI standard, most synthesizers, drum machines, sequencers, and other hardware devices, implemented propriety communication protocols to communicate with one another. This often meant that devices from different manufacturers could not talk with one another, or required other middle-ware systems to translate the messages between devices.

Although MIDI was originally conceptualized to communicate between hardware devices, the MIDI standard defines communication for both hardware and software, and today most music software has adopted the MIDI standard for communication. Most MIDI communication consists of a two or three-byte message which includes a status byte, followed by one or two data bytes. Status

bytes begin with a ‘1’ (e.g. 1xxxxxxx) and data bytes begin with a ‘0’ (e.g. 0xxxxxxx). Each byte is surrounded by a start and stop bit, resulting in each packet being a total of 10-bits long.

There are five MIDI message *formats*:

1. **Channel Voice:** Controls the instrument’s sixteen voices and is used to play notes, send CC (controller data), etc.
2. **Channel Mode:** Controls the way the device responds to incoming MIDI messages (monophonic/polyphonic, non-multitimbral/multitimbral).
3. **System Common:** Messages that must be sent across the entire MIDI system/network, to all devices, regardless of channel.
4. **System Real-Time:** Mostly used for synchronization and clocking of connected devices. These contain *only* status bytes (no data bytes), and include MIDI Clock, start, stop, continue, system reset, etc. Because they are timing critical, Real-Time Messages can be inserted into the middle of any multi-byte MIDI message.
5. **System Exclusive:** System Exclusive, or SysEx, is data designated to be used only by one piece of gear or manufacturer. SysEx was designed as a system in which non-standard MIDI messages could be sent to specific hardware units, for example remote patch editing, patch bank select, or parameters not supported by continuous controllers.

For more information on MIDI and the full MIDI specification, please refer to the official MIDI Manufacturers Association website²⁷.

C.2 Open Sound Control

Open Sound Control (OSC) is a network based communication protocol designed to take advantage of modern networking to deliver fast, descriptive, bidirectional communication between software applications, hardware devices,

²⁷ <http://www.MIDI.org>

and instruments. OSC is extremely useful for musical purposes because of its high speed, flexibility, and high resolution time stamping. Additionally, OSC is cross-platform and is available for most programming languages, making it a highly attractive communication protocol.

An advantage of OSC over other communication protocols (e.g. MIDI) is its descriptive and dynamic URL-style symbolic naming scheme. Information is communicated between devices or applications by packing and sending data over a URL-like address, e.g. `/myinstrument/sensor1/[data1 data2]` (where `data1` and `data2` are some data values which are being transmitted). This leads to another useful trait of OSC, which is that data can be bundled together and sent as a package (both `data1` and `data2` are sent at the same time in the example). How data is bundled is completely up to the developer and depends on the implementation and requirements of the system.

There are many other features that make OSC an extremely useful communication protocol for music and multimedia scenarios (e.g. sharing data and musical information with other people over a local computer network or the internet). For many of these reasons, it was important to include OSC support in the development of Nuance. Please refer to the Nuance section (3.2) for more information on motivations to include OSC directly within the software, and for more information on OSC refer to (Wright, Freed, and Momeni 2003) and www.opensoundcontrol.org.

C.3 TUIO

TUIO is an open framework that has been largely embraced by the online multi-touch interface community. Built on top of OSC, TUIO was designed to define a common protocol and API for tangible and multi-touch interfaces to send information such as touch and object events. TUIO is open source, cross-platform, and supported by nearly all of the available tangible and multi-touch vision tracking systems. A custom TUIO protocol was used in this research to broadcast information from the SmartFiducials to the computer in A.4.

Appendix D

Machine Learning

Teaching the computer to learn from experience

In the typical model of computing, machines can only execute operations in which they are explicitly programmed to perform. But what if a computer could learn from experience as humans do? This is the primary goal of machine learning, an exciting branch of artificial intelligence, which aims to create algorithms that can evolve, or learn behaviors. This is achieved by creating machine learning algorithms that are capable of deducing the complex relationships that exist within sample training data. The training data however cannot possibly account for all instances of all unknown inputs, and so a main goal of machine learning algorithms is to *generalize* the relationships in the data as much as possible. In this way the algorithm can be optimized to make correct decisions (outputs) with new and unknown cases (inputs). As this work in this dissertation focused on a particular type of machine learning scenario called “supervised learning”, the following sections provide an overview of supervised learning, key terminology, and a brief description of the algorithms used in the research.

D.1 Supervised Learning

Supervised learning is a branch of machine learning in which an algorithm produces a mathematical model (function), which can produce some output, given some input (data) (that it may or may not have seen before). As illustrated in Figure 75, the algorithm learns to infer the model by training on a data set of previous observations (input and output pairs). Once the model has been trained, new inputs are fed into the model, which then computes a new output. The data

that is input into the algorithm is a feature-vector of observation examples, with their corresponding outputs.

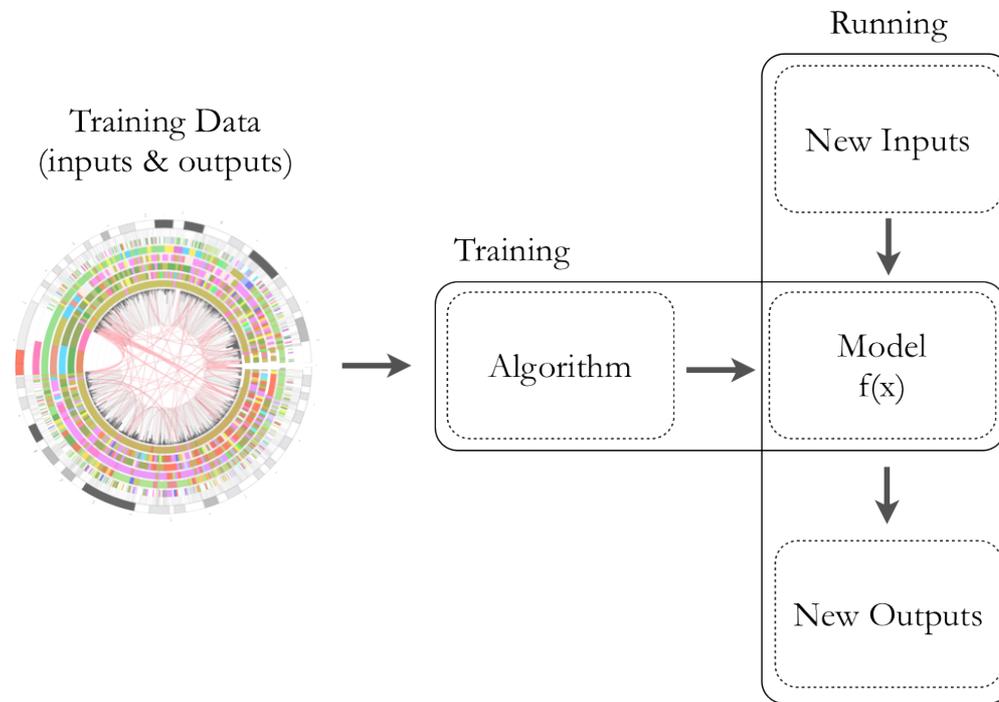


Figure 75: Illustration of a typical supervised learning flow, adapted from (Fiebrink 2011)

For an example of a typical classification task, let's revisit the performer recognition experiments from Chapter 4. The training sets included feature vectors (a list of attributes) of the player's performance features, which were extracted from the audio and sensor data. The feature vectors included acoustical measures extracted from the audio, raw sensor values, and other computed statistics. Each feature vector in the training set was labeled with the corresponding label for each performer (the "output" in a performer classification scenario). Given the labeled training set, the algorithm generalizes and learns the relationships of the features, builds the model, and can assign a label (e.g. predicting who the performer is) to some new unlabeled feature vector input.

Classification tasks are not the only types of supervised learning problems. Whereas in a classification scenario the output is a member of a finite set (in the performer recognition example the output could only be one of the performers in the training set), in a *regression* problem, the output can be any real value. A classic example of regression is predicting the selling price of a house given some set of features (size of the house, number of rooms, number of bathrooms, neighborhood, selling price of other homes in the area, other market statistics, etc.). Given these set of features, one might want to predict the optimal value to sell the house at. As the research in this dissertation does not explicitly deal with regression problems, this section won't look into regression any further, and instead will turn to explaining the other terms, methods, and algorithms (classifiers) that were used in this research.

D.2 *K*-fold Cross-Validation

Cross-validation is a technique used in machine learning and statistical analysis which helps achieve greater generalization and accuracy in the output (results); it helps to reduce over-fitting when a separate validation (test) set is not available.

For example supposed one were to attempt performer classification as per Chapter 4. Because machine learning algorithms normally attempt to create a model which best fits the training set, testing on validation data from the same population (the training set) may result in over-fitting. To overcome this problem, when an independent test set is not available, *k*-fold cross-validation can be used. In *k*-fold cross validation, the training set is partitioned into *k* subsamples (sets). The model is trained using *k* - 1 of the subsamples, and the left over set is used as an independent test (validation) set. This process is repeated over the number of folds (*k* times), and the results from each fold are averaged. Each of the subsamples is used for validation only once, which ensures that all observations (instances) are used for both training and testing.

D.3 Algorithms

There are many different algorithms for generalizing models, which are suited for different applications and tasks. This section will provide introductions to the various families of algorithms used in the machine learning scenarios presented the research. For more detailed information on the specific algorithms, and others, refer to (Witten, Frank, and Hall 2011).

D.3.1 Decision Trees

Decision trees are predictive models in which tree-like data structures are created to predict the output values. In decision tree algorithms, starting from the root, *branches* represent features that lead to *leaves*—or target output attributes. Decision trees are particular appropriate for binary classification problems, or other scenarios where there are fixed sets of output attributes or real values.

As an example, suppose one would like to determine the genre of an audio recording given the following training data (instruments represent the feature vectors or the input, and genre is the corresponding classification label or output class):

Table 25: Sample feature set of instruments for genre classification problem using a decision tree classifier

Instruments	Genre
Electric guitar, drums, bass (upright), piano (electric)	Jazz
Electric guitar, drums, bass (electric), piano (electric)	Rock
Electric guitar, drums, bass (electric), piano (acoustic)	Jazz
Acoustic guitar, drums, Electric bass, piano (acoustic)	Rock

In such a case, a tree may be constructed as in Figure 76. In this example, each instrument type would be a *node* of the tree, or *attribute* of the instances feature vector. The branches (e.g. electric guitar or acoustic guitar) represent the values, which lead to leaf nodes, or *class* of the instance (e.g. the genre).

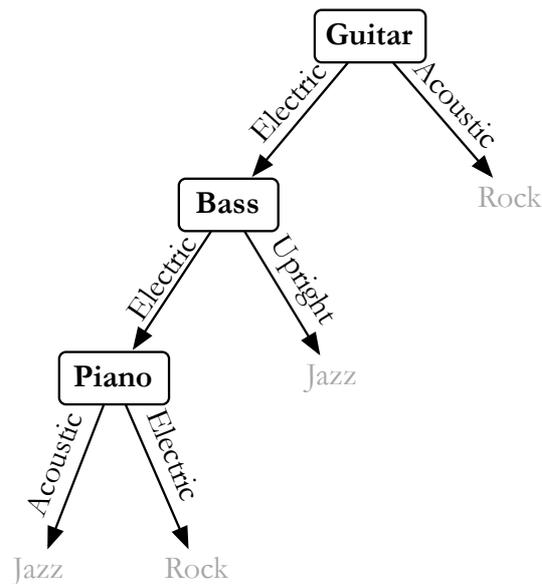


Figure 76: Sample decision tree constructed from the feature set of instruments (for genre classification) listed in Table 25

A benefit of decision trees is that they are whitebox systems—you can visually see the relationships that represent the data. While they are easy to understand and visualize, they can be prone to data over-fitting (whereas they do not *generalize* the data relationships enough), although most algorithms typically employ various *pruning* techniques to optimize their generality.

D.3.2 Naive Bayes

Naive Bayes is a simple probabilistic classifier based on the Bayesian theorem, which calculates the probability of an event occurring given the probability of another event that has already occurred. Each variable (feature) is considered independently from all other features, regardless of relationships that may or may not exist between them in the feature space. Although having simplified assumptions, Naive Bayes continues to work very well in complex real-world machine learning problems, with the benefit of short training times. Depending on the problem however, covariance between features can sometimes greatly increase performance and so another algorithm may outperform Naive Bayes.

D.3.3 k-Nearest Neighbor (kNN)

A kNN or *k-nearest neighbor* algorithm is a classification algorithm that classifies objects based on the proximity of training examples in the feature space. A typical implementation involves a distance function that calculates the distance of the test-points' features to the feature sets of the training data (e.g. using Euclidian distance or another distance metric). In a “majority wins” approach, the class with the maximum k -nearest neighbors within the boundaries determines the class to be assigned to the test point.

As an example, Figure 77 illustrates a typical binary classification problem where the rounded rectangle (labeled ‘?’) in the center is an unknown test-point (input) which one would like to assign a class label (class 1 or class 2). If $k = 3$, the test-point would be labeled as class 2, as the three nearest neighbors includes two from class two, and only one from class three. It is common to use more advanced distance functions, or weight the class instances differently, e.g., weighting the class instances less the further away they are from the test point.

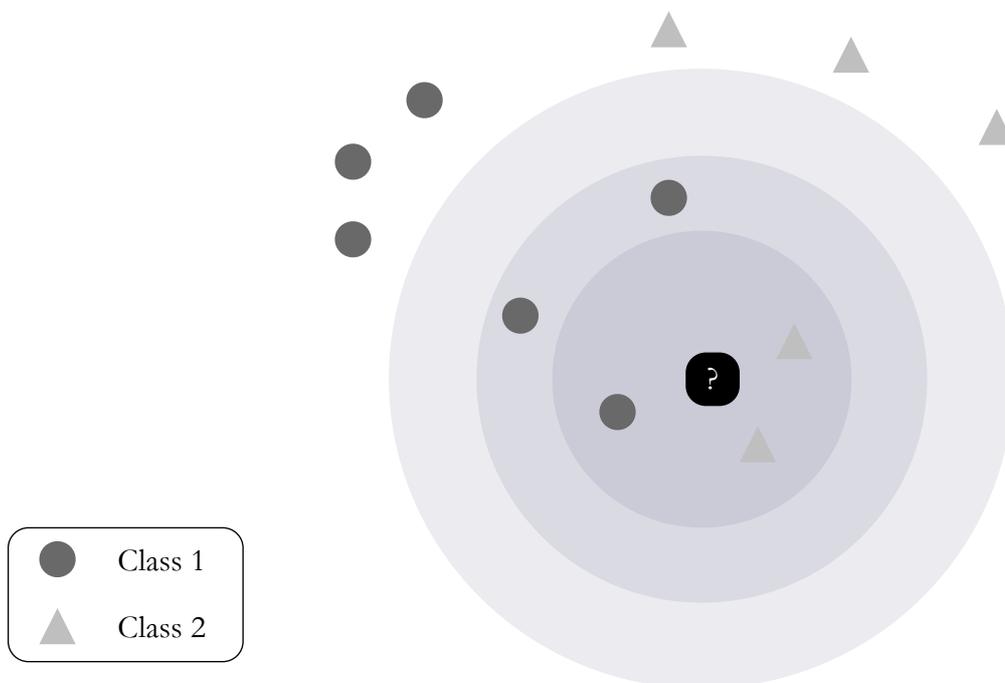


Figure 77: Illustration of kNN classifier where there are two classes (class 1 and class 2), and a rounded rectangle marked ‘?’ in the center is the test or prediction point

D.3.4 Artificial Neural Networks (ANNs)

Artificial neural networks (ANN) are a type of algorithm modeled off of the biological interconnectivity of the human brain. ANNs are typically adaptive systems that can model highly complex relationships between inputs and outputs, finding patterns in data. While much about the human brain is yet to be discovered, ANNs operate under the following assumption of the brains biological functioning: the brain is composed of a complex interconnected structure of neurons. The connections between neurons continually adjust themselves as humans learn or gain new experiences. As signals are sent between neurons, the influence a signal has at a receiving neuron is a parameter that changes as one learns, altering the neurons output to other neurons, etc.

As such, a simple binary classification neural network is illustrated in Figure 78. In the example, the input neurons (features) are interconnected through one *hidden layer* of neurons, which activate through weighted *activation functions*, ultimately leading to one of two outputs (e.g. class 1 or class 2).

In the research presented in this dissertation, we typically used a multilayer perception (MLP), which is a feedforward artificial neural network. Each node in a layer connects to every other node in the following layer with a certain weight w_{ij} . While training the MLP, the weights are updated during a process called *backpropagation*, which has the explicit goal of minimizing the output errors and updating the weights of the hidden layer activation functions using gradient descent.

An ANN can theoretically support any number of hidden layers, with non-linear activation functions. As such, neural networks perform extremely well with complex data, and large feature spaces.

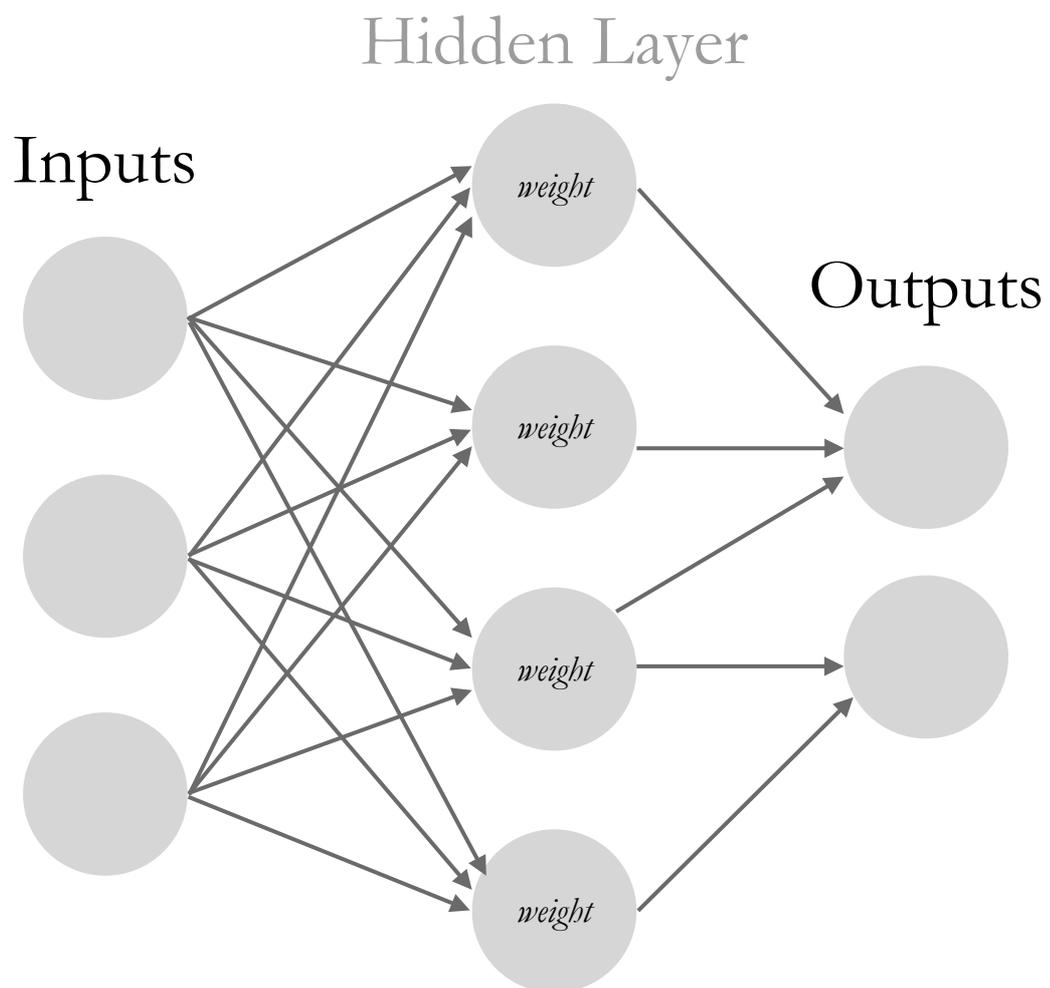


Figure 78: Simple artificial neural network with one hidden layer

Appendix E

Refereed Academic Journals and Publications

1. Hochenbaum, J., Kapur, A. *Nuance: A Software Tool for Capturing Synchronous Data Streams From Multimodal Musical Systems*. In Proceedings of the International Computer Music Conference. Ljubljana, Slovenia. September 9-15, 2012.
2. Hochenbaum, J., Kapur, A. *Improving Onset Detection Accuracy in Non-Percussive Sounds Using Multimodal Fusion*. In Proceedings of the Australasian Computer Music Conference. Brisbane, Australia. July 12-15, 2012.
3. Hochenbaum, J., Kapur, A. *Drum Stroke Computing: Multimodal Signal Processing for Drum Stroke Identification and Performance Metrics*. In Proceedings of the International Conference on New Interfaces for Musical Expression. Ann Arbor, Michigan. May 21-23, 2012.
4. Vallis, O., Diakopoulos, D., Hochenbaum, J., Kapur, A. *Building on the Foundations of Network Music: Exploring Interaction Contexts and Shared Robotic Instruments*. Organised Sound. Volume 17, Issue 1. 2012.
5. Kapur, A., Darling, M., Diakopoulos, D., Murphy, J., Hochenbaum, J., Vallis, O., Bahn, C. *The Machine Orchestra: An Ensemble of Human Laptop Performers and Robotic Musical Instruments*. Computer Music Journal. Volume 35, Issue 4. 2011.
6. Vallis, O., Hochenbaum, J., Murphy, J., Kapur, A. *The Chronome: A Case Study in Designing New Continuously Expressive Musical Instruments*. In Proceedings of the Australasian Computer Music Conference. Auckland, New Zealand. 2011.
7. Hochenbaum, J., Kapur, A. *Adding Z-Depth and Pressure Expressivity to Tangible Tabletop Surfaces*. In Proceedings of the International Conference on New Interfaces for Musical Expression. Oslo, Norway. May 30 – June 1, 2011.
8. Kapur, A., Darling, M., Murphy, J., Hochenbaum, J., Diakopoulos, D. *The KarmetiK NotomotoN: A New Breed of Musical Robot for Teaching and*

- Performance*. In Proceedings of the International Conference on New Interfaces for Musical Expression. Oslo, Norway. May 30 – June 1, 2011.
9. Hochenbaum, J., Kapur, A., and Wright, M., *Multimodal Musician Recognition*. In Proceedings of the International Conference on New Interfaces for Musical Expression, Sydney, Australia, June 2010.
 10. Hochenbaum, J., Vallis, O., Diakopoulos, D., Murphy, J., and Kapur, A., *Designing Expressive Musical Interfaces for Tabletop Surfaces*. In Proceedings of the International Conference on New Interfaces for Musical Expression, Sydney, Australia, June 2010.
 11. Vallis, O., Hochenbaum, J., and Kapur, A., *A Shift Towards Iterative and Open-Source Design for Musical Interfaces*. In Proceedings of the International Conference on New Interfaces for Musical Expression, Sydney, Australia, June 2010.
 12. Kapur, A., Darling, M., Wiley, M., Vallis, O., Hochenbaum, et al. *The Machine Orchestra* Proceedings of the International Computer Music Conference, New York City, New York, June 2010.
 13. Diakopoulos, D., Vallis O., Hochenbaum J., Murphy, J., Kapur, A. *21st Century Electronica: MIR Techniques for Classification and Performance*. Proceedings of the 2009 International Society on Music Information Retrieval Conference. Kobe, Japan. October 2009. **Winner of Best Poster Presentation Award.**
 14. Hochenbaum, J., Vallis, O., *Bricktable: A Musical Tangible Multi-Touch Interface*. In Proceedings of the 2009 Berlin-Open Conference. Berlin, Germany. 2009.
 15. Hochenbaum, J., Vallis, O., Diakopoulos, D., Akten, M., Murphy, J. *Musical Applications for Multi-Touch Interfaces*. Workshop on Media Arts, Science, and Technology. (January 29-30, 2009). MAST. UCSB, Santa Barbara, CA. 2009
 16. Vallis, O., Hochenbaum, J., and Kapur, A. 2008. *Extended Interface Solutions for Musical Robotics*. Tenth IEEE International Symposium on Multimedia, ISM. IEEE Computer Society, pp. 495-496. 2008.

Bibliography

References, History, Citations

- Akbar Khan, Ali. 2004. *The Classical Music of North India: The Music of the Baba Allauddin Gharana as Taught by Ali Akbar Khan at the Ali Akbar College of Music*. Munshiram Manoharlal.
- Amatriain, Xavier, Pau Arumi, and David Garcia. 2006. "CLAM: a Framework for Efficient and Rapid Development of Cross-platform Audio Applications." In *Proceedings of the 14th Annual ACM International Conference on Multimedia*, 951–954. New York, NY, USA: ACM.
- Anon. 2006. "MIREX Audio Onset Detection Evaluation Results." *2006:Audio Onset Detection Results*. October 12. http://www.music-ir.org/mirex/wiki/2006:Audio_Onset_Detection_Results.
- Anon. 2007. "MIREX Audio Onset Detection Evaluation Results." *2007:Audio Onset Detection Results*. http://www.music-ir.org/mirex/wiki/2007:Audio_Onset_Detection_Results.
- Anon. 2009. "MIREX Audio Onset Detection Evaluation Results." *2009:Audio Onset Detection Results*. http://www.music-ir.org/mirex/wiki/2009:Audio_Onset_Detection_Results.
- Aryafar, Kamelia, and Ali Shokoufandeh. 2011. "Music Genre Classification Using Explicit Semantic Analysis." In *Proceedings of the 1st International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies*, 33–38. MIRUM '11. New York, NY, USA: ACM.
- Askenfelt, Anders. 1989. "Measurement of the Bowing Parameters in Violin Playing. II: Bow–bridge Distance, Dynamic Range, and Limits of Bow Force." *The Journal of the Acoustical Society of America* 86 (2): 503–516.
- Bayes, Mr., and Mr. Price. 1763. "An Essay Towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S." *Philosophical Transactions* 53 (January 1): 370–418.
- Bell, Bo, Jim Kleban, Dan Overholt, Lance Putnam, John Thompson, and JoAnn Kuchera-Morin. 2007. "The Multimodal Music Stand." *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*.
- Bello, JP, L Daudet, S Abdallah, C Duxbury, M Davies, and MB Sandler. 2005. "A Tutorial on Onset Detection in Music Signals." *Speech and Audio Processing, IEEE Transactions On* 13 (5): 1035–1047.
- Benning, Manjinder Singh, Ajay Kapur, Bernie C Till, and George Tzanetakis. 2007. "Multimodal Sensor Analysis of Sitar Performance: Where Is the Beat?" In *2007 IEEE 9th Workshop on Multimedia Signal Processing*, 74–77.
- Berio, Luciano. 1999. "Cronaca Del Luogo". Opera August, Salzburg Festival.
- Boersma, Paul. 1993. "Accurate Short-Term Analysis of the Fundamental Frequency and the Harmonics-To-Noise Ratio of a Sampled Sound." *Proc. Institute of Phonetic Sciences* Volume 17: 97–110.

- Bofinger, I., and G. Whateley. 2002. "iCon—the Realisation of 'the Virtual Conservatorium'." *ISME Commission for the Education of the Professional Musician*: 1–15.
- Bongers, Bert, and Gerrit C. Veer. 2007. "Towards a Multimodal Interaction Space: Categorisation and Applications." *Personal Ubiquitous Comput.* 11 (8) (December): 609–619.
- Bouillot, N., and J. R. Cooperstock. 2009. "Challenges and Performance of High-fidelity Audio Streaming for Interactive Performances." In *Proceedings of the 9th International Conference on New Interfaces for Musical Expression*.
- Brecht, B., and G. Garnett. 1995. "Conductor Follower." In *Proceedings of the International Computer Music Conference*.
- Busso, Carlos, Zhigang Deng, Serdar Yildirim, Murtaza Bulut, Chul Min Lee, Abe Kazemzadeh, Sungbok Lee, Ulrich Neumann, and Shrikanth Narayanan. 2004. "Analysis of Emotion Recognition Using Facial Expressions, Speech and Multimodal Information." In *Proceedings of the 6th International Conference on Multimodal Interfaces*, 205–211. ICMI '04. New York, NY, USA: ACM.
- Camurri, Antonio, Paolo Coletta, Giovanna Varni, and Simone Ghisio. 2007. "Developing Multimodal Interactive Systems with EyesWeb XMI." In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, 305–308. NIME '07. New York, NY, USA: ACM.
- Cano, Pedro, Markus Koppenberger, and Nicolas Wack. 2005. "Content-based Music Audio Recommendation." In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, 211–212. MULTIMEDIA '05. New York, NY, USA: ACM.
- Cataltepe, Zehra, Yusuf Yaslan, and Abdullah Sonmez. 2007. "Music Genre Classification Using MIDI and Audio Features." *EURASIP Journal on Advances in Signal Processing* 2007 (1): 574–579.
- De Cheveign'e, Alain, and Hideki Kawahara. 2002. "YIN, a Fundamental Frequency Estimator for Speech and Music." *The Journal of the Acoustical Society of America* 111 (4): 1917–1930.
- Cheveigné, Alain de. 2006. "Multiple F0 Estimation." In *Computational Auditory Scene Analysis*, edited by Wang and Brown. IEEE Press Wiley-Interscience.
- Collins, Nick. 2006. "Investigating Computational Models of Perceptual Attack Time." In *Proceedings of the 9th International Conference on Music Perception & Cognition (ICMPC)*, 923–929. Bologna, Italy.
- Collins, Nick. 2010. *Introduction to Computer Music*. John Wiley & Sons Ltd.
- Cook, Perry. 2001. "Principles for Designing Computer Music Controllers." In , 1–4. National University of Singapore.
- Cook, Perry. 2009. "Re-Designing Principles for Computer Music Controllers: A Case Study of SqueezeVox Maggie." *Proceedings of the International Conference on New Interfaces for Musical Expression* 11: 218–221.
- Dahl, Sofia. 2005. "On the Beat: Human Movement and Timing in the Production an Perception of Music". Ph.D, Royal Institute of Technology.

- Dannenberg, Roger B., Belinda Thom, and David Watson. 1997. "A Machine Learning Approach to Musical Style Recognition." In *Proc. International Computer Music Conference*, 344–347.
- Degara-Quintela, Norberto, Pena, Antonio, and Torres-Guijarro, Soledad. 2009. "A Comparison of Score-Level Fusion Rules For Onset Detection In Music Signals." In *10th International Society for Music Information Retrieval Conference*.
- Diakopoulos, Dimitri, Owen Vallis, Jordan Hochenbaum, Jim Murphy, and Ajay Kapur. 2009. "21st Century Electronica: MIR Techniques for Classification and Performance." In *Proceedings of the 2009 International Society on Music Information Retrieval Conference*. Kobe, Japan.
- Dix, Alan, Janet E. Finlay, Gregory D. Abowd, and Russell Beale. 2003. *Human-Computer Interaction*. 3rd ed. Prentice Hall.
- Dixon, Simon. 2001. "Automatic Extraction of Tempo and Beat From Expressive Performances." *Journal of New Music Research* 30 (1): 39 – 58.
- Dixon, Simon. 2006. "Onset Detection Revisited." In *Proceedings of the 9th International Conference on Digital Audio Effects (DAFX'06)*. Montreal, Canada.
- Dixon, Simon. 2007. "Evaluation of the Audio Beat Tracking System BeatRoot." *Journal of New Music Research* 36 (1) (March): 39–50.
- Dolhansky, Brian, Andrew Mcpherson, and Kim Youngmoo. 2011. "Designing an Expressive Virtual Percussive Instrument." In *Proceeding of the Sound and Music Computing Conference*.
- Duignan, Matthew, James Noble, and Robert Biddle. 2010. "Abstraction and Activity in Computer-mediated Music Production." *Computer Music Journal (CMJ)* 34 (4): 22–33.
- Dumas, Bruno, Denis Lalanne, and Sharon Oviatt. 2009. "Multimodal Interfaces: A Survey of Principles, Models and Frameworks." In *Human Machine Interaction*, edited by Denis Lalanne and Jürg Kohlas, 3–26. Berlin, Heidelberg: Springer-Verlag.
- Eggink, Jana, and Guy Brown. 2003. "Application of Missing Feature Theory to the Recognition of Musical Instruments in Polyphonic Audio." In *Proceedings of the International Society on Music Information Retrieval Conference*. The Johns Hopkins University. #.
- Erskine, Jason, and Ajay Kapur. 2011. "Extended Techniques for Indonesian Performance." In *Proceedings of the Australasian Computer Music Conference*. Auckland, New Zealand.
- Ferguson, Sam, Kirsty Beilharz, and Claudia Calò. 2012. "Navigation of Interactive Sonifications and Visualisations of Time-series Data Using Multi-touch Computing." *Journal on Multimodal User Interfaces* 5 (3): 97–109.
- Fiebrink, Rebecca. 2011. "Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance". Ph.D, Princeton, NJ, USA: Princeton University.
http://www.cs.princeton.edu/~fiebrink/Rebecca_Fiebrink/thesis.html.
- Fitzgerald, Derry. 2004. "Automatic Drum Transcription and Source Separation". Ph.D Thesis, Dublin: Dublin Institute of Technology.
- Gillet, O., and G. Richard. 2008. "Transcription and Separation of Drum Signals From Polyphonic Music." *IEEE Transactions on Audio, Speech, and Language Processing* 16 (3) (March): 529–540.

- Glinsky, Albert. 2000. *Theremin: Ether Music and Espionage*. University of Illinois Press.
- Goto, Masataka, and Yoichi Muraoka. 1994. "A Sound Source Separation System for Percussion Instruments." In *Transactions of the Institute of Electronics, Information and Communication Engineers*, J77-D-II:901–911.
- Goto, Masataka, and Yoichi Muraoka. 1999. "Real-time Beat Tracking for Drumless Audio Signals: Chord Change Detection for Musical Decisions." *Speech Commun.* 27 (3-4): 311–335.
- Hasan, Leila, Nicholas Yu, and Joseph A. Paradiso. 2002. "The Termenova: a Hybrid Free-gesture Interface." Proceedings of the 2002 Conference on New Interfaces for Musical Expression.
- Henzie, Charles A. 1960. *Amplitude and Duration Characteristics of Snare Drum Tones*. Indiana University.
- Herrera, P, A. Klapuri, and M. Davy. 2006. "Automatic Classification of Pitched Musical Instrument Sounds." In *Signal Processing Methods for Music Transcription*. Springer US.
- Hochenbaum, Jordan, and Ajay Kapur. 2012. "Drum Stroke Computing: Multimodal Signal Processing for Drum Stroke Identification and Performance Metrics." In *International Conference on New Interfaces for Musical Expression*.
- Hochenbaum, Jordan, Ajay Kapur, and Matt Wright. 2010. "Multimodal Musician Recognition." In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Sydney, Australia.
- Hochenbaum, Jordan, and Owen Vallis. 2009. "Bricktable: A Musical Tangible Multi-Touch Interface." In *Proceedings of Berlin Open Convergence 09'*. Berlin, Germany.
- Hochenbaum, Jordan, Owen Vallis, Dimitri Diakopoulos, Memo Akten, and Jim Murphy. 2009. "Musical Applications for Multi-Touch Interfaces." In *Workshop on Media Arts, Science, and Technology (MAST)*. Santa Barbara, California.
- Hochenbaum, Jordan, Owen Vallis, Dimitri Diakopoulos, James Murphy, and Ajay Kapur. 2010. "Designing Expressive Musical Interfaces for Tabletop Surfaces." In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Sydney, Australia.
- Hunt, Andy, Marcelo M Wanderley, and Matthew Paradis. 2002. "The Importance of Parameter Mapping in Electronic Instrument Design." In , 1–6. National University of Singapore.
- Hunt, Andy, and Marcelo M. Wanderley. 2002. "Mapping Performer Parameters to Synthesis Engines." *Organised Sound* 7 (2) (August): 97–108.
- Izadi, Shahram, Steve Hodges, Stuart Taylor, Dan Rosenfeld, Nicolas Villar, Alex Butler, and Jonathan Westhues. 2008. "Going Beyond the Display: a Surface Technology with an Electronically Switchable Diffuser." In *UIST '08: Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, 269–278. Monterey, CA, USA: ACM.
- Jegade, O., and G. Shive. 2001. *Open and Distance Education in the Asia Pacific Region*. Open University of Hong Kong Press.
- Johnston, Blake, and Ajay Kapur. 2012. "EZither: Extended Techniques for Customised Digital Bowed String Instrument." In *Proceedings of the Australasian Computer Music Conference*. Brisbane, Australia.

- Jordà, Sergi, Martin Kaltenbrunner, Günter Geiger, and Ross Bencina. 2005. "The reacTable." In *Proceedings of the International Computer Music Conference*. Barcelona, Spain.
- Kaltenbrunner, Martin, and Ross Bencina. 2007. "reacTIVision: a Computer-vision Framework for Table-based Tangible Interaction." *Proceedings of the 1st International Conference on Tangible and Embedded Interaction*.
- Kaltenbrunner, Martin, Till Bovermann, Ross Bencina, and Enrico Costanza. 2005. "TUIO - A Protocol for Table Based Tangible User Interfaces." In *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation*.
- Kapoor, Ashish, and Rosalind W. Picard. 2005. "Multimodal Affect Recognition in Learning Environments." *Proceedings of the 13th Annual ACM International Conference on Multimedia*.
- Kapur, Ajay. 2008. *Digitizing North Indian Music: Preservation and Extension Using Multimodal Sensor Systems, Machine Learning and Robotics*. VDM Verlag.
- Kapur, Ajay, Michael Darling, Dimitri Diakopoulos, Jim W. Murphy, Jordan Hochenbaum, Owen Vallis, and Curtis Bahn. 2011. "The Machine Orchestra: An Ensemble of Human Laptop Performers and Robotic Musical Instruments." *Computer Music Journal* 35 (4): 49–63.
- Kapur, Ajay, Graham Percival, Mathieu Lagrange, and George Tzanetakis. 2007. "Pedagogical Transcription For Multimodal Sitar Performance." In *Proceedings of the International Society on Music Information Retrieval Conference*.
- Kapur, Asha, Ajay Kapur, Naznin Virji-Babul, George Tzanetakis, and Peter Driessen. 2005. "Gesture-Based Affective Computing on Motion Capture Data." In *Affective Computing and Intelligent Interaction*, 1–7. Springer.
- Kessous, Loic, Stelios Asteriadis, Ginevra Castellano, Kostas Karpouzis, George Caridakis, Amaryllis Raouzaïou, and Lori Malatesta. 2010. "Multimodal Emotion Recognition from Expressive Faces, Body Gestures and Speech." *International Federation for Information Processing Digital Library* 247 (1) (August 25).
- Kitahara, Tetsuro, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. 2007. "Instrument Identification in Polyphonic Music: Feature Weighting to Minimize Influence of Sound Overlaps." *EURASIP J. Appl. Signal Process.* 2007 (1) (January): 155–155.
- Koons, David B., Carlton J. Sparrell, and Kristinn R. Thorisson. 1993. "Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures." In *Intelligent Multimedia Interfaces*, edited by Mark T. Maybury, 257–276. Menlo Park, CA, USA: American Association for Artificial Intelligence.
- Lancaster, H. 2007. "Music from Another Room: Real-time Delivery of Instrumental Teaching." In *NACTMUS National Conference, Queensland Conservatorium of Music, Brisbane, 29 June-1 July 2007*.
- Lartillot, Olivier. 2011. "MIRtoolbox 1.3.4 User's Manual". Finnish Centre of Excellence in Interdisciplinary Music Research, University of Jyväskylä, Finland.
- Lartillot, Olivier, Eerola, Tuomas, Toivainen, Petri, and Fornari, Jose. 2008. "A Unifying Framework for Onset Detection, Tempo Estimation, and Pulse Clarity Prediction." In *11th International Conference on Digital Audio Effects*. Espoo, Finland.

- Little, David, and Bryan Pardo. 2008. "Learning Musical Instruments from Mixtures of Audio with Weak Labels." In *Proceedings of the International Society on Music Information Retrieval Conference*, 127–132.
- Livshin, Arie A., and Xavier Rodet. 2004. "Musical Instrument Identification in Continuous Recordings." In *In Proc. of DAFX*.
- Machover, Tod. 1992. *Hyperinstruments - A Progress Report 1987 - 1991*. MIT Media Laboratory.
- Machover, Tod, and Joe Chung. 1989. "Hyperinstruments: Musically Intelligent and Interactive Performance and Creativity Systems." In *Proceedings of the International Computer Music Conference*, 186–190.
- Mathews, Max, and Andrew Schloss. 1989. "The Radio Drum as a Synthesizer Controller." In *Proceedings of the 1989 International Computer Music Conference (ICMC)*.
- Mayer, Rudolf, and Andreas Rauber. 2011. "Musical Genre Classification by Ensembles of Audio and Lyrics Features." In *Proceedings of the International Society on Music Information Retrieval Conference*, 56:675–680.
- McConnell, Steve. 2004. *Code Complete: A Practical Handbook of Software Construction*. 2nd ed. Microsoft Press.
- McGurk, H., and J. MacDonald. 1976. "Hearing Lips and Seeing Voices." *Nature* 264 (5588): 746–748.
- Miranda, Eduardo Reck, and Marcelo M Wanderley. 2006. *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. Vol. 21. The Computer Music and Digital Audio Series. A-R Editions.
- Murphy, James, Ajay Kapur, and Carl Burgin. 2010. "The Helio: A Study of Membrane Potentiometers and Long Force Sensing Resistors." In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Sydney, Australia.
- Neal, J.G., and Shapiro, S.C. 1991. "Intelligent Multi-media Interface Technology." In *Intelligent User Interfaces*, 11–43.
- Neumayer, Robert, and Andreas Rauber. 2008. "Multimodal Analysis of Text and Audio Features for Music Information Retrieval." In *Multimodal Processing and Interaction*, edited by Petros Maragos, Alexandros Potamianos, Patrick Gros, and Borko Furht, 33:1–17. Multimedia Systems and Applications Series. Springer US.
- Newton-Dunn, Henry, Hiroaki Nakano, and James Gibson. 2003. "Block Jam: a Tangible Interface for Interactive Music." *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*.
- Nigay, Laurence, and Joëlle Coutaz. 1993. "A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion." In , 172–178. ACM Press.
- Norman, Donald. 1988. *The Psychology Of Everyday Things*. Basic Books.
- O'Sullivan, Dan, and Tom Igoe. 2004. *Physical Computing: Sensing and Controlling the Physical World with Computers*. 1st ed. Thomson.
- Orio, Nicola. 2006. "Music Retrieval: A Tutorial and Review." *Foundations and Trends in Information Retrieval* 1 (1): 1–96.
- Overholt, Dan. 2011. "The Overtone Fiddle: An Actuated Acoustic Instrument." In *Proceedings of the International Conference on New Interfaces for Musical Expression*.

- Oviatt, S. L. 2000. "Multimodal Interface Research: A Science Without Borders." In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'2000)*, edited by B. Yuan Huang & X. Tang, 3:1–6. Beijing.
- Paradiso, J. A. 1999. "The Brain Opera Technology: New Instruments and Gestural Sensors for Musical Interaction and Performance." *Journal of New Music Research* 28: 130–149.
- Patten, James, Ben Recht, and Hiroshi Ishii. 2002. "Audiopad: a Tag-based Interface for Musical Performance." *Proceedings of the 2002 Conference on New Interfaces for Musical Expression*.
- Paulus, J. K., and A. P. Klapuri. 2003. "Conventional and periodic N-grams in the transcription of drum sequences." In *2003 International Conference on Multimedia and Expo, 2003. ICME '03. Proceedings*, 2:II– 737–40 vol.2. IEEE.
- Peiper, Chad, David Warden, and Guy Garnett. 2003. "An Interface for Real-time Classification of Articulations Produced by Violin Bowing." In , 192–196. National University of Singapore.
- Perceptual Science Lab, UC Santa Cruz. 2012. "BA+GA=DA". University Research Laboratory. *Perceptual Science Lab, University of California at Santa Cruz*. <http://mambo.ucsc.edu/psl/dwmdir/da.html>.
- Percival, Graham, and Andrew Schloss. 2008. "Computer-Assisted Musical Instrument Tutoring with Targeted Exercises". University of Victoria, Masters Thesis Interdisciplinary Studies (Computer Science and Music).
- Petersen, K., J. Solis, K. Taniguchi, T. Ninomiya, T. Yamamoto, and A. Takanishi. 2008. "Development of the Waseda Flutist Robot No. 4 Refined IV: Implementation of a Real-time Interaction System with Human Partners." In *IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. Scottsdale, AZ.
- Raisamo, R. 1999. "Multimodal Human-Computer Interaction: a Constructive and Empirical Study". University of Tampere,.
- Ramirez, R., E. Maestre, A. Pertusa, E. Gomez, and X. Serra. 2007. "Performance-Based Interpreter Identification in Saxophone Audio Recordings." *IEEE Transactions on Circuits and Systems for Video Technology* 17 (3): 356–364.
- Ramirez, R., A. Perez, S. Kersten, and E. Maestre. 2008. "Performer Identification in Celtic Violin Recordings." In *Proceedings of the International Society on Music Information Retrieval Conference*.
- Raphael, Christopher. 2010. "Music Plus One and Machine Learning": 21–28.
- Rasamimanana, Nicolas, Emmanuel Flety, and Frederic Bevilacqua. 2006. "Gesture Analysis of Violin Bow Strokes." *Lecture Notes in Computer Science*. 145–155.
- Rasamimanana, Nicolas, Emmanuel Fléty, and Frédéric Bevilacqua. 2006. "Gesture in Human-Computer Interaction and Simulation." In , 3881:145–155. *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg.
- Rowe, Robert. 2001. *Machine Musicianship*. Cambridge, MA: MIT Press.
- Russolo, Luigi. 1986. *The Art of Noises (L'arte Dei Rumori)*. Trans. Barclay Brown. Pendragon Press.
- Scheirer, Eric. 1998. "Tempo and Beat Analysis of Acoustic Musical Signals." *The Journal of the Acoustical Society of America* 103 (1): 588–601.

- Schomaker, L., J. Nijtmans, A. Camurri, F. Lavagetto, P. Morasso, C. Benoit, T. Guiard-marigny, et al. 1995. *A Taxonomy of Multimodal Interaction in the Human Information Processing System: Report of the Esprit Project 8579 MLAMI*.
- Seyerlehner, Klaus, and Markus Schedl. 2009. "Block-Level Audio Features for Music Genre Classification." *Proceedings of the International Society on Music Information Retrieval Conference*: 151–158.
- Somerville, Peter, and Ra L. Uitdenbogerd. 2007. "Note-Based Segmentation and Hierarchy in the Classification of Digital Musical Instruments." In *Proceedings of the International Computer Music Conference*, 240–247.
- Stamatatos, Efstathios. 2001. "A Computational Model for Discriminating Music Performers." In *Proceedings of the MOSART Workshop on Current Research Directions in Computer Music*, 65–69.
- Stamatatos, Efstathios, and Gerhard Widmer. 2002. "Music Performer Recognition Using an Ensemble of Simple Classifiers." In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI 02)*.
- Stamatatos, Efstathios. 2002. "Quantifying the Differences Between Music Performers: Score Vs. Norm." In *Proceedings of the International Computer Music Conference (ICMC)*.
- Stamatatos, Efstathios, and Widmer, Gerhard. 2005. "Automatic Identification of Music Performers with Learning Ensembles." *Artificial Intelligence* 165 (1): 37–56.
- Tanaka, Atau. 2010. "Mapping Out Instruments, Affordances, and Mobiles." In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*. Sydney, Australia.
- Tanaka, Atau, and R. Benjamin Knapp. 2002. "Multimodal Interaction in Music Using the Electromyogram and Relative Position Sensing." *Proceedings of the 2002 Conference on New Interfaces for Musical Expression*.
- Tindale, Adam. 2007. "A Hybrid Method for Extended Percussive Gesture." In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, 392–393. NIME '07. New York, NY, USA: ACM.
- Tindale, Adam, Ajay Kapur, and George Tzanetakis. 2011. "Training Surrogate Sensors in Musical Gesture Acquisition Systems." *IEEE Transactions on Multimedia* 13 (1) (February): 50–59.
- Tindale, Adam, Ajay Kapur, George Tzanetakis, Peter Driessen, and Andrew Schloss. 2005. "A Comparison of Sensor Strategies for Capturing Percussive Gestures." In *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, 200–203. NIME '05. Singapore, Singapore: National University of Singapore.
- Tindale, Adam, Ajay Kapur, George Tzanetakis, and Ichiro Fujinaga. 2004. "Retrieval of Percussion Gestures Using Timbre Classification Techniques." In *Proceedings of the International Society on Music Information Retrieval Conference*.
- Toh, Chee, Bingjun Zhang, and Ye Wang. 2008. "Multiple-Feature Fusion Based Onset Detection for Solo Singing Voice." In *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR'08)*.
- Trueman, D., and P. Cook. 2000. "BoSSA: The Deconstructed Violin Reconstructed." *Journal of New Music Research* 29: 121–130.

- Tzanetakis, G., and P. Cook. 2002. "Musical Genre Classification of Audio Signals." *Speech and Audio Processing, IEEE Transactions On* 10 (5) (July): 293 – 302.
- Tzanetakis, G., Ajay Kapur, and Richard McWalter. 2005. "Subband-based Drum Transcription for Audio Signals." In *IEEE 7th Workshop on Multimedia Signal Processing*, 1–4.
- Tzanetakis, George, and Perry Cook. 1999. "MARSYAS: a Framework for Audio Analysis." *Organized Sound* 4 (3): 169–175.
- Waisvisz, Michel. 1985. "The Hands." In *Proceedings of the International Computer Music Conference.*, 313–318.
- Wang, Ge. 2008. "The Chuck Audio Programming Language: A Strongly-timed and On-the-fly Environ/mentality". Princeton University.
- Weinberg, Gil, and Scott Driscoll. 2006. "Robot-human Interaction with an Anthropomorphic Percussionist." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1229–1232. CHI '06. New York, NY, USA: ACM.
- Wessel, David, and Matthew Wright. 2002. "Problems and Prospects for Intimate Musical Control of Computers." *Computer Music Journal* 26 (3) (September): 11–22.
- Widmer, Gerhard. 2001. "Using AI and Machine Learning to Study Expressive Music Performance: Project Survey and First Report." *AI Communications* 14.
- Wiley, Meason, and Ajay Kapur. 2009. "Multi-Laser Gestural Interface - Solutions for Cost-Effective and Open Source Controllers." In *Proceedings of the International Conference on New Interfaces for Musical Expression*.
- Witten, Ian H., Eibe Frank, and Mark A. Hall. 2011. *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*. 3rd ed. Morgan Kaufmann.
- Wold, E., T. Blum, D. Keislar, and J. Wheaton. 1996. "Content-based Classification, Search, and Retrieval of Audio." *IEEE Multimedia* 3 (3): 27–36.
- Wright, Matthew. 2008. "The Shape of an Instant: Measuring and Modeling Perceptual Attack Time with Probability Density Functions" (March): 202.
- Wright, Matthew, Adrian Freed, and Ali Momeni. 2003. "Open Sound Control: State of the Art 2003." Proceedings of the 2003 Conference on New Interfaces for Musical Expression.
- Yoshii, Kazuyoshi, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. 2006. "Hybrid Collaborative and Content-based Music Recommendation Using Probabilistic Model with Latent User Preferences." In *Proceedings of the International Society on Music Information Retrieval Conference*, 296–301.
- Young, Diana. 2002. "The Hyperbow Controller: Real-time Dynamics Measurement of Violin Performance." Proceedings of the 2002 Conference on New Interfaces for Musical Expression.
- Young, Diana. 2007. "A Methodology for Investigation of Bowed String Performance Through Measurement of Violin Bowing Technique". Massachusetts Institute of Technology (MIT), PhD Thesis, MIT Media Lab.

- Zhen, Chao, and Jieping Xu. 2010. "Multi-modal Music Genre Classification Approach." In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference On*, 8:398–402.