School of Electronic Engineering and Computer Science
Queen Mary University of London

Scattering Transform for Playing Technique Recognition

Changhong Wang

PhD thesis

Submitted in partial fulfilment of the requirements for the Degree of
Doctor of Philosophy in Electronic Engineering
October 2021

Statement of Originality

I, Changhong Wang, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged and my contribution indicated. Previously published material is also acknowledged herein.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Signature: Changhong Wang

Date: 13/10/2021

Abstract

Playing techniques are expressive elements in music performances that carry important information about music expressivity and interpretation. When displaying playing techniques in the time–frequency domain, we observe that each has a distinctive spectro-temporal pattern. Based on the patterns of regularity, we group commonly-used playing techniques into two families: pitch modulation-based techniques (PMTs) and pitch evolution-based techniques (PETs). The former are periodic modulations that elaborate on stable pitches, including vibrato, tremolo, trill, and flutter-tongue; while the latter contain monotonic pitch changes, such as acciaccatura, portamento, and glissando.

In this thesis, we present a general framework based on the scattering transform for playing technique recognition. We propose two variants of the scattering transform, the adaptive scattering and the direction-invariant joint scattering. The former provides highly-compact representations that are invariant to pitch transpositions for representing PMTs. The latter captures the spectro-temporal patterns exhibited by PETs. Using the proposed scattering representations as input, our recognition system achieves start-of-the-art results. We provide a formal interpretation of the role of each scattering component confirmed by explanatory visualisations.

Whereas previously published datasets for playing technique analysis focused primarily on techniques recorded in isolation, we publicly release a new dataset to evaluate the proposed framework. The dataset, named CBFdataset, is the first dataset on the Chinese bamboo flute (CBF), containing full-length CBF performances and expert annotations of playing techniques. To provide evidence on the generalisability of the proposed framework, we test it over three additional datasets with a variety of playing techniques. Finally, to explore the applicability of the proposed scattering representations to general audio classification problems, we introduce two additional applications: one applies the adaptive scattering for identifying performers in polyphonic orchestral music and the other uses the joint scattering for detecting and classifying chick calls.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my primary supervisor, Emmanouil Benetos, for his dedicated support and guidance throughout my doctoral studies. The many conversations with him shaped the ideas of this thesis and helped me become an independent researcher. I would also like to thank my second supervisor Elaine Chew, who introduced me to the amazing field of music signal analysis. Without her acceptance of me transferring from a very different field, all these would not happen. Thank you also to Simon Dixon for the constructive feedback on my work at different stages.

A special thanks to Vincent Lostanlen, whom I have collaborated with during most of my PhD research. The discussions with him helped me go deeper and deeper on the topic of scattering transform. Many thanks also to Elisabetta Versace, Yudong Zhao, György Fazekas, Mark Sandler, Shuge Wang, and Xiaojie Meng for the joint work and publications; and to all the musicians for the help on data collection.

I sincerely thank my examiners, Juan Pablo Bello, Jason Hockman, and Joakim Andén, for their insightful comments and extremely useful suggestions on my thesis. I was fortunate to have Meinard Müller and Alexander Lerch as my WiMIR mentors and will never forget the early morning chats with Meinard in 2019.

There are so many people from the C4DM family that I would like to say thank you to: to William Wilkinson and Marco Martínez Ramírez for sharing thoughts on a wide range of topics; to Dan Stowell, Daniel Stoller, and Veronica Morfi for the inspiring discussions; to Giulio Moro, Jiajie Dai, Beici Liang, and Simin Yang for the suggestions at the beginning of my research; to Lele Liu for the nice evenings playing music together; and to many many others.

I would also like to thank the China Scholarship Council and Queen Mary University of London for the scholarship that makes all these possible.

Finally, my most heartfelt gratitude goes to my family, for their love and support, and for always encouraging me to pursue my dreams.

Contents

Abstract **3**

Acknowledgements **4**

1 Introduction **14**

1.1 Motivation and Aim 14

1.2 Contributions 16

1.3 Thesis Outline 17

1.4 Associated Publications 18

2 Background **21**

2.1 Music Performance Analysis 21

2.2 Computational Research on Playing Techniques 22

2.2.1 Methodologies 23

2.2.2 Datasets 26

2.2.3 Evaluation metrics 29

2.3 Chinese Bamboo Flute Music and Playing Techniques . . 30

2.4 Time–Frequency Representations for Music Signal Analysis 32

2.4.1 STFT spectrogram, CQT spectrogram, mel-frequency spectrogram 33

2.4.2 Wavelet transform 34

2.5 Scattering Transform 36

2.5.1 Time scattering 38

2.5.2 Time–Frequency scattering 39

3 CBFdataset: a Dataset of Chinese Bamboo Flute Performances **43**

3.1 Isolated versus Performed Playing Techniques 43

3.2 Playing Technique Grouping 45

3.3 CBFdataset Collection and Annotation 46

3.4 CBFdataset Content and Statistics 47

3.5 Conclusions 49

4 Adaptive Scattering for Pitch Modulation-based Technique Recognition **51**

4.1 Characteristics of Pitch Modulation-based Techniques . . 52

4.2 Adaptive Scattering 52

4.2.1 Adaptive scattering representations 53
4.2.2 Decomposition trajectories 58
4.3 Playing Technique Recognition 60
4.3.1 Feature extraction 61
4.3.2 Recognition system 63
4.4 Evaluation 64
4.4.1 Dataset 64
4.4.2 Metrics 64
4.4.3 Baseline 65
4.4.4 Comparative studies 65
4.4.5 Results 68
4.5 Conclusions 76

5 Joint Scattering for Pitch Evolution-based Technique Recognition 79

5.1 Characteristics of Pitch Evolution-based Techniques . . . 80
5.2 Direction-invariant Joint Time–Frequency Scattering . . 80
5.3 Playing Technique Recognition 85
5.3.1 Feature extraction 86
5.3.2 Recognition system 88
5.4 Baselines 89
5.4.1 Baselines for binary classification scheme 89
5.4.2 Baselines for multiclass classification scheme . . . 97
5.5 Evaluation 97
5.5.1 Binary classification scheme 98
5.5.2 Multiclass classification scheme 101
5.6 Additional Evaluations 107
5.7 Conclusions 112

6 Scattering Transform Applications 115

6.1 Adaptive and Standard Time Scattering for Performer Identification 115
6.1.1 Background 117
6.1.2 Dataset 118
6.1.3 Vibrato detection 118
6.1.4 Performer identification 124
6.1.5 Discussion and conclusions 130
6.2 Joint Scattering for Chick Call Recognition 132
6.2.1 Background 132
6.2.2 Dataset 133
6.2.3 Chick call detection 135
6.2.4 Chick call recognition 137
6.2.5 Discussion and conclusions 147

7 Conclusions and Future Work 149

7.1 Summary of Contributions 149
7.2 Discussion 151

7.3 Future Work 153

Bibliography **155**

List of Figures

2.1 An example of a Chinese bamboo flute 31

2.2 Diagram of the scattering transform 37

2.3 Relationship between different scattering operators . . . 37

3.1 Comparison of isolated and performed glissandi 44

3.2 Spectrograms of commonly used musical playing techniques 46

3.3 Playing technique statistics in the CBFdataset 49

4.1 Visual comparison of PMT characteristics 53

4.2 Example of adaptive scattering representation extraction 56

4.3 Comparison of three adaptive scattering decomposition trajectories (original) 59

4.4 Comparison of three adaptive scattering decomposition trajectories (preprocessed) 59

4.5 Flutter-tongue detection result for an example excerpt . 74

4.6 Effect of SVM hyperparameters during training 76

5.1 Calculating the joint time–frequency scattering 83

5.2 Visualisation of the joint time–frequency scattering for PETs 84

5.3 Diagram of rule-based segmentation for glissando detection baseline 91

5.4 Long-term feature calculation for an upward glissando . . 92

5.5 Diagram of HMM-based glissando detection baseline . . . 93

5.6 Portamento detection result for an example excerpt . . . 100

5.7 Playing technique recognition results on CBFdataset and additional datasets 102

5.8 Confusion matrices of CBF playing technique recognition using the proposed scattering representations 104

5.9 Confusion matrices of CBF playing technique recognition using MFCCs . . . 105
5.10 Confusion matrices of CBF playing technique recognition using MPS . . . 106
5.11 Confusion matrices of playing technique recognition in the SOL dataset. . . . 110
5.12 Confusion matrices of playing technique recognition in the VocalSet . . . 111

6.1 Distribution of annotated vibratos in the Violin-II dataset 119
6.2 Example vibrato detection result using dominant band trajectory . . . 121
6.3 Example vibrato detection result using predominant melody trajectory . . . 121
6.4 Example vibrato detection result using fundamental frequency trajectory . . . 122
6.5 Confusion matrices of performer identification in the Violin-II dataset using the standard time scattering . . . 129
6.6 Schematic of laboratory conditions for chick sound collection 134
6.7 Visualisation of chick call onset detection and segmentation results . . . 136
6.8 Direction of chick call frequency changes captured by joint time–frequency scattering . . . 138

7.1 Example of a trill modelling using adaptive scattering . . 154

List of Tables

2.1 Summary of existing computational research on playing techniques. 24

2.2 Summary of available datasets with playing techniques 28

4.1 Characteristics of pitch modulation-based techniques (PMTs). 52

4.2 Hyperparameters of the proposed adaptive scattering representations 62

4.3 Frame sizes and dimensionalities of the adaptive scattering representations for PMTs 63

4.4 Frame sizes and dimensionalities of the adaptive scattering and existing scattering representations for PMTs 68

4.5 Frame-based performance comparison of different decomposition trajectories for PMT recognition 70

4.6 Event-based performance comparison of different decomposition trajectories for PMT recognition 70

4.7 Frame-based performance comparison of different methods for PMT recognition 72

4.8 Event-based performance comparison of different methods for PMT recognition 72

4.9 Computational cost comparison of different scattering representations 75

5.1 Characteristics of pitch evolution-based techniques (PETs) 81

5.2 Hyperparameters of the proposed direction-invariant joint time–frequency scattering 86

5.3 Dimensionalities and frame sizes of the proposed and standard joint-time frequency scattering for PETs 87

5.4 Evaluation results of the F0-informed glissando detection system 95

5.5 Pitch estimation accuracy for full-length pieces and glissando segments in the CBF-glissDB. 95
5.6 Evaluation results of the fully-automated glissando detection system 96
5.7 Frame-based performance comparison of different methods on PET recognition 99
5.8 Event-based performance comparison of different methods on PET recognition 99
5.9 Comparison of different scattering representations for PETs in terms of runtime 101

6.1 Composer and name of each concerto in the Violin dataset. 118
6.2 Results of vibrato detection in Violin-II dataset 125
6.3 Results of performer identification in the Violin-II dataset 128
6.4 Onset detection and segmentation results in the chick call dataset 136
6.5 Number of pleasure, contact, and uncertain calls in the chick call dataset 138
6.6 Frame-based results of chick call recognition using the proposed and the baseline methods 144
6.7 Event-based results of chick call recognition using the proposed and the baseline methods 145

List of Abbreviations

AMT	Automatic music transcription
AdaTS	Adaptive time scattering
AdaTRS	Adaptive time–rate scattering
CBF	Chinese bamboo flute
CBFdataset	A dataset of Chinese bamboo flute performances
CNNs	Convolutional neural networks
CQT	Constant-Q transform
dJTFS	Direction-invariant joint time–frequency scattering
FDM	Filter diagonalisation method
HLF	High-level features
HMMs	Hiden Markov models
JTFS	Joint time–frequency scattering
MFCC-Only	Mel-frequency cepstral coefficients only
MIR	Music information retrieval
MPA	Music performance analysis
PCA	Principal component analysis
PETs	Pitch evolution-based techniques
PMTs	Pitch modulation-based techniques
Scat-Only	Scattering features only
SED	Sound event detection
Seg-HLF	Segmentation and high-level features
Seg-MFCC	Segmentation and mel-frequency cepstral coefficients
Seg-Scat	Segmentation and scattering features
SOL	Studio On Line
StaTS	Standard time scattering
SVMs	Support vector machines
VF-DS	Vibrato feature distribution similarity
VocalSet	A signing voice dataset
VPset	Vibrato/portamento dataset

Chapter 1

Introduction

This thesis presents a computational framework based on the scattering transform for the automatic recognition of playing techniques in music signals. In this chapter, we provide the motivations and aim of this work in Section 1.1. Section 1.2 and Section 1.3 list the contributions and outline of the thesis, respectively. Associated publications with this thesis are provided in Section 1.4.

1.1 Motivation and Aim

Playing techniques in music signals, such as vibratos and tremolos in instrument playing or in singing voice, contain important information of the music style and the performers' interpretation. The modeling and automatic recognition of playing techniques may benefit research in automatic transcription of musical ornaments (Gainza and Coyle, 2007), realistic music generation (Oord et al., 2016), computer-aided music pedagogy (Han and Lee, 2014), instrument classification (Hall et al., 2013; Lostanlen et al., 2018), and performance analysis (Yang, 2017).

The motivation of this work comes form three aspects: the scarcity of data for playing technique analysis in the literature, the possibility of developing a general-purpose and explainable audio representation for playing technique recognition, and the benefits that many applications in music signal analysis (such as those listed above) may gain. Current computational research on playing techniques suffers from a scarcity of real-world performances with playing technique annotations. Playing techniques are rare events in full-length performances and annotating them requires expert knowledge of both the music and the playing techniques. Datasets employed in existing research consist of mainly playing techniques recorded in isolation. Isolated playing techniques can vary greatly from the same techniques used in live performances.

Existing research efforts on playing technique recognition either developed specific methods for specific playing technique(s) or intended to recognise multiple types of playing techniques based on a large set of audio features (Su et al., 2014b) or using data-driven methods (Wilkins et al., 2018). However, playing techniques are music events independent of performer, pitch, genre, instrument, and region which a playing technique recognition system should be invariant to. Feeding with high-dimensional features is a possible but not optimal solution since such an approach would not explain what information of a playing technique is captured.

With limited data available, one may seek for a compact representation that reduces the variabilities irrelevant to the task at hand. These variabilities of playing techniques indicated by the characteristics mentioned above can be interpreted as the invariance of the targeted representation to time-shifts, time-warps, and frequency-transpositions. Take the vibrato technique as an example: a short translation or a small dilation of the audio signal, or a different pitch on which the technique is played should not change its class identity. Based on an investigation of existing time–frequency representations for audio, we find that the scattering transform, a flexible framework for building invariant, stable, and informative signal representations (Mallat, 2012), fills the gaps with mathematical guarantees. Exploring representations for playing techniques from an invariance property perspective in general also enables the applicability of this research to audio signals in other domains with similar patterns.

The aim of the thesis is to develop a general-purpose audio representation for playing technique recognition in music signals. To this end, we analyse the spectro-temporal patterns of two families of playing techniques and propose two variants of the scattering transform correspondingly, to encode the characteristic information of each family of playing techniques. Based on the proposed representations, we build recognition systems which take audio signals as input, calculate the representations, feed them into a machine learning classifier, and output playing technique events with a temporal location and label. For ecologically valid analysis of playing techniques in context, we create the first dataset on the Chinese bamboo flute with full-length performances and expert playing technique annotations, to evaluate our proposed methodology; and make the dataset publicly available to the community. To evaluate the generalisability of the proposed methodology, we verify the system on different datasets with a variety of instrumental and vocal techniques. We also test the applicability of the proposed methodology to two classification problems beyond playing technique recognition and music signal analysis: one identifies violin performers in polyphonic orchestral music and the other detects and classifies chick calls.

1.2 Contributions

We summarise the main contributions of each chapter as follows.

Chapter 2

- A through literature review of existing computational research on playing techniques.

Chapter 3

- The first dataset on the Chinese bamboo flute, the CBFdataset, which is publicly released for computational research on playing techniques recorded in context.

Chapter 4

- A variant of the scattering transform framework, the adaptive scattering, which provides representations for pitch modulation-based techniques (PMTs), a group of periodic modulations elaborated on stable pitches.
- An automatic system for detecting and classifying PMTs.
- A formal interpretation of the role of each component in the scattering feature extractor, confirmed with explanatory visualisations.

Chapter 5

- A variant of the joint time–frequency scattering, the direction-invariant joint time–frequency scattering, for pitch evolution-based techniques (PETs), a group of playing techniques exhibiting monotonic pitch changes.
- An automatic system for detecting and classifying PETs.
- A baseline method for detecting glissandi in real-world music performances.
- Generalisability evaluation of the proposed recognition system on three additional datasets with a variety of instrumental and vocal techniques.

Chapter 6

- Application of the proposed adaptive time scattering and the standard time scattering for performer identification in polyphonic orchestral music.
- Application of the joint time–frequency scattering representation for detecting and classifying chick calls.

1.3 Thesis Outline

The rest of the thesis is organised as follows.

Chapter 2 provides background information related to music performance analysis, an overview of existing computational research on playing techniques, an introduction of Chinese bamboo flute music and Chinese bamboo flute playing techniques, and a brief description of well-known time–frequency representations for audio signals, followed by an overview of the scattering transform.

Chapter 3 first presents the difference between isolated and performed playing techniques with examples, followed by a grouping of commonly used playing techniques based on their spectro-temporal patterns. Finally, it introduces the collection and annotation process, and the content of the CBFdataset we created and publicly released.

Chapter 4 proposes a variant of the scattering transform framework, the adaptive scattering, for representing pitch modulation-based techniques (PMTs). A recognition system is then built and evaluated on the CBFdataset, with explanatory visualisations.

Chapter 5 modifies the joint time–frequency scattering into a direction-invariant representation for pitch evolution-based techniques (PETs). An automatic system is then proposed as a baseline method for detecting glissando, a type of PETs. We build and evaluate two recognition systems in this chapter, one for detecting PETs only and the other for recognising both PMTs and PETs simultaneously. To provide further evidence, we apply the proposed recognition systems on three additional datasets with a variety of playing techniques.

Chapter 6 tests the applicability of the proposed framework on two more applications beyond playing technique recognition and music signal analysis. One identifies violin performers in polyphonic orchestral music using vibratos detected by our proposed playing technique recognition system; the other detects and classifies chick calls in that the chick calls exhibit spectro-temporal patterns similar to certain playing techniques.

Chapter 7 concludes the thesis, with a summary of the contributions, discussions on the strengths and weaknesses, and possible directions for future work.

1.4 Associated Publications

(i) Changhong Wang, Emmanouil Benetos, Vincent Lostanlen, and Elaine Chew. **Adaptive Scattering Transforms for Playing Technique Recognition**, submitted to *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP).*

This is a journal paper that combines the core content of Chapter 4 and Chapter 5 and that extends the content of Publications (ii) and (iii). It develops a general framework for playing technique recognition in musical signals, evaluated on the CBFdataset, and with the generalisability verified over three additional datasets with a variety of playing techniques.

CW contributed the majority of work towards this publication. EB provided important suggestions on the extended content, such as the selection of additional evaluation datasets, experimental design, comparison across the datasets, and the evaluation metrics. VL gave useful feedback on the theoretical development of the proposed adaptive scattering transforms. EC contributed to technical discussions and writing.

(ii) Changhong Wang, Vincent Lostanlen, Emmanouil Benetos, and Elaine Chew. **Playing Technique Recognition by Joint Time-Frequency Scattering**. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 881–885.

Chapter 5 is an extension of this peer-reviewed conference paper, evaluating the system on an enlarged dataset, adding a baseline method for glissando detection, and testing the generalisability of the proposed recognition system on three additional datasets.

CW proposed the main idea, designed the experiments, and wrote the majority of the paper. VL provided inspiring insights regarding the directionality of the joint time–frequency scattering. EB contributed to theoretical discussions and writing while EC offered suggestions from a music perspective.

(iii) Changhong Wang, Emmanouil Benetos, Vincent Lostanlen, and Elaine Chew. **Adaptive Time–Frequency Scattering for Periodic Modulation Recognition in Music Signals**, *International Society for Music Information Retrieval Conference (ISMIR)*, Nov 2019, pp. 809–815.

This peer-reviewed conference paper forms the basis of Chapter 4. The latter improves the content of the paper by rerunning experiments with an enlarged dataset and extends the content with a mathematical definition of the proposed adaptive scattering representations, and an event-based evaluation metric.

CW proposed the main idea and did the majority of the witting for this publication, with feedback from EB on the theoretical development and experimental design. VL and EC contributed to technical discussions and witting.

(iv) Changhong Wang, Emmanouil Benetos, Xiaojie Meng, and Elaine Chew. **HMM-based Glissando Detection for Recordings of Chinese Bamboo Flute**, *Sound and Music Computing Conference (SMC)*, May 2019, pp. 545–550.

This is the first published paper in the timeline towards this thesis, of which the main idea goes into Section 5.4. Theoretical development was shared by CW and EB. CW ran the experiments and wrote the majority of the paper with technical feedback from EB. Both EB and EC provided useful guidance on the paper writing. Contributions to the data collection and annotation were shared by all the authors.

(v) Yudong Zhao, Changhong Wang, György Fazekas, Emmanouil Benetos, and Mark Sandler. **Violinist Identification Based on Vibrato Features**, *European Signal processing conference (EUSIPCO)*, Aug 2021.

Section 6.1 is inspired from the above paper, although the content of the above section is different from the paper. This work developed two systems based on vibrato features of manually annotated vibrato notes for violinist identification. CW contributed to one of the two systems on the experimental design, evaluation, and writing.

(vi) Changhong Wang, Emmanouil Benetos, Shuge Wang, and Elisabetta Versace. **Joint Scattering for Automatic Chick Call Recognition**, submitted to *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022).*

Section 6.2 is an extension of the above paper, with more chicks included in the evaluation of the detection system and one more scheme considered in the classification system. The idea came from the discussions between CW with EB. CW run all the experiments and write the majority of the paper, with feedback from EB and EV. SW contributed to the data collection and annotation.

Chapter 2

Background

This chapter provides the background and theoretical basis that later chapters of the thesis are built upon. We review existing computational research on music performance analysis and on playing techniques in Section 2.1 and Section 2.2, respectively. Section 2.3 introduces the music and playing techniques of the Chinese bamboo flute. Section 2.4 presents time–frequency audio representations for music signal analysis; finally, Section 2.5 presents an overview of the scattering transform.

2.1 Music Performance Analysis

Music is a performing art. A music idea initialised by the composer in the form of a music score or other representations may be interpreted by the performers in different ways. Music performance analysis (MPA) can be approached from different perspectives: ethnomusicological (Cook, 2014), musicological (Gabrielsson, 1999), and computational (Chew, 2000) amongst other perspectives. Computational research on MPA aims at obtaining a basic understanding of music performances in a quantitative and empirical way (Lerch et al., 2020), which is the scope of this thesis.

Despite the ubiquity of performed music, the analysis on how this music is created still lags behind other areas in music information retrieval (MIR), due to the lack of data, resolution of the problem, and indefinite boundaries of the definitions. MIR researchers focus more on score-like information and metadata than performance information (Lerch et al., 2020), while the latter is the focus of MPA. The nature of performance information is often subtle and less well-defined as compared to score-like information. Additionally, there are a large variety of parameters that influence a music performance and some of them are beyond the recorded audio signal, such as gesture parameters.

In the literature, performance parameters proposed for the measurement of music performances include tempo, timing, dynamics, intonation, and articulation (Cancino-Chacón et al., 2018). The majority of the work intends to identify general trends of a music performance or to compare performances between different interpretations of the same composition. For example, Srinivasamurthy et al. (2017) studied the tempo and rhythmic elaboration in Hindustani music; Liem et al. (2011) proposed an approach to analyse expressive timing between multiple recordings of the same composition; Cancino-Chacón et al. (2017) evaluated linear and non-linear models of expressive dynamics in classical piano music; and Devaney et al. (2012) compared intonation characteristics between professional and non-professional singers.

As compared to the above mentioned parameters, playing techniques in a performance context are less-explored, although a complete mastery of the instrumental techniques is one of the major components of excellence in music performance (Gabrielsson, 1999). Additionally, playing techniques are one of the most difficult skills to acquire in instrument learning (Menzies and McPherson, 2015), which require considerable time and effort. This is especially the case for folk instruments, where the key points of the techniques are normally delivered to students by live demonstration and oral instructions (Zhang, 2011). Even with these instructions, it may still take a long time for students to manage a playing technique and to flexibly apply it into performances. Computational research on playing techniques can provide technical support for developing pedagogy tools that may greatly help with this learning process. This is one of the possible benefits that this research can bring with and we will discuss other potential applications in Chapter 7. We first review existing computational research on playing techniques in Section 2.2.

2.2 Computational Research on Playing Techniques

Due to the annotation-intensive nature and scarcity of playing techniques in real-world performances, previous computational research on playing techniques was typically instrument- or technique-specific, or focused on playing techniques recorded in highly controlled environments. We summarise the existing research in Table 2.1 with a complete list of playing techniques analysed and the corresponding methodologies applied according to instrument type.

As can be seen, prior research has focused mainly on Western instruments. Guitar playing techniques are the most frequently explored ones as compared to other instruments (Giraldo and Ramírez, 2015; Ozaslan and Arcos, 2010; Reboursière et al., 2012; Su et al., 2014b; Chen et al., 2015; Abeßer et al., 2010; Abeßer and Schuller, 2017). These techniques are commonly categorised by the active hand, which leads to the categorisation of expression-style (left-hand) and plucking-style (right-hand) techniques. Piano technique recognition only includes trills (Brown and Smaragdis, 2004) and pedalling techniques (Liang et al., 2017). Playing technique analysis on other Western instruments covers violin (Charles, 2010; Su et al., 2014a), drums (Wu and Lerch, 2016; Herrera et al., 2002), cello (Ducher and Esling, 2019), Irish flute (Jančovič et al., 2015) and highland pipe (Menzies and McPherson, 2015). Non-Western instruments include the erhu (Yang, 2017), guqin (Huang et al., 2020), ney (Özaslan et al., 2012), and Chinese bamboo flute (Ayers, 2003; 2004; 2005). Due to their similarity with instrumental playing techniques, vocal techniques (Neocleous et al., 2015; Wilkins et al., 2018) are also included in the table for completeness. In the following, we analyse the research work in Table 2.1 from three fronts: methodologies, datasets, and evaluation metrics.

2.2.1 Methodologies

Early research on playing technique recognition either focused on specific playing technique(s) to explicitly incorporate prior knowledge or fed a large set of features to machine learning classifiers. Liang et al. (2017) developed a system that classifies pedalling techniques, i.e., 1/4 pedal, 1/2 pedal, 3/4 pedal, and full pedal, in classical piano performances. The system used support vector machines (SVMs) as the classifier which takes as input the mean and standard deviation of the gesture data between detected onsets and offsets. The filter diagonalisation method (FDM), which efficiently extracts high resolution spectral information for short time signals, was applied to vibrato detection in erhu performance

Instrument	Playing techniques	Methodology	Cite
Guitar	Ornamentation defined by rules	Dynamic time warping sequence matching	Giraldo and Ramírez (2015)
	Legato, glissando	Symbolic, aggregate approximation	Ozaslan and Arcos (2010)
	Muted, harmonic, bend, slide, hammer-on, pull-off	Rule-based detection	Reboursière et al. (2012)
	Muting, vibrato, pull-off, hammer on, sliding, bending	Sparse cepstral and phase codes	Su et al. (2014b); Chen et al. (2015)
	Plucking: finger, picked, muted, slap; expression: harmonics, vibrato, bending, slide, etc.	Feature extraction, SVMs with radial basis function kernel	Abeßer et al. (2010); Abeßer and Schuller (2017)
Piano	Trills	Independent component analysis	Brown and Smaragdis (2004)
	Pdealling techniques	SVMs	Liang et al. (2017)
Violin	Beginners' faults on playing techniques	K-nearest neighbours	Charles (2010)
	Flageolet, non-vibrato, pizzicato, sordino, spiccato, sul ponticello, sul tasto, tremolo	Sparse modelling	Su et al. (2014a)
Drums	Strike, buzz roll, drag, flam	Non-negative matrix factorization-based activation	Wu and Lerch (2016)
	A variety of drum playing techniques	K-nearest neighbours, partial decision tree, etc.	Herrera et al. (2002)
Cello	Vibrato, tremolo, trill, glissando, etc.	Folded convolutional neural network	Ducher and Esling (2019)
Irish flute	Cut, strike, crann, roll, shake	Rule-based method; HMMs; neural network	Jančovič et al. (2015); Ali-MacLachlan (2019)
Highland pipe	60 ornamentations	Dynamic time warping	Menzies and McPherson (2015)
Ney	Vibrato, kaydirma	Rule-based characterisation	Özaslan et al. (2012)
Erhu	Vibrato, portamento	FDM, HMMs	Yang (2017)
Guqin	None, vibrato, protamento	Neural network	Huang et al. (2020)
Chinese bamboo flute	Trill, flutter-tongue, tremolo	Synthesising	Ayers (2003; 2004)
Other	Vibrato, glissando	Rule-based methods	Renato Panda and Paiva (2018)
Multi-instruments	140 instrumental playing techniques (isolated)	Scattering transform	Lostanlen et al. (2018)
Vocal techniques	Vibrato, glissando, linear note change	COSFIRE filter	Neocleous et al. (2015)
	17 different vocal techniques	Convolutional neural networks	Wilkins et al. (2018)

Table 2.1: Summary of existing computational research on playing techniques.

(Yang, 2017). It used the fundamental frequency (F0) estimated by the pYIN pitch detection algorithm (Mauch and Dixon, 2014) as input. Yang (2017) also detected erhu portamenti using Hidden Markov models (HMMs). Gainza et al. (2004) presented an automatic detection system for two single-note ornaments in traditional Irish flute playing, 'cut' and 'strike'. They first segmented the recordings based on the onset detection results using three methods: amplitude change of both temporal and spectral domains, and fundamental frequency. The segments below 90 ms are then classified as ornaments, and the specific type of the detected ornament depends on the fundamental frequency change. Menzies and McPherson (2015) recognised ornamentations of the great highland bagpipe from professional bagpipers' and students' performance recordings using dynamic time warping. Brown and Smaragdis (2004) improved the estimation performance of the pitch and timing of each note present in the piano trills based on independent component analysis.

Feeding a large set of features is a possible but not optimal way for detecting multiple types of playing techniques. Chen et al. (2015) detected electric guitar playing techniques based on F0 sequence pattern recognition. Using a set of timbre and pitch features, they classified five playing techniques: bend, vibrato, hammer-on, pull-off, and slide. Based on a systematic analysis of the bass playing techniques, Abeßer and Schuller (2017) selected a set of features and employed SVMs to classify both the plucking style techniques (finger, picked, muted, slap) and the expression style techniques (harmonics, vibrato, bending, slide). Su et al. (2014a) implemented sparse modeling for detecting violin playing techniques including flageolet, non-vibrato, pizzicato, sordino, spiccato, sul ponticello, sul tasto, and tremolo using a large set of temporal, spectral, cepstral, and harmonic features.

More recent research on playing technique recognition investigated data-driven methods. Ducher and Esling (2019) proposed a folded constant-Q transform (CQT) (see Subsection 2.4.1) representation, which was used as the input to a recurrent neural network for violin playing technique recognition. The evaluation data was synthetic audio sequences including randomly generated notes and chords with all possible violin playing techniques. Liang et al. (2019) applied convolutional neural network (CNNs) for detecting sustain-pedal playing techniques which were also trained on generated piano data. Encoding specifications for notes with different pedalling conditions in standard MIDI, a Yamaha Disklavier piano was used to playback the recordings. Ali-MacLachlan (2019) proposed a two-stage method for automatic detection of Irish flute ornamentations, which first detected note onsets from full-length recordings followed by classifying each inter-onset segment into one of the three classes: note, cut, and strike (see Section 5.3, Ali-MacLachlan, 2019). At the classification stage, a feedfoward neural network with 2 hidden layers and 20 neurons per each layer was used; and the input features were 13 mel-frequency cepstral coefficients, 12 chroma features, and the length of the inter-onset segments.

Huang et al. (2020) classified 6 types of guqin playing techniques with onset and offset annotations based on different levels of features extracted from the CQT, pitch salience function, and pitch contour. Wilkins et al. (2018) trained CNNs for classifying 10 types of vocal techniques recorded in long segments: straight, vibrato, belt, lip trill, breathy, vocal fry, trillo, inhaled, trill, and spoken. Wang et al. (2020b) compared different deep learning models for recognising the playing techniques of erhu and Chinese bamboo flute, where for the latter instrument the authors used the CBFdataset which is proposed in this thesis (see Section 3.4). Some of these methods might achieve good results but are not explainable on what information of a playing technique has been captured. Additionally, most of these methods above used generated data or isolated playing techniques, where the generalisability remains to be tested. To the author's knowledge, there is not yet a general framework for playing technique recognition in real-world performances that can be used to detect multiple types of playing techniques and that is possible to be generalised across different instruments and datasets.

2.2.2 Datasets

As discussed above, previously published datasets for playing technique analysis often focused on isolated playing techniques, recorded in highly controlled environments (Su et al., 2014a; Wilkins et al., 2018; Lostanlen et al., 2018). Although in the context of full-length performances, playing techniques exhibit considerable variations as compared to being played in isolation (see Section 3.1), existing datasets still provide valuable information for evaluating playing technique recognition systems or for other computational research of playing techniques. We list publicly available datasets in Table 2.2 for the community to conveniently access suitable datasets. Except the singing voice dataset (VocalSet) (Wilkins et al., 2018) and the vibrato/portamento dataset (VPset) (Yang, 2017), both including singing voice, the remaining datasets all comprise instrument playing. Regarding the number of instruments included, Contimbre, Studio On Line (SOL) dataset, and VPset (also includes instrument playing besides singing) are datasets containing multiple types of instruments, while the others (except VocalSet), as indicated by the dataset names, consist of a single instrument type for each dataset. Among these datasets, we discuss three in detail which we use as the *additional datasets* in Section 5.6 to evaluate the generalisability of our proposed methodology: VPset, SOL dataset, and VocalSet.

VPset Proposed in Yang (2017), the vibrato[1]/portamento[2] dataset includes two separate subsets. The vibrato subset comprises 4 full-length performances played on the Chinese instrument erhu and the Western instrument violin, 64 short excerpts of solo instrument playing, and vibrato annotations. The duration of the vibrato subset is 25 minutes. Besides having the same erhu and violin recordings as the vibrato subset, the portamento subset also includes recordings of Beijing opera singing and portamento annotations; the total audio duration of the portamento subset is 55 minutes. It is not applicable to concatenate the two subsets into one since there are no vibrato annotations for Beijing opera singing in the portamento subset, as there are no portamento annotations for solo instrument playing in the vibrato subset. For simplicity, we hereafter denote these two subsets as the VPset. When it comes specifically to vibrato detection, we will refer to the vibrato subset in Section 5.6; similarly for portamento detection.

SOL dataset Studio On Line[3] (version 0.9HL) (Lostanlen et al., 2018) is a multitype instrument dataset, comprising 12 categories of instruments playing isolated tones, with a total duration of 27.1 hours. It covers 140 types of playing techniques although some playing techniques have a small number of samples. To evaluate the generalisability of our proposed methodology for playing technique recognition (see Section 5.6),

[1] https://github.com/skx300/vibrato_dataset

[2] https://github.com/skx300/portamento_dataset

[3] https://forum.ircam.fr/projects/detil/orchids/

Dataset	Size	Content	Cite
Cello	14.1	18 playing techniques	Ducher and Esling (2019)
ConTimbre	270GB	Commercial, instrumental notes with playing techniques, 150 orchestral instruments, 4000 playing modes	Hummel (2014)
DrumPt	1.25	4 playing techniques, 2000 annotations (audio from ENST)	Wu and Lerch (2016)
ENST-Drums	1.25	318 segments, playing techniques	Gillet and Richard (2006)
Guitar playing techniques	7.2	6580 clips, 7 playing techniques	Chen et al. (2015)
IDMT-SMT-Bass	3.6	4300 notes, 10 playing techniques	Abeßer et al. (2010)
IDMT-SMT-Guitar	5.7	9 guitar playing techniques, 5100 notes	Kehling et al. (2014)
ITM-Flute-99 dataset	0.91	99 recordings, 15310 notes, 2244 cuts and 672 strikes.	Ali-MacLachlan (2019)
MDB Drums	0.36	Drum playing techniques	Southall et al. (2017)
SOL dataset	27.1	12 instruments, 140 instrumental playing techniques	Lostanlen et al. (2018)
VPset	1.3	Vibrato and portamento techniques on erhu	Yang (2017)
Violin Gestures Dataset	2.3	880 recordings, 5 bowing techniques	Sarasúa et al. (2017)
VocalSet	10.1	20 singers, 17 vocal techniques, 3560 clips	Wilkins et al. (2018)

Table 2.2: Summary of available datasets with playing techniques. Size of all datasets is in hours except the ConTimbre dataset of which the duration is not available.

we focus on commonly used playing techniques and consider only techniques with over 100 samples in the SOL dataset. Non-techniques like crescendos and decrescendos are beyond the scope of this thesis. The list of playing techniques can be found in Figure 5.7; the audio recordings with the considered data have a total duration of 9.8 hours. For the playing technique labels, we follow the original annotations except for five labels resulting from merging similar patterns: sul-tasto/ponticello, pizzicato, glissando, trill, and flatterzunge. For example, we merge the labels trill-major-second-up and trill-minor-second-up into one label trill. Note that glissando in the SOL dataset corresponds to portamento on the Chinese bamboo flute (see Section 3.2), both consisting of smooth pitch changes; and that flatterzunge here is equivalent to flutter-tongue on the Chinese bamboo flute.

VocalSet A singing voice dataset[4] (Wilkins et al., 2018). It has recordings of 10.1 hours of 20 professional singers (11 male, 9 female) performing 17 different vocal techniques. To make the results comparable to that obtained in Wilkins et al. (2018), we focus on the same ten techniques: straight, vibrato, belt, lip trill, breathy, vocal fry, trillo, inhaled, trill, and spoken, as shown in Section 5.6. The number of trill and spoken techniques in this dataset are below 100, with 95 and 20 examples, respectively.

2.2.3 Evaluation metrics

For the evaluation of playing technique recognition system outputs, two approaches can be found in the literature (Wilkins et al., 2018; Ducher and Esling, 2019; Yang, 2017): frame- and clip-based evaluation. Metrics for both methods include precision $\mathcal{P}$, recall $\mathcal{R}$, F-measure $\mathcal{F}$, and accuracy $\mathcal{A}$, defined as:

$$\mathcal{P} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \mathcal{R} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \mathcal{F} = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P} + \mathcal{R}}, \tag{2.1}$$

$$\mathcal{A} = \frac{\text{TP}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{2.2}$$

where $\text{TP}, \text{FP}, \text{FN}, \text{TN}$ are true positives, false positives, false negatives, and true negatives, respectively (Müller, 2015). Frame-based evaluation

[4] https://zenodo.org/record/1193957

compares the output labels with the ground truth in a frame wise manner. Since most existing datasets comprise playing techniques performed on single notes, it is common to evaluate the classification result on a clip level. This means that the recognition system outputs one label for each clip and compares it with the ground truth.

Not yet being applied for evaluating playing technique recognition results, event-based evaluation using these metrics is frequently considered for other audio signal processing problems, such as sound event detection (SED) (Mesaros et al., 2016) and automatic music transcription (AMT) (Benetos, 2012). The events in SED are general-purpose audio events, such as fire alarm and door opening, with associated onsets and offsets; while those in AMT are music notes with an onset, an offset, and a pitch value. Event-based evaluation compares a list of events output by the SED or the AMT system with the reference list of events.

There are mainly two ways of comparing the events: onset-only and onset-offset for both SED and AMT. For onset-only evaluation, an event is considered to be correctly detected when its onset falls within a window constraint of the ground truth (Mesaros et al., 2016); and a music note is regarded to be correctly transcribed when its onset falls within a window constraint of the ground truth and its pitch is within ± a quarter tone of the ground truth pitch (Benetos, 2012). Besides these rules, the onset-offset evaluation requires the offset of the event (or the music note) to be also within a window constraint of the ground truth offset or surround the ground truth offset within a percentage of the ground-truth's duration. The window constraint and percentage values depend on the targeted accuracy of the specific tasks. Since playing techniques are also music events with certain durations, we consider in this thesis also event-based metrics for playing technique recognition, in addition to the frame- and clip-based metrics.

2.3 Chinese Bamboo Flute Music and Playing Techniques

The Chinese bamboo flute (CBF), also known as the *dizi* (笛子) or *zhudi* (竹笛), is one of the most ancient instruments in the world. Made from bamboo, the CBF normally has 8 effective holes on the flute: a blowing hole, a membrane hole, and 6 finger holes, as shown in Figure 2.1 (a).

There are also 4 end (or auxiliary) holes, 2 in line with the finger holes and the other 2 on the opposite side of the flute, where the former benefit the air ventilation and the latter are normally used to attach a decoration cord. Different from the Western flute with open holes, the CBF is characterised by a hole covered with a thin membrane (see Figure 2.1 (b)) between the blowing hole and the sixth finger hole. The membrane is driven by the acoustic pressure in the resonator and radiates sounds when it vibrates. Wrinkles in the membrane are critical to the production of the characteristic CBF timbre (Tsai, 2003).

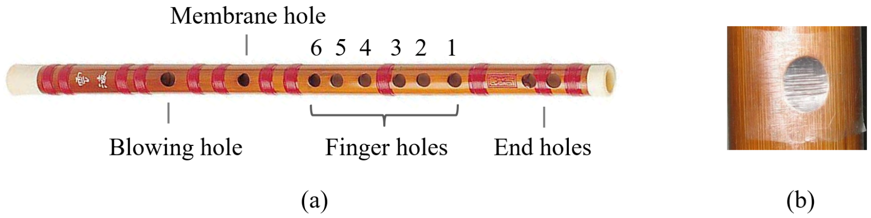


Figure 2.1: An example of a Chinese bamboo flute[5]: (a) meaning of each hole on the flute; (b) the membrane hole covered with a membrane.

There are two main schools of CBF music: the Southern style and the Northern style, which originate from the regional Chinese opera styles prevailing from the 17th century (Li, 2016; Zhao, 2001; Zhang, 2011). At that stage, the CBF is an important accompanying instrument for the operas, such as *kunqu* in the south of China and *bangzixi* in the north of China. The Southern style music is featured by slow and soft melodies, which are usually played by *qudi*; while the Northern style music is lively and bright, which are often performed by *bangdi*. *Qudi* and *bangdi* are two groups of CBF: the former is longer in length and higher in tones than the latter. Typical examples of *qudi* include C, D, and E flutes and those of *bangdi* are F, G, and A flutes (Zhan, 2009). The type of the flute is defined by the tone of the third finger hole according to Zhao (2001). The tonal range of CBFs is normally two octaves plus two tones, for example, G4 to A6 for the *qudi* in C (Tsai, 2003). A complete introduction of CBF and CBF music is beyond the scope of this thesis, where we will focus mainly on CBF playing techniques.

Listeners may often be captivated by the unique timbre of the CBF, which belies the twenty or more playing techniques invoked when per-

[5] Both (a) and (b) were adapted from the figures on https://baike.baidu.com/item/%E7%AC%9B%E5%AD%90/495283?fr=aladdin, accessed at 12:12, 08/03/2021 (GMT).

forming on the instrument. Influenced by the singing styles of the regional operas, musicians introduced some of the vocal techniques into CBF playing. Typical Southern style techniques on the CBF include trill (颤音 chanyin), appoggiatura (叠音 dieyin), end-note (赠音 zengyin), and repeated note (打音 dayin); while tonguing (吐音 tuyin), portamento (滑音 huayin), acciaccatura (垛音 duoyin), and flutter-tongue (花舌 huashe) are frequently used Northern style ones (Zhang, 2011). There are also other playing techniques such as vibrato (气颤音 qichanyin), tremolo (气震音 qizhenyin), multiphonics (泛音 fanyin), flying finger (飞指 feizhi), and circular breath (循环换气 xunhuanhuanqi), where the flying finger is a Northern style playing technique and the remaining ones are normally performed in Southern style pieces. The large repertoire of playing techniques provides the CBF a rich platform for computational analysis of playing techniques.

To the author's knowledge, there is limited computational research on the CBF. Tsai (2003) studied the acoustic effects of the membrane on CBF tones and found out that the membrane enhances upper harmonics of CBF tones while restricting its tone range. Ayers (2003; 2004) has done some analysis of CBF playing techniques through synthesis, which included only trills, tremolos and flutter-tongue. To fill the gap, we explore in this thesis the computational research on commonly used CBF playing techniques that can be generalised to other instruments and datasets, and aims at building a general framework for playing technique recognition based on the scattering transform. Prior to the introduction of the scattering transform, we review some other time–frequency representations that are frequently used for music signal analysis.

2.4 Time–Frequency Representations for Music Signal Analysis

Playing technique recognition is a classification problem where the main sources of intra-class variability are translations and small deformations, also known as time-shifts and time-warps, respectively. In this section, we introduce briefly three well-known time–frequency representations for music signal analysis: the STFT spectrogram, the constant-Q transform (CQT) spectrogram, and the mel-frequency spectrogram. The wavelet transform which forms the basis for the scattering transform is also presented. We compare these representations in terms of their invariance properties to the two types of variabilities, which significantly influence classification performance (Mallat, 2012).

2.4.1 STFT spectrogram, CQT spectrogram, mel-frequency spectrogram

As a main signal processing tool, the Fourier transform decomposes a signal $\boldsymbol{x}(t)$ into its frequency components:

$$\hat{\boldsymbol{x}}(\omega) = \int \boldsymbol{x}(t) \exp(-i\omega t) \, \mathrm{d}t, \tag{2.3}$$

for time $t \in \mathbb{R}$ and frequency $\omega \in \mathbb{R}$. Since the Fourier coefficients are averaged over the entire time domain, the transform is a globally time-shifting invariant representation, which does not provide information on when a frequency occurs over time. To localise the analysis, the short-time Fourier transform (STFT) is introduced by multiplying the signal with a window function $\boldsymbol{\phi}_T(t)$ of duration T (Andén and Mallat, 2014):

$$\mathrm{Y}\boldsymbol{x}(t, \omega) = \int \boldsymbol{x}(\tau)\boldsymbol{\phi}_T(\tau - t) \exp(-i\omega\tau) \, \mathrm{d}\tau, \tag{2.4}$$

where τ is the integration variable.

An *STFT spectrogram* $|\mathrm{Y}\boldsymbol{x}(t, \omega)|^2$ is the squared magnitude of the short-time Fourier transform (see Section 2.5.2, Müller, 2015). It is a representation locally invariant to time-shifts when the shifting amount is smaller than the window size of the STFT. However, a time-warping or dilation of the signal will result in a pitch shifting in the STFT spectrogram; therefore it is unstable to time-warping (Andén and Mallat, 2014). Additionally, the window size for the STFT is fixed at one time of the calculation, which may work for the time-shifting invariance requirement of some classification problems but not for that of the others.

The CQT spectrogram and mel-frequency spectrogram are both perceptually-motivated time–frequency representations for audio signals. As compared to the STFT with fixed time and frequency resolutions, the *constant-Q transform* comprises frequency bins logarithmically spaced and has a constant ratio of center frequencies over bandwidths, which is the quality factor (Schörkhuber and Klapuri, 2010). This means that the frequency resolution is better for low frequencies and the time resolution is better for high frequencies. The window argument in CQT is the product of time, frequency, and the reciprocal of the quality factor $q \in \mathbb{R}$, which is related to the selectivity of the analysis (Youngberg and Boll, 1978):

$$\mathrm{C}\boldsymbol{x}(t,\omega) = \int \boldsymbol{x}(\tau)\boldsymbol{\phi}_T\big((t-\tau)\omega/q\big)\exp(-j\omega\tau)d\tau. \tag{2.5}$$

The CQT is essentially a wavelet transform (see Subsection 2.4.2), which provides a certain robustness to time-warping while not being invariant to time-shifting. These invariance properties also apply to the *CQT spectrogram* $|\mathrm{C}\boldsymbol{x}(t,\omega)|^2$.

The *mel-frequency spectrogram* $\mathrm{M}\boldsymbol{x}(t,\lambda)$ is a modification of the STFT spectrogram, where the frequency bands of the latter are mapped into the mel scale by a mel filter bank $\hat{\boldsymbol{\psi}}_\lambda(\omega)$ (Andén and Mallat, 2014):

$$\mathrm{M}\boldsymbol{x}(t,\lambda) = \frac{1}{2\pi}\int |\hat{\boldsymbol{x}}(t,\omega)|^2 |\hat{\boldsymbol{\psi}}_\lambda(\omega)|^2 \,\mathrm{d}\omega. \tag{2.6}$$

$\hat{\boldsymbol{\psi}}_\lambda(\omega)$ comprises linearly-spaced bandpass filters below 1000 Hz and logarithmically-spaced bandpass filters above it (Stevens et al., 1937; Peeters, 2004), where we use λ to broadly denote the centre frequency of each filter. The mel-frequency averaging removes deformation instability by large displacements of high frequencies resulting from time-warping; however, this averaging creates information loss (Andén and Mallat, 2014).

2.4.2 Wavelet transform

To analyse signal structure with different scales, it is necessary to use time–frequency atoms with different time supports. The STFT and wavelet transform are two examples of such time–frequency atoms. Instead of using sinusoids as the STFT, the wavelet transform decomposes signals with wavelet bases. A wavelet $\boldsymbol{\psi}(t)$ is a bandpass filter with a zero average $\int_{-\infty}^{+\infty}\boldsymbol{\psi}(t)dt = 0$. It is normalised $\|\boldsymbol{\psi}(t)\| = 1$ and centred in the neighbourhood of $t = 0$. To measure the temporal evolution of frequency transients, we use a complex analytic wavelet, which can separate amplitude and phase components; therefore, its Fourier transform is null on the half-line of negative frequencies, i.e., for all $\omega < 0$, $\hat{\boldsymbol{\psi}}(\omega) = 0$. For more detailed theory on wavelet transform, we recommend Section 4.3 of Mallat (2008) and Section 5.1 of Vetterli and Kovacevic (1995).

The wavelet transform deals with deformations by separating the variations of a signal $\boldsymbol{x}(t)$ at different scales with wavelets. This is achieved by decomposing $\boldsymbol{x}(t)$ over a wavelet filterbank $\boldsymbol{\psi}_\lambda(t)$, which is dilated from a mother wavelet $\boldsymbol{\psi}(t)$ by the scaling factor of $2^{-\lambda}$ with

$$\boldsymbol{\psi}_\lambda(t) = 2^\lambda \boldsymbol{\psi}(2^\lambda t), \tag{2.7}$$

where $\lambda \in \mathbb{R}$ is the log-frequency variable of $\boldsymbol{\psi}_\lambda(t)$. Therefore, $\boldsymbol{\psi}_\lambda(t)$ is a constant-Q filterbank with the centre frequency logarithmically spaced.

As dilations increase, the time support of the wavelet expands along the time axis. For classification problems, we are often interested in a time structure smaller than T, which corresponds to a maximum scale 2^{-J}. To cover the entire frequency axis with the wavelet filter banks, we define a lowpass filter $\boldsymbol{\phi}_T(t)$ which covers the remaining frequencies. A *wavelet transform* calculates the local average of $\boldsymbol{x}(t)$ at the scale 2^{-J}, and variations at scales $2^{-\lambda} > 2^{-J}$ with wavelet convolutions (Mallat, 2016):

$$\mathrm{W}\boldsymbol{x}(t, \lambda) = \{\boldsymbol{x} * \boldsymbol{\phi}_T(t), \boldsymbol{x} * \boldsymbol{\psi}_\lambda(t)\}. \tag{2.8}$$

The modulus of the wavelet transform is called a *wavelet modulus transform*:

$$|\mathrm{W}|\boldsymbol{x}(t, \lambda) = \{\boldsymbol{x} * \boldsymbol{\phi}_T(t), |\boldsymbol{x} * \boldsymbol{\psi}_\lambda(t)|\}. \tag{2.9}$$

Different from the representations discussed in Subsection 2.4.1, the wavelet transform is stable to small deformations (time-warping) but it is not translation (time-shifting) invariant (Mallat, 2012). Examining the scalogram, Andén and Mallat (2014) found that although the modulus operation removes the complex phase, it does not lose information because the temporal variation of the multiscale envelopes is kept. Averaging these multiscale envelopes will produce a representation invariant to time shifts. However, this averaging comes at the detriment of fast temporal modulations with time structures smaller than T. In this context, the core idea of the scattering transform is to recover these temporal modulations by means of a second wavelet decomposition, modulus, and averaging. Therefore, the obtained scattering representation has both desirable properties: stability to time warps and invariance to time shifts.

2.5 Scattering Transform

In this section, we introduce the scattering transform and provide an overview of different scattering operators. Proposed in Mallat (2012), the scattering transform has the structure of a convolutional neural network (CNN): both comprise a cascade of convolutions, nonlinearities, and pooling operations. The difference is that the filters of the scattering transform are not learnt but defined as wavelets.

Figure 2.2 displays the cascading of scattering operations with the example of a musical trill. Let $\boldsymbol{x}(t)$ be an audio waveform and $\boldsymbol{\psi}_{\lambda_k}(t)$ with $k \in \mathrm{N}_+$ the wavelet filterbank at the kth-order scattering decomposition. $t \in \mathbb{R}$ is the time variable and $\lambda_k \in \mathbb{R}$ is the log-frequency variable of $\boldsymbol{\psi}_{\lambda_k}(t)$. Here, an "order" of the scattering transform is analogous to a "layer" in terms of CNNs. Take the first order for instance: by convolving $\boldsymbol{x}(t)$ with each wavelet in $\boldsymbol{\psi}_{\lambda_1}(t)$ and applying complex modulus, we obtain the first-order wavelet modulus transform $\mathbf{U}_1\boldsymbol{x}(t, \lambda_1)$, also known as scalogram. Note that $\mathbf{U}_1\boldsymbol{x}(t, \lambda_1)$ is stable to small deformations but not translation-invariant. The scattering transform aims at an invariance property up to some time structure T by average pooling, which is realised by applying to each frequency band in $\mathbf{U}_1\boldsymbol{x}(t, \lambda_1)$ a lowpass filter $\boldsymbol{\phi}_T(t)$ of cutoff frequency T^{-1}. This results in the first-order scattering transform $\mathbf{S}_1\boldsymbol{x}(t, \lambda_1)$. Cascading the operations of wavelet convolutions with $\boldsymbol{\psi}_{\lambda_k}(t)$ and complex modulus generates a "scattering network", after which the average pooling of the kth-order wavelet modulus transform $\mathbf{U}_k\boldsymbol{x}(t, \lambda_k)$ by $\boldsymbol{\phi}_T(t)$ yield the kth-order scattering transform $\mathbf{S}_k\boldsymbol{x}(t, \lambda_k)$. For completeness, we also extract the zeroth-order scattering transform $\mathbf{S}_0\boldsymbol{x}(t)$ by convolving $\boldsymbol{x}(t)$ with $\boldsymbol{\phi}_T(t)$.

Similar to CNNs which may have horizontal and vertical filters (Goodfellow et al., 2016), one may apply wavelet convolutions along the frequency axis of a given time–frequency representation. The different ways of applying wavelet convolutions form different scattering operators, as shown in Figure 2.3. Each operator captures a specific signal pattern, thus making the scattering transform a flexible framework for different music signal analysis tasks. We present in detail the time scattering, separable scattering, and joint scattering, which are the operators that

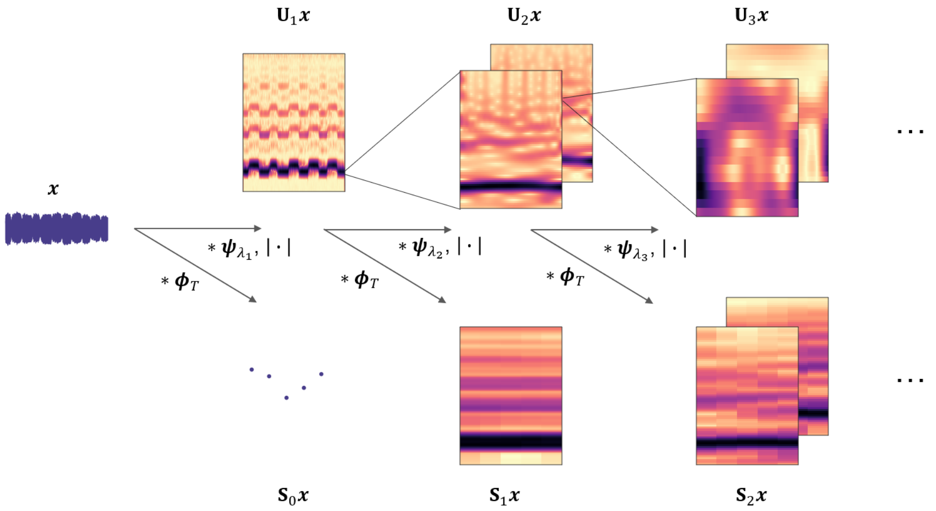

$\mathbf{U}_1\boldsymbol{x}$
$\mathbf{U}_2\boldsymbol{x}$
$\mathbf{U}_3\boldsymbol{x}$
$\boldsymbol{x}$
…
$*\boldsymbol{\psi}_{\lambda_1}, |\cdot|$
$*\boldsymbol{\psi}_{\lambda_2}, |\cdot|$
$*\boldsymbol{\psi}_{\lambda_3}, |\cdot|$
$*\boldsymbol{\phi}_T$
$*\boldsymbol{\phi}_T$
$*\boldsymbol{\phi}_T$
…
$\mathbf{S}_0\boldsymbol{x}$
$\mathbf{S}_1\boldsymbol{x}$
$\mathbf{S}_2\boldsymbol{x}$

Figure 2.2: Diagram of the scattering transform for a trill example. Convolving waveform $\boldsymbol{x}$ with wavelet filterbank $\boldsymbol{\psi}_{\lambda_1}$ and taking complex modulus obtain the first-order wavelet modulus transform $\mathbf{U}_1\boldsymbol{x}$. Average pooling of $\mathbf{U}_1\boldsymbol{x}$ by lowpass filter $\boldsymbol{\phi}_T$ results in the first-order scattering transform $\mathbf{S}_1\boldsymbol{x}$. Cascading these operations, i.e., convolution with $\boldsymbol{\psi}_{\lambda_k}$ ($k \in \mathbf{N}_+$), complex modulus, and average pooling by $\boldsymbol{\phi}_T$, generates the kth-order wavelet modulus transform $\mathbf{U}_k\boldsymbol{x}$ and scattering transform $\mathbf{S}_k\boldsymbol{x}$, forming a "scattering network".

the subsequent chapters of the thesis are built upon, and leave the exploration of the spiral scattering (Lostanlen and Mallat, 2015) as future work. Throughout the thesis, we use Morlet wavelets for wavelet convolutions in the scattering framework. This is because Morlet wavelets have an exactly null average while reaching a quasi-optimal trade-off in time–frequency localisation (Mallat, 2008). Our source code is based on the ScatNet toolbox[6] and is publicly available for reproducibility at `c4dm.eecs.qmul.ac.uk/CBFdataset.html`.

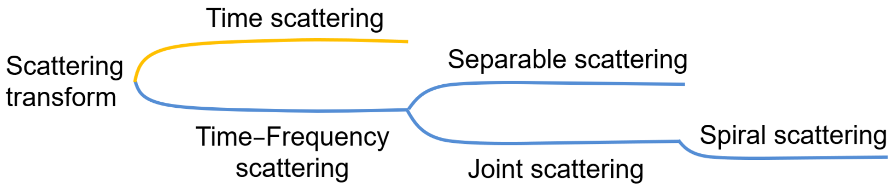


Figure 2.3: Relationship between different operators in the scattering transform framework.

[6] https://www.di.ens.fr/data/software/scatnet

2.5.1 Time scattering

According to Andén and Mallat (2014), the scattering transform preserves the energy of the signal; and for an audio signal with T below 1.5 secs, the energy is mainly absorbed by the first- and second-order scattering transform while the third-order and above capture negligible amounts. We focus in this thesis on the scattering transform of these two orders only. For simplicity, we denote the log-frequency variables of the wavelet filterbanks at the first and second order as λ and v_{t}, replacing λ_1 and λ_2 in $\boldsymbol{\psi}_{\lambda_k}(t)$ above. The corresponding wavelet filterbanks are then $\boldsymbol{\psi}_{\lambda}(t)$ and $\boldsymbol{\psi}_{v_{\mathrm{t}}}(t)$. $\boldsymbol{\psi}_{\lambda}(t)$ is obtained by dilation of a "mother wavelet" $\boldsymbol{\psi}(t)$ with a scaling factor equal to $2^{-\lambda}$, yielding:

$$\boldsymbol{\psi}_{\lambda}(t) = 2^{\lambda}\boldsymbol{\psi}\big(2^{\lambda}t\big), \tag{2.10}$$

and likewise at the second order, $\boldsymbol{\psi}_{v_{\mathrm{t}}}(t)$ is generated by replacing λ with v_{t} in Eq. (2.10). We also use the notation $\mathbf{X}(t,\lambda)$ as a shorthand for the scalogram of the waveform $\boldsymbol{x}(t)$:

$$\mathbf{X}(t,\lambda) = \mathbf{U}_1\boldsymbol{x}(t,\lambda) = \big|\boldsymbol{x} * \boldsymbol{\psi}_{\lambda}\big|(t). \tag{2.11}$$

After averaging $\mathbf{X}(t,\lambda)$ along the time axis by a lowpass filter $\boldsymbol{\phi}_T(t)$, we obtain the *first-order scattering transform* (Mallat, 2012):

$$\mathbf{S}_1\boldsymbol{x}(t,\lambda) = \Big(\big|\boldsymbol{x} * \boldsymbol{\psi}_{\lambda}\big| * \boldsymbol{\phi}_T\Big)(t), \tag{2.12}$$

which is locally invariant to time-shifting and stable to time-warping.

Similarly, we decompose each frequency band of the scalogram $\mathbf{X}(t,\lambda)$ by another wavelet filterbank $\boldsymbol{\psi}_{v_{\mathrm{t}}}(t)$. We denote the log-frequency variable associated to this filterbank by v_{t}, where the subscript t signifies that it captures the temporal variation of the scalogram. After complex modulus and local averaging, we then obtain the *second-order scattering transform* (Mallat, 2012):

$$\mathbf{S}_2\boldsymbol{x}(t,\lambda,v_{\mathrm{t}}) = \Big(\big|\mathbf{X} \overset{t}{*} \boldsymbol{\psi}_{v_{\mathrm{t}}}\big| \overset{t}{*} \boldsymbol{\phi}_T\Big)(t,\lambda), \tag{2.13}$$

where the symbol $\overset{t}{*}$ denotes a one-dimensional (1-D) convolution over the time variable t. When applied to the two-dimensional (2-D) scalogram $\mathbf{X}(t,\lambda)$, this 1-D convolution is implicitly broadcast over the variable λ.

To capture only the temporal variation regardless of the absolute energy of the waveform, we normalise the second-order coefficients $\mathbf{S}_2\boldsymbol{x}(t,\lambda,v_{\mathrm{t}})$ over the first-order coefficients $\mathbf{S}_1\boldsymbol{x}(t,\lambda)$. Motivated by auditory perception, the logarithm is applied to the normalised coefficients (Andén and Mallat, 2014). The *log-normalised second-order scattering transform* is expressed as (Andén and Mallat, 2014):

$$\widetilde{\mathbf{S}}_2\boldsymbol{x}(t,\lambda,v_{\mathrm{t}}) = \log_2\left(\frac{\mathbf{S}_2\boldsymbol{x}(t,\lambda,v_{\mathrm{t}})}{\mathbf{S}_1\boldsymbol{x}(t,\lambda)+\varepsilon}\right), \tag{2.14}$$

where $\varepsilon > 0$ is a small additive offset whose role is to avoid division by zero.

The above convolutions are calculated in the time domain only, thus we also call $\mathbf{S}_1\boldsymbol{x}$ and $\mathbf{S}_2\boldsymbol{x}$ the first- and second-order *time scattering*. Since the time scattering is the fundamental form of the scattering transform, we also refer to it as the *standard time scattering* in the subsequent chapters. It captures the long-term temporal structure of the signal, which is invariant to time-shifts and stable to time-warps.

2.5.2 Time–Frequency scattering

In addition to the invariance to time-shifts and time-warps provided by the time scattering, the time–frequency scattering goes further by adding frequency scattering (Andén et al., 2019). The frequency scattering has a similar framework as the time scattering, but applies a spectral wavelet filterbank along the log-frequency axis. This operation provides frequency transposition invariance in the log-frequency dimension. Within the time–frequency scattering, there are two sub-categories, *separable* scattering (Baugé et al., 2013) and *joint* scattering (Andén et al., 2019), depending on how the frequency scattering is implemented. We investigate the performance of these two operators for playing technique recognition in Chapter 4 and Chapter 5, respectively, and compare their performance with that of the proposed scattering transform variants. We discuss the original definition and interpretation of these two operators in this section.

Separable Scattering

The *separable scattering* comprises separable steps of time and frequency scattering transforms (Baugé et al., 2013). It was proposed to capture variations along the log-frequency axis and to provide frequency-transposition invariance. Convolving $\mathbf{S}_2\boldsymbol{x}(t, \lambda, \upsilon_\mathrm{t})$ with a spectral filterbank $\boldsymbol{\psi}_{\upsilon_\mathrm{f}}(\lambda)$ along the log-frequency axis, taking complex modulus, and averaging, we obtain the second-order separable scattering:

$$\mathbf{S}_2^{\mathrm{separa}}\boldsymbol{x}(t, \lambda, \upsilon_\mathrm{t}, \upsilon_\mathrm{f}) = \Big(\big|\mathbf{S}_2\boldsymbol{x} \overset{\lambda}{*} \boldsymbol{\psi}_{\upsilon_\mathrm{f}}\big| \overset{\lambda}{*} \boldsymbol{\phi}_F\Big)(t, \lambda). \tag{2.15}$$

$\boldsymbol{\phi}_F(\lambda)$ is a lowpass filter along the log-frequency axis, providing a frequency transposition invariance up to F (in octave units). The spectral wavelet filterbank $\boldsymbol{\psi}_{\upsilon_\mathrm{f}}(\lambda)$ is dilated from the mother wavelet $\boldsymbol{\psi}(\lambda)$ by a scaling factor of $2^{-\upsilon_\mathrm{f}}$.

When the desired property is frequency-transposition invariance only with no requirement on capturing spectral variations, we can average $\mathbf{S}_2\boldsymbol{x}(t, \lambda, \upsilon_\mathrm{t})$ directly along the log-frequency axis by $\boldsymbol{\phi}_F(\lambda)$. This forms a special case of the separable scattering, i.e., the *frequency-averaged time scattering* (Andén and Mallat, 2014):

$$\mathbf{S}_2^{\mathrm{freqAver}}\boldsymbol{x}(t, \lambda, \upsilon_\mathrm{t}, \upsilon_\mathrm{f}) = \Big(\mathbf{S}_2\boldsymbol{x} \overset{\lambda}{*} \boldsymbol{\phi}_F\Big)(t, \lambda). \tag{2.16}$$

Joint scattering

To capture time–frequency geometry along the time and the acoustic frequency axes simultaneously requires applying the time and the frequency scattering jointly, which was defined as the *joint time–frequency scattering* (Andén et al., 2019). Rather than decomposing the signal by a temporal and a spectral wavelet filterbank in separate steps, the joint time–frequency scattering uses a wavelet filterbank dilated from a 2-D mother wavelet $\boldsymbol{\Psi}(t, \lambda) = \boldsymbol{\psi}^{(\mathrm{t})}(t)\boldsymbol{\psi}^{(\mathrm{f})}(\lambda)$, which is the product of the two 1-D mother wavelets along the time and the log-frequency axes. An orientation variable $\theta = \pm 1$ is introduced to reflect the oscillation direction (up or down) of the spectro-temporal pattern. $\theta = -1$ flips the centre frequency of wavelet $\boldsymbol{\psi}^{(\mathrm{f})}(\lambda)$ from 2^λ to -2^λ. Dilating by $2^{-\upsilon_\mathrm{t}}$ along t and dilating by $2^{-\upsilon_\mathrm{f}}$ along λ, and reflecting according the θ

yields the 2-D wavelet filterbank:

$$\boldsymbol{\Psi}_{v_{\mathrm{t}},v_{\mathrm{f}},\theta}(t,\lambda) = (2^{v_{\mathrm{t}}+v_{\mathrm{f}}})\boldsymbol{\psi}^{(\mathrm{t})}(2^{v_{\mathrm{t}}}t)\boldsymbol{\psi}^{(\mathrm{f})}\big(\theta 2^{v_{\mathrm{f}}}\lambda\big). \tag{2.17}$$

The second-order joint time–frequency scattering $\mathbf{S}_2^{\mathrm{joint}}\boldsymbol{x}(t,\lambda,v_{\mathrm{t}},v_{\mathrm{f}},\theta)$ is then obtained by convolving the scalogram $\mathbf{X}(t,\lambda)$ with $\boldsymbol{\Psi}_{v_{\mathrm{t}},v_{\mathrm{f}},\theta}(t,\lambda)$, taking complex modulus, and averaging by a 2-D lowpass filter $\boldsymbol{\phi}_{T,F}(t,\lambda)$ with translation invariance up to T and frequency-transposition invariance up to F:

$$\mathbf{S}_2^{\mathrm{joint}}\boldsymbol{x}(t,\lambda,v_{\mathrm{t}},v_{\mathrm{f}},\theta) = \Big(\big|\mathbf{X} \overset{t,\lambda}{*} \boldsymbol{\Psi}_{v_{\mathrm{t}},v_{\mathrm{f}},\theta}\big| \overset{t,\lambda}{*} \boldsymbol{\phi}_{T,F}\Big)(t,\lambda), \tag{2.18}$$

where the symbol $\overset{t,\lambda}{*}$ denotes a 2-D convolution over both the time variable t and the log-frequency variable λ.

Spiral scattering

Although the spiral scattering is not the focus of this thesis, we include it with a brief introduction for completeness. Besides wavelet convolutions along the time and the log-frequency axes, the spiral scattering adds a convolution across octaves to capture the harmonic structure of voiced sounds, such as vowels or music tones (Lostanlen and Mallat, 2015). This is achieved by rolling up the logarithm of the acoustic frequency, i.e., λ, into a pitch spiral, where octave intervals correspond to full turns and the partials with distance of one or multiple octave(s) get aligned on a radius. Assuming we use Q filters per octave in the first-order time scattering, pitch height $k \in \mathbb{N}$ and pitch chroma $\chi \in \mathbb{N}$ in the spiral correspond to the integer part $\lfloor\lambda\rfloor$ and the fractional part $\{\lambda\}$, respectively:

$$\lambda = \lfloor\lambda\rfloor + \{\lambda\} = k + \frac{\chi}{Q} \tag{2.19}$$

where $\chi < Q$.

The spiral mother wavelet is then defined as

$$\boldsymbol{\Psi}(t,\lambda) = \boldsymbol{\psi}^{(\mathrm{t})}(t)\boldsymbol{\psi}^{(\mathrm{k})}(\lfloor\lambda\rfloor)\boldsymbol{\psi}^{(\chi)}(\{\lambda\}), \tag{2.20}$$

which captures variations across octaves at fixed chroma and along neighbouring constant-Q bands. Dilating by $2^{-v_{\mathrm{t}}}$ along time, $2^{-v_{\mathrm{k}}}$ across octaves, and $2^{-v_{\chi}}$ along neighbouring constant-Q bands, we obtain the spiral filterbank

$$\begin{aligned}&\boldsymbol{\Psi}_{v_\mathrm{t},v_\mathrm{k},\theta_\mathrm{k},v_\chi,\theta_\chi}(t,\lambda) = \\ &\qquad \big(2^{v_\mathrm{t}+v_\mathrm{k}+v_\chi}\big)\boldsymbol{\psi}^{(\mathrm{t})}\big(2^{v_\mathrm{t}}t\big)\boldsymbol{\psi}^{(\mathrm{k})}\big(\theta_\mathrm{k}2^{v_\mathrm{k}}\lfloor\lambda\rfloor\big)\boldsymbol{\psi}^{(\chi)}\big(\theta_\chi 2^{v_\chi}\{\lambda\}\big),\end{aligned} \tag{2.21}$$

where θ_k and θ_χ are the oscillation directions along $\lfloor\lambda\rfloor$ and along $\{\lambda\}$, respectively. Convolving the scalogram $\mathbf{X}(t,\lambda)$ with the spiral wavelet filterbank above, taking complex modulus, and averaging by a lowpass filter $\boldsymbol{\phi}_{T,K,X}(t,\lfloor\lambda\rfloor,\{\lambda\})$, we obtain the second-order *spiral scattering transform*:

$$\begin{aligned}&\mathbf{S}_2^{\mathrm{spiral}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{k},\theta_\mathrm{k},v_\chi,\theta_\chi) = \\ &\qquad \Big(\big|\mathbf{X} \overset{t,k,\chi}{*} \boldsymbol{\Psi}_{v_\mathrm{t},v_\mathrm{k},\theta_\mathrm{k},v_\chi,\theta_\chi}\big| \overset{t,k,\chi}{*} \boldsymbol{\phi}_{T,K,X}\Big)(t,\lfloor\lambda\rfloor,\{\lambda\}),\end{aligned} \tag{2.22}$$

where the symbol $\overset{t,\lambda,\chi}{*}$ denotes a 3-D convolution along time, across octaves, and along neighbouring constant-Q bands. T, J, X reflects the invariance properties along the respective axes.

Chapter 3

CBFdataset: a Dataset of Chinese Bamboo Flute Performances

Due to the annotation-intensive nature and scarcity of playing techniques in music performances, there is not yet any dataset covering multiple types of playing technique annotations in real-world performances. For ecological validity, we create a new publicly available dataset with playing techniques recorded in context. The dataset, named CBFdataset, comprises full-length performances of the Chinese bamboo flute (CBF) and expert annotations of playing techniques. We investigate the difference between isolated and performed playing techniques in Section 3.1. Based on the patterns of regularity in the time–frequency domain, Section 3.2 groups commonly used playing techniques into two families. We introduce the collection process and content of the CBFdataset in Section 3.3 and Section 3.4, respectively. Publications associated with this chapter include Wang et al. (2019a), Wang et al. (2019b), Wang et al. (2020a), and Wang et al. (submitted) since these publications all used part of or the complete CBFdataset as the main evaluation dataset (see Section 1.4 for publication details).

3.1 Isolated versus Performed Playing Techniques

Up to now, most of the research literature has focused on playing techniques that have been recorded in highly controlled environments (see Subsection 2.2.2). Yet, we find that, in the context of a music performance, playing techniques exhibit considerable variations as compared to when they are played in isolation. Figure 3.1 displays the spectrograms of an isolated glissando and glissandi performed in a full-length performance from the CBFdataset (see Section 3.4). Audio recordings of these glissando examples and the whole performance recording are available online[1]. As can be observed, player 1 obviously lengthens the first note of the glissando at position A, while player 2 applies a co-articulation here, i.e., flutter-tongue combined with glissando. For the glissando performed by player 1 at position B, there is a pitch mutation to lower octave, rather than consecutive note changes exhibited in other cases. These variations point to a need for collecting playing techniques in real-world music recordings.

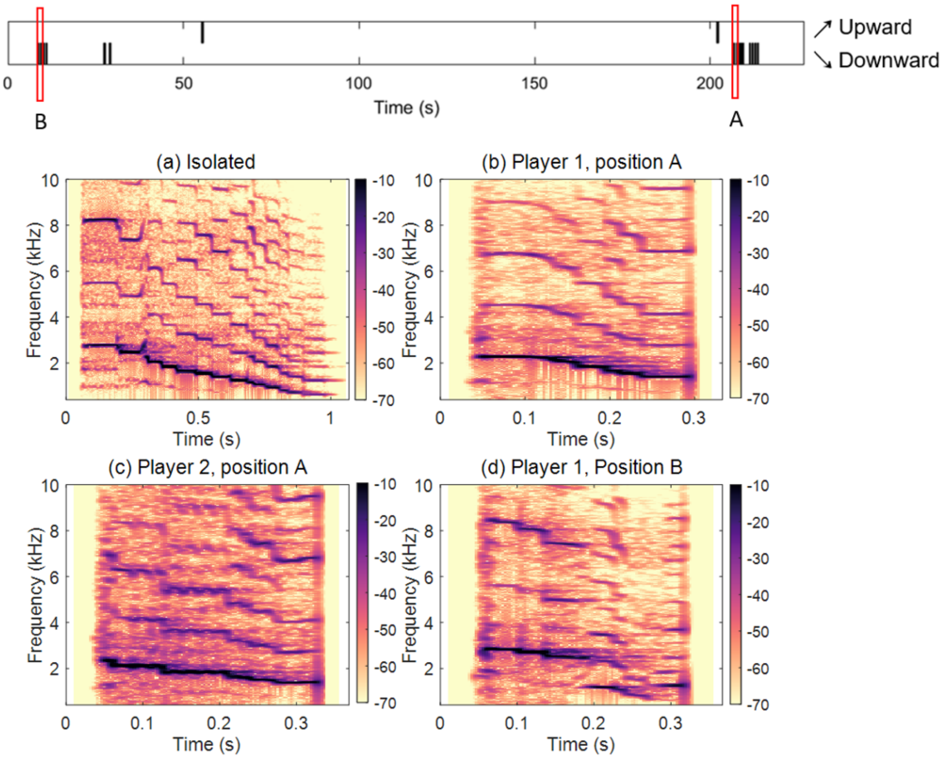


Figure 3.1: Comparison of isolated glissando and glissandi performed by two performers at different positions in the full-length performance of *Busy Delivering Harvest* in the CBFdataset (see Section 3.4). Top: time positions of upward and downward glissandi in the piece. Bottom: (a) isolated glissando; (b) performed glissando by player 1 at position A; (c) performed glissando by player 2 at position A; (d) performed glissando by player 1 at position B.

[1] https://changhongw.github.io/publications/gliss_demo.html

3.2 Playing Technique Grouping

As introduced in Section 2.2, some playing techniques are instrument-specific due to the physical characteristics of certain instruments. To develop a general framework for playing technique recognition, we aim at playing techniques that can generalise potentially across musical instruments and possibly to singing voice. According to an investigation of playing techniques in music signals as listed in Table 2.1, we focus on seven commonly used playing techniques: vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando.

When displaying these playing techniques in the time–frequency domain, we observe that each technique has a distinctive spectro-temporal pattern as shown in Figure 3.2. Based on the patterns of regularity, we group these playing techniques into two families: *pitch modulation-based techniques* (PMTs) and *pitch evolution-based techniques* (PETs). The former refers to periodic modulations that elaborate on stable pitches and are temporally symmetric. Typical examples are vibrato, tremolo, trill, and flutter-tongue shown in Figure 3.2 (a), all with periodic modulations appearing on each harmonic partial. The difference between these playing techniques exists in the rate, frequency depth, and shape of the modulations, which we introduce in detail in Section 4.1. PETs are playing techniques which contain monotonic pitch changes over time and are temporally asymmetric. Acciaccatura, portamento, and glissando in Figure 3.2 (b) are three examples from this group of playing techniques. Portamento is a continuous slide between two notes. Glissando is a slide across a series of discrete tones. Acciaccatura, in the case of the Chinese bamboo flute includes a sharp attack and strong air flow on the first note followed by a rapid transition to the second note. We develop recognition systems for PMTs and PETs in Chapter 4 and Chapter 5, respectively.

In the case of CBF, these seven playing techniques—vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando—are also known as 气颤音 (qichanyin), 气震音 (qizhenyin), 指颤音 (zhichanyin), 花舌 (huashe), 垛音 (duoyin), 滑音 (huayin), and 历音 (liyin) (Zhan, 2009). The first three are typical playing techniques in the Southern style pieces, while the last four are those frequently used in the Northern style pieces. The difference between the Southern and Northern styles can be found in Section 2.3.

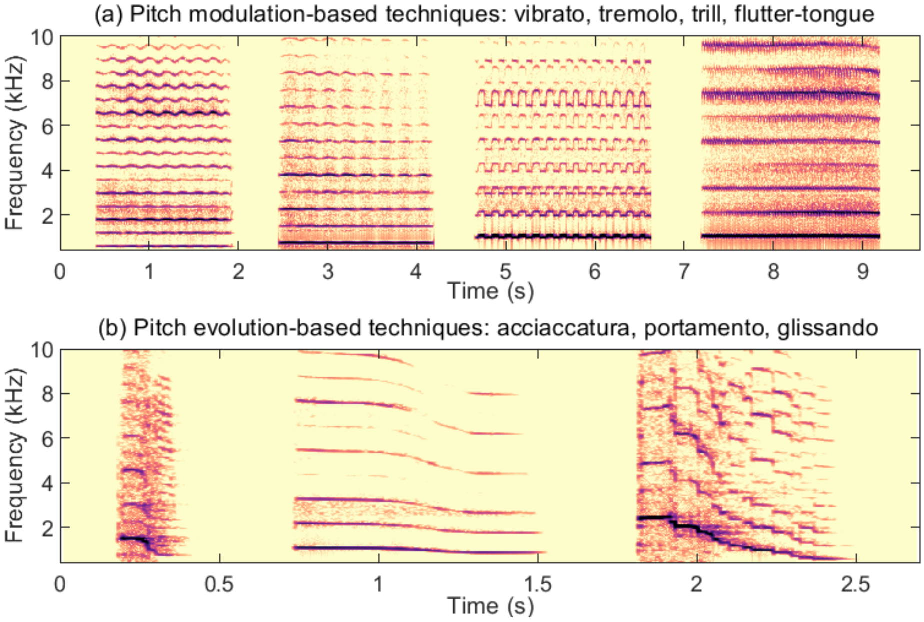


Figure 3.2: Spectrograms of commonly used playing techniques in musical signals which are grouped into two families: (a) pitch modulation-based techniques (PMTs) and (b) pitch evolution-based techniques (PETs).

3.3 CBFdataset Collection and Annotation

Our data collection[2] process has taken into account diversity of playing techniques, performers, flute types, pieces, and styles. The pieces were selected based on the discussions with Xiaojie Meng, who is a professional CBF performer and educator. The selected pieces are all classic CBF performances which are abundant of playing techniques. The performers were doctoral, master, and undergraduate students from the China Conservatory of Music, studying Chinese bamboo flute performing. This conservatory is one of the most authoritative conservatories on CBF education and research in the world. The performers were selected based on the demographics and years of training. All data was recorded in acoustically treated environments of a professional recording studio using a Zoom H6 recorder with its stock microphones, in xy stereo configuration, at 44.1kHz/24-bits.

[2] The data collection gained ethics approval by Queen Mary University of London with the reference number 1732.

The annotations of the playing techniques were a joint effort of the performers, the author of this thesis, the author's supervisors (Dr Emmanouil Benetos and Prof Elaine Chew), and authoritative CBF educators. Seven commonly used playing techniques (see Section 3.2), i.e., vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando, were annotated in all performances. During the annotation process, the recordings and the standard scores were given back to the players after their performance. The performers annotated the playing techniques on the scores by listening to their own recordings. The author of the thesis transferred them into audio-synchronised annotations using Sonic Visualiser (Cannam et al., 2010). Each playing technique was annotated with a start time, an end time, and playing technique type. Rounds of discussions with the performers were launched if there were uncertain labelling cases until a final agreement was reached. All the recordings and annotations are publicly available at `c4dm.eecs.qmul.ac.uk/CBFdataset.html`.

3.4 CBFdataset Content and Statistics

The complete CBFdataset comprises 2.6-hour recordings of monophonic full-length performances and isolated playing techniques on the Chinese bamboo flute, and annotations of seven types of playing techniques: vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando. Note that the CBFdataset was originally split into two subsets, the CBF-periDB and the CBF-petsDB, according to the groups of playing techniques introduced in Section 3.2. These two subsets were released separately in the associated publications (Wang et al., 2019a) and (Wang et al., 2020a), respectively. In the thesis, we combine them into one as the CBFdataset, and use the entire dataset as the evaluation dataset throughout the thesis for CBF playing technique recognition.

All data in the CBFdataset was recorded by 10 professional CBF performers. Each performer played both isolated playing techniques covering all notes on the CBF and two full-length pieces among *Busy Delivering Harvest* 《扬鞭催马运粮忙》, *Jolly Meeting* 《喜相逢》, *Morning* 《早晨》, and *Flying Partridge* 《鹧鸪飞》. The first two are Northern style pieces performed on the G flute, while the last two are Southern style ones played on the C flute (see an introduction of the CBF music style and CBF type in Section 2.3). Each piece was performed using the type of flute originally suggested by the composer which was annotated on the score. Performers were grouped by flute type, i.e., each player performed one Southern and one Northern style piece (except one player who performed two Northern pieces for personal reasons). Each performer used their own flutes.

Figure 3.3 shows the number of examples of each type of playing technique performed by each performer in the CBFdataset. The top subfigure displays the number of playing techniques in the full-length recordings of the four pieces: *Busy Delivering Harvest* (BH), *Jolly Meeting* (JM), *Morning* (Mo), and *Flying Partridge* (FP). The bottom subfigure shows the number of isolated playing techniques of each type performed by each performer except isolated tremolos, which were not recorded at the data collection. As can be observed, the number of examples for different playing technique classes is highly imbalanced, even for the same piece performed by different performers. For example, the number of both flutter-tongue and portamento techniques used by Performer 5 in the piece *Busy Delivering Harvest* are nearly two times that applied by Performer 8 to the same piece. Northern style playing techniques, i.e., flutter-tongue, acciaccatura, portamento, and glissando, rarely appear in the Southern style pieces except *Morning*. *Morning* is the first Southern style CBF piece that has Northern style playing techniques introduced (Zhang, 2011).

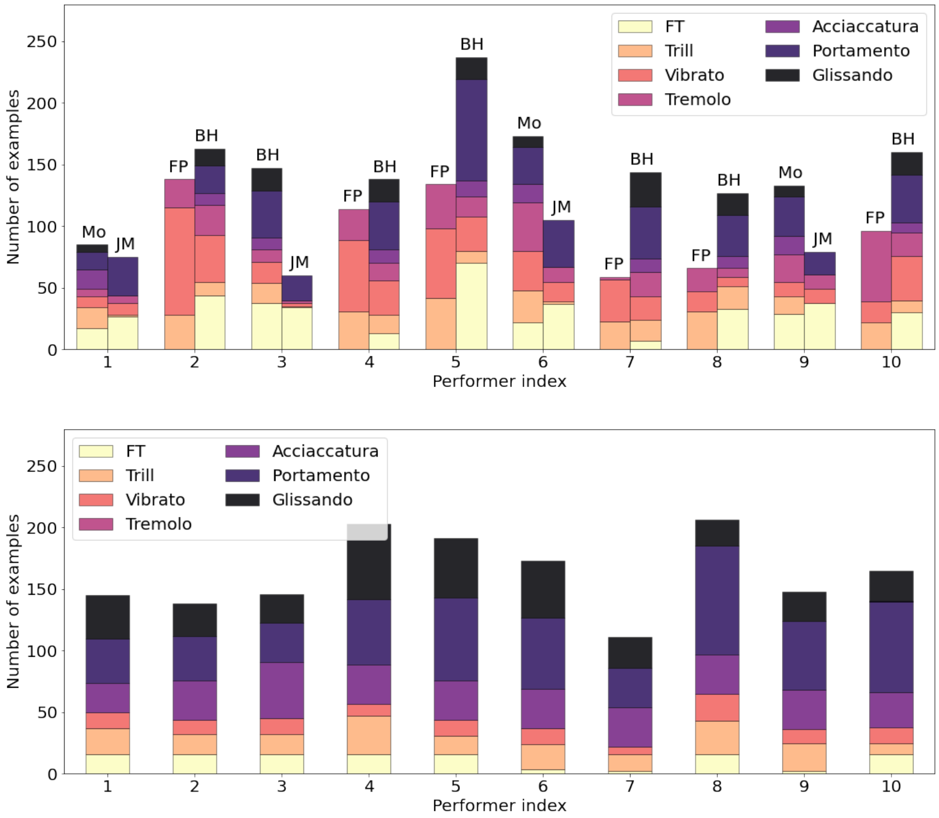


Figure 3.3: Number of examples for each type of playing technique performed by each performer in the CBFdataset (FT=flutter-tongue). Top: number of playing technique examples in the full-length pieces; Bottom: number of isolated playing techniques. BH=*Busy Delivering Harvest*, JM=*Jolly Meeting*, Mo=*Morning*, and FP=*Flying Partridge* are the names of the four pieces. The first two are Northern style pieces performed on the G flute and the other two are Southern style ones played on the C flute. Each performer used their own flutes of these two types.

3.5 Conclusions

In this chapter, we have created the first dataset on the Chinese bamboo flute (CBF), the CBFdataset, for ecological valid analysis of playing techniques in context. The dataset comprise recordings of full-length performances and isolated playing techniques on the CBF, and playing technique annotations. Aiming at developing a general framework for playing technique recognition, we annotated seven commonly used playing techniques—vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando—that may be generalised potentially to other instruments and possibly to singing voice. Including mainly monophonic real-world performances, the CBFdataset could also be potentially used for other computational music research such as music performance analysis (Lerch et al., 2020) and music style transfer (Dai et al., 2018).

Chapter 4

Adaptive Scattering for Pitch Modulation-based Technique Recognition

Vibratos, tremolos, trills, and flutter-tongue are playing techniques frequently found in vocal and instrumental music. A common feature of these techniques is the periodic modulations bound to stable pitches which we define as the pitch modulation-based techniques (PMTs) in Section 3.2. In this chapter we propose a variant of the scattering transform, the adaptive scattering, which provides representations that are invariant to large frequency-transpositions besides the invariance to time-shifts and time-warps of the standard time scattering introduced in Section 2.5 and that exhibit lower redundancy. Based on the characteristics of PMTs, we explore three types of adaptive scattering operators, the adaptive time scattering, the adaptive time–rate scattering, and the combination of these two operators, for detecting and classifying PMTs in real-world music performances. We analyse characteristics of PMTs in Section 4.1 and define the adaptive scattering representations in Section 4.2. Section 4.3 and Section 4.4 present the recognition system and evaluation results, respectively. Publications associated with this chapter include Wang et al. (2019a) and Wang et al. (submitted) (see Section 1.4 for publication details).

4.1 Characteristics of Pitch Modulation-based Techniques

Although all PMTs result in some periodic modulations in the time–frequency domain, each has distinct characteristics, as listed in Table 4.1. The extent and shape characteristics are based on music theory and the rate information is summarised from the CBFdataset (see Section 3.4). Flutter-tongue has a much higher modulation rate as compared to the other three modulations. For the other three types of techniques with similar modulation rates, the discriminative information lies in the modulation extent and shape of the modulation unit. The *modulation unit* refers to the unit pattern that repeats periodically within the modulation. It can be an amplitude modulation (AM), a frequency modulation (FM), or a spectro-temporal modulation, as intuitively observed from the spectrograms shown in Figure 4.1. Trills are note-level modulations, for which the frequency variations are at least one semitone. This extent of modulation is much larger than that of vibratos and tremolos. The shape of the modulation unit for the trill is more square-like than vibratos' sinusoidal form. The difference between vibrato and tremolo is that vibratos are FMs, while tremolos are AMs. We show how this discriminative information is captured by the proposed adaptive scattering representations in Section 4.2.

Technique	Rate (Hz)	Extent	Shape
Flutter-tongue	25-50	$<$ 1 semitone	Sawtooth-like
Vibrato	3-10	$<$ 1 semitone	Sinusoidal (FM)
Tremolo	3-8	$\approx$ 0 semitone	Sinusoidal (AM)
Trill	3-10	Note level	Square-like

Table 4.1: Characteristics of pitch modulation-based techniques (PMTs).

4.2 Adaptive Scattering

Although the second-order time scattering (see Subsection 2.5.1) is able to capture amplitude and frequency modulations in audio signals (Andén and Mallat, 2012), it is a redundant representation for PMTs. This is because the second-order time scattering decomposes all frequency

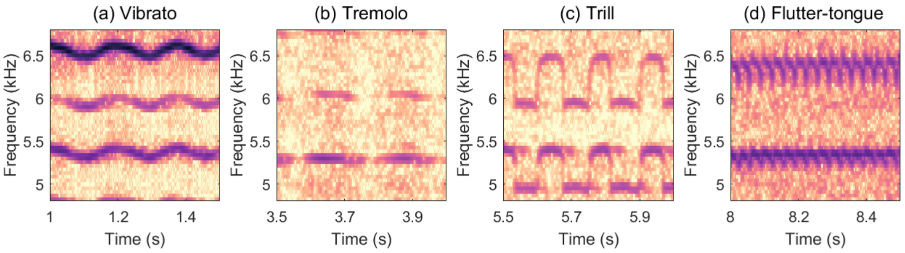


Figure 4.1: Visual comparison of PMT characteristics for (a) vibrato, (b) tremolo, (c) trill, and (d) flutter-tongue, in the time–frequency domain (partially enlarged from Figure 3.2 (a)).

bands of the scalogram; while for PMTs, the harmonic structure of the modulations, as observed in Figure 4.1, suggests that one harmonic partial sufficiently captures all the characteristic information: rate, extent, and shape. Additionally, playing techniques may be performed on pitches covering the whole tonal range of an instrument or on tones that a singer can possibly sing; therefore playing techniques require representations invariant to large frequency-transpositions.

To reduce representation redundancy without losing the discriminative information and to provide large frequency-transposition invariance, we propose the *adaptive scattering*, a variant of the scattering transform framework (see Section 2.5), for representing PMTs. Instead of decomposing all frequency bands of the scalogram, the adaptive scattering calculates the second-order scattering transform adaptively around the *decomposition trajectory*, a one-dimensional time series in the scalogram. This is how the "adaptive" term comes from. Typical decomposition trajectories include dominant band, fundamental frequency, and predominant melody, which we discuss in Subsection 4.2.2. Based on the characteristics of PMTs, we propose three adaptive scattering representations in Subsection 4.2.1, the adaptive time scattering, the adaptive time–rate scattering, and the combination of these two operators.

4.2.1 Adaptive scattering representations

In this section, we introduce the adaptive scattering representations—the adaptive time scattering, the adaptive time–rate scattering, and their combination—using the dominant band as the example decomposition trajectory. Investigation of other decomposition trajectories is described in Subsection 4.2.2. From the first-order time scattering coefficients $\mathbf{S}_1\boldsymbol{x}(t,\lambda)$ (see Subsection 2.5.1) shown in Figure 4.2 (b), we extract the frame-wise index of the frequency band with maximum acoustic energy:

$$\lambda_{\max}(t) = \arg\max_{\lambda}\Big(\mathbf{S}_1\boldsymbol{x}(t,\lambda)\Big), \tag{4.1}$$

where $t \in \mathbb{R}$ and $\lambda \in \mathbb{R}$ are the time variable and the log-frequency variable of the wavelet filterbank $\boldsymbol{\psi}_\lambda(t)$, respectively. $\boldsymbol{\psi}_\lambda(t)$ is the wavelet filterbank dilated from the mother wavelet $\boldsymbol{\psi}(t)$ by a scaling factor of $2^{-\lambda}$. $\lambda_{\max}(t)$ forms the *dominant band trajectory.*

PMTs are spectro-temporal patterns normally spreading over several frequency bands. To extract information of the full modulation pattern, we introduce an L-band tolerance centred symmetrically at the decomposition trajectory. L is the total number of frequency bands decomposed. We then define the *expanded decomposition trajectory* as:

$$\Lambda(t) = \left\{\lambda_{\max}(t) + l \,\middle|\, -\frac{L}{2} \leq l \leq \frac{L}{2}\right\}. \tag{4.2}$$

We locate the L-band decomposition trajectory of the scalogram by expressing its log-frequency axis in local coordinates with respect to the dominant band trajectory:

$$\mathbf{X}_\Lambda(t,l) = \mathbf{X}\Big(t,\Lambda\Big). \tag{4.3}$$

Convolving $\mathbf{X}_\Lambda(t,l)$ temporally with another wavelet filterbank $\boldsymbol{\psi}_{v_\mathrm{t}}(t)$, applying complex modulus, and averaging locally with $\boldsymbol{\phi}_T(t)$, we obtain the *adaptive time scattering* (AdaTS):

$$\mathbf{S}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t,l,v_\mathrm{t}) = \Big(\big|\mathbf{X}_\Lambda \overset{t}{*} \boldsymbol{\psi}_{v_\mathrm{t}}\big| \overset{t}{*} \boldsymbol{\phi}_T\Big)(t,l). \tag{4.4}$$

The wavelet filterbank $\boldsymbol{\psi}_{v_\mathrm{t}}(t)$ is also dilated from the mother wavelet $\boldsymbol{\psi}(t)$ by a scaling factor of 2^{-v_t}. $\boldsymbol{\phi}_T(t)$ is the lowpass filter providing a time-shifting invariance up to T. In the equation above, $\mathbf{S}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t,l,v_\mathrm{t})$ is a three-dimensional representation along t, l, and v_t. On the flip side, its number of log-frequency bands l is equal to L, i.e., much less than the number of log-frequency bands λ of the second-order time scattering $\mathbf{S}_2\boldsymbol{x}(t,\lambda,v_\mathrm{t})$. The reason is that PMTs are modulations bound to stable pitches rather than large frequency changes; a few frequency bands expanded around the decomposition trajectory sufficiently cover the full modulation pattern.

To capture only the temporal variation regardless of the absolute energy of the waveform, we normalise the adaptive time scattering coefficients $\mathbf{S}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t, l, v_{\mathrm{t}})$ over the corresponding first-order time scattering coefficients $\mathbf{S}_1\boldsymbol{x}(t, l) = \mathbf{S}_1\boldsymbol{x}(t, \lambda)\,|_{\lambda=l}$ and take the logarithm of the normalised coefficients (Andén and Mallat, 2014) to mimic auditory perception. Similarly to Subsection 2.5.1, we derive the *log-normalised adaptive time scattering*:

$$\widetilde{\mathbf{S}}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t, l, v_{\mathrm{t}}) = \log_2\left(\frac{\mathbf{S}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t, l, v_{\mathrm{t}})}{\mathbf{S}_1\boldsymbol{x}(t, l) + \varepsilon}\right), \tag{4.5}$$

where ε is a small additive offset that avoids division by zero. Figure 4.2 (c) shows the log-normalised AdaTS decomposed from the dominant band trajectory in Figure 4.2 (a).

Besides the difference on the fundamental modulation rate, the AdaTS of PMTs exhibits different spectral characteristics along the modulation rate axis, as observed from Figure 4.2 (c). Tremolo has only the fundamental modulation rate while trill and vibrato carry upper harmonic partials. Trill has a richer harmonic structure than vibrato. These characteristics may provide additional information for the recognition of PMTs. Therefore, we propose to apply frequency scattering along the modulation rate axis of $\widetilde{\mathbf{S}}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t, l, v_{\mathrm{t}})$ and define the resulting representation as the *adaptive time–rate scattering* (AdaTRS):

$$\mathbf{S}_2^{\mathrm{AdaTRS}}\boldsymbol{x}(t, l, v_{\mathrm{t}}, v_{\mathrm{f}}) = \left(\left|\widetilde{\mathbf{S}}_2^{\mathrm{AdaTS}}\boldsymbol{x} \overset{v_{\mathrm{t}}}{*} \boldsymbol{\psi}_{v_{\mathrm{f}}}\right| \overset{v_{\mathrm{t}}}{*} \boldsymbol{\phi}_F\right)(t, l, v_{\mathrm{t}}). \tag{4.6}$$

$\boldsymbol{\psi}_{v_{\mathrm{f}}}(v_{\mathrm{t}})$ is a wavelet filterbank dilated from the mother wavelet $\boldsymbol{\psi}(v_{\mathrm{t}})$ by a scaling factor of $2^{-v_{\mathrm{f}}}$. $\boldsymbol{\phi}_F(v_{\mathrm{t}})$ is a lowpass filter along the modulation rate axis. The frequency scattering has a similar form as the time scattering where the former generally applies a wavelet filterbank along the acoustic frequency axis, i.e., $\boldsymbol{\psi}_{v_{\mathrm{f}}}(\lambda)$, such as the joint time–frequency scattering and the separable time–frequency scattering introduced in Subsection 2.5.2. In such cases, it generates representations that are invariant to frequency-transpositions and captures modulation information along the log-frequency axis of the scalogram. In this chapter, the proposed AdaTS itself is invariant to large frequency-transpositions due to the adaptive operation. Applying frequency scattering on top of

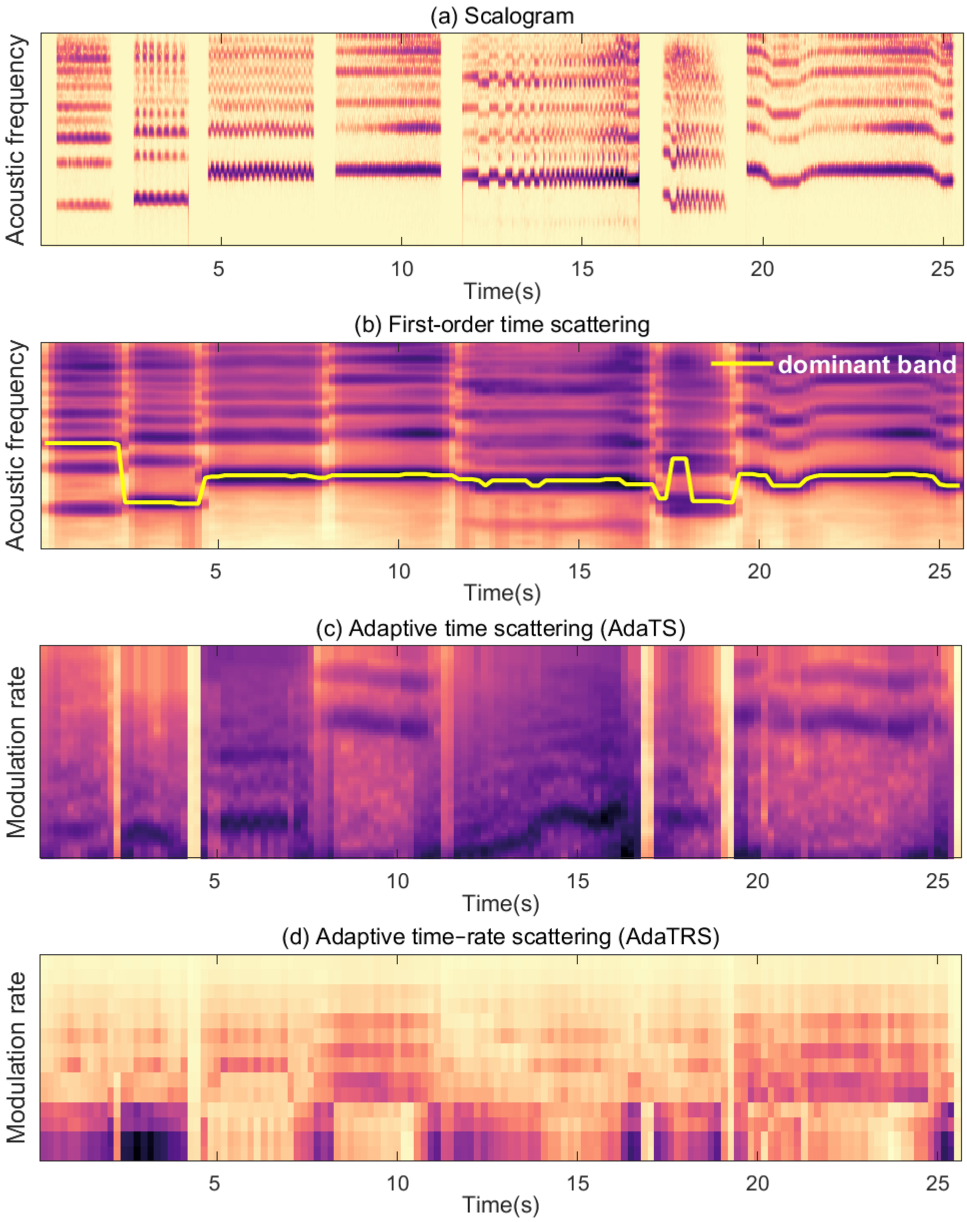


Figure 4.2: Extracting the adaptive scattering representations for PMTs: vibrato, tremolo, trill, flutter-tongue, variable rate trill, variable extent trill, variable pitch flutter-tongue: (a) scalogram; (b) dominant band trajectory in the first-order time scattering; (c) adaptive time scattering obtained by localising and decomposing scalogram trajectories; (d) adaptive time–rate scattering obtained by applying a spectral filterbank and averaging. The AdaTS+AdaTRS is the frame-wise concatenation of (c) and (d).

the AdaTS is to capture its characteristics along the modulation rate axis. Figure 4.2 (d) displays an example of the AdaTRS by applying frequency scattering on top of Figure 4.2 (c).

We define *AdaTS+AdaTRS* as the concatenation of outputs from Eq. (4.5) and Eq. (4.6):

$$\begin{aligned}&\mathbf{S}_2^{\text{AdaTS+AdaTRS}}\boldsymbol{x}(t,l,v_{\text{t}},v_{\text{f}}) = \\ &\qquad\Big\{\widetilde{\mathbf{S}}_2^{\text{AdaTS}}\boldsymbol{x}(t,l,v_{\text{t}}), \mathbf{S}_2^{\text{AdaTRS}}\boldsymbol{x}(t,l,v_{\text{t}},v_{\text{f}})\Big\}. \qquad (4.7)\end{aligned}$$

We compare the performance of the AdaTS, AdaTRS, and AdaTS+ AdaTRS on PMT recognition in Section 4.3.

We have introduced in Subsection 2.5.2 that the scattering transform has a similar structure as convolutional neural networks (CNNs) but using predefined wavelets as the filters. The hyperparameters of these wavelets provide a way to encode the prior knowledge of the task at hand. From the analysis in Section 4.1, the core information for PMT recognition lies in the modulation rate, extent, and shape. We explain in detail on how this information can be encoded into the adaptive scattering representations using the examples in Figure 4.2. The figure shows, respectively, (a) the scalogram, (b) the first-order time scattering, (c) the adaptive time scattering, and (d) the adaptive time–rate scattering representations of a series of PMT examples in the CBFdataset (see Section 3.4). The first four are modulations based on stable pitches or constant parameters: vibrato, tremolo, trill, and flutter-tongue. The last three are cases with time-varying parameters: trill with variable rate, trill with variable extent, and flutter-tongue with time-varying pitch.

As can be seen from Figure 4.2 (c), flutter-tongue is the most discriminative PMT with the highest modulation rate. Dominant band decomposition also captures trills because of their large modulation extent. This can be interpreted by filters with a bandwidth larger than one semitone, which blurs other subtle modulations. To specifically detect vibratos and tremolos, we can use filters with bandwidth less than one semitone and concatenate the decompositions of multiple frequency bands in the scalogram. Assume we have frequency bands of 1/16 octave bandwidth in scalogram; ideally, the adaptive time scattering of tremolo should display only the fundamental modulation rate with no upper harmonics since tremolo is an AM. This is verified by the second example in Figure 4.2 (c). However, vibratos are FMs with modulations spread over neighbouring frequency bands. Decomposing neighbouring frequency bands above or below the dominant band provides additional information to distinguish vibratos from tremolos. All this discriminative information can be visualised from the fundamental modulation rate and the richness of the harmonics in the adaptive time scattering in Figure 4.2 (c). Although the last example is flutter-tongue with time-varying pitch, the modulation rates shown in Figure 4.2 (c) are relatively stable. The variable rate of the trill from 12 to 17 sec is captured by the gradually increased fundamental modulation rate. The variable extent information of the trill from 18 to 19 sec could be captured when multiple frequency bands around the dominant band are decomposed. To capture the spectral structure along the modulation rate axis of the AdaTS (Figure 4.2 (c)), we apply frequency scattering and obtain the AdaTRS (Figure 4.2 (d)), which provides additional information for discriminating the PMTs.

4.2.2 Decomposition trajectories

The dominant band trajectory discussed above corresponds to the frequency band with maximum acoustic energy, which may not be stable due to octave jumps, i.e., frequency switches between harmonic partials, as observed from the trill example at around 18 sec in Figure 4.2 (b). To suppress the influence of such frequency switches, we preprocess the dominant band trajectory by interpolation and smoothing. Zero frequency values of the trajectory are linearly interpolated according to the non-zero frequencies of neighbouring frames. A median filter is then applied to the trajectory to reduce the influence of frequency band switches. Comparing the dominant band trajectory before (Figure 4.3 top) and after (Figure 4.4 top) preprocessing, we notice that the latter exhibits higher stability, with the aforementioned octave jumps being smoothed out. Besides improving the dominant band trajectory as above, we also explore other possible decomposition trajectories for the adaptive scattering such as fundamental frequency and predominant melody.

Fundamental frequency The fundamental frequency (F0) is defined for periodic or nearly periodic sounds only, as the inverse of its period (Klapuri, 2003). The F0s in this thesis are extracted using the pYIN pitch estimation algorithm (Mauch and Dixon, 2014), which guarantees

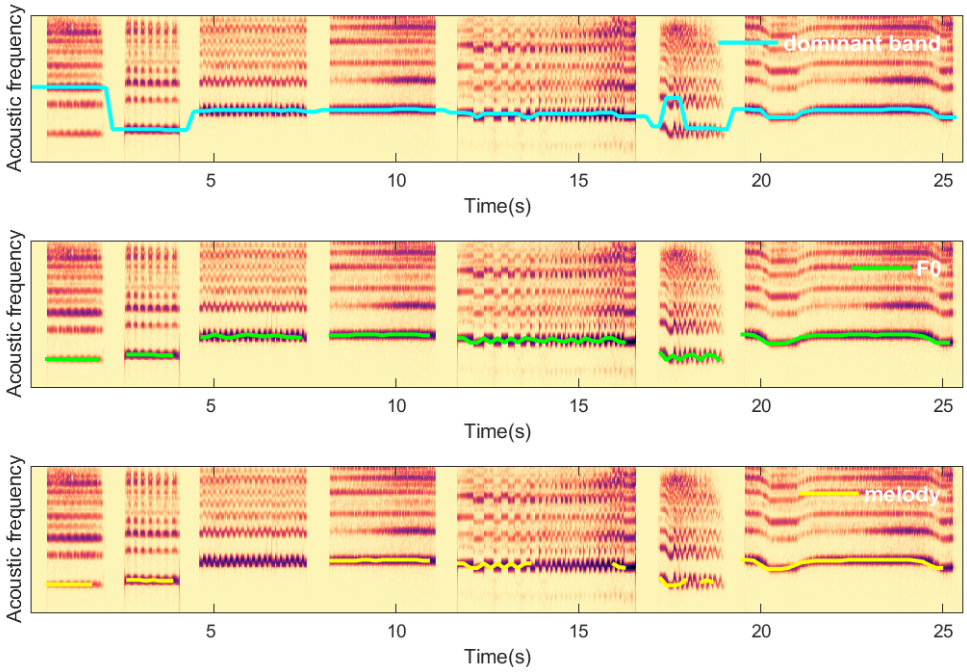


Figure 4.3: Comparison of three adaptive scattering decomposition trajectories in the scalogram of the PMT examples in Figure 4.2: dominant band, fundamental frequency (F0), and predominant melody (melody), all before preprocessing.

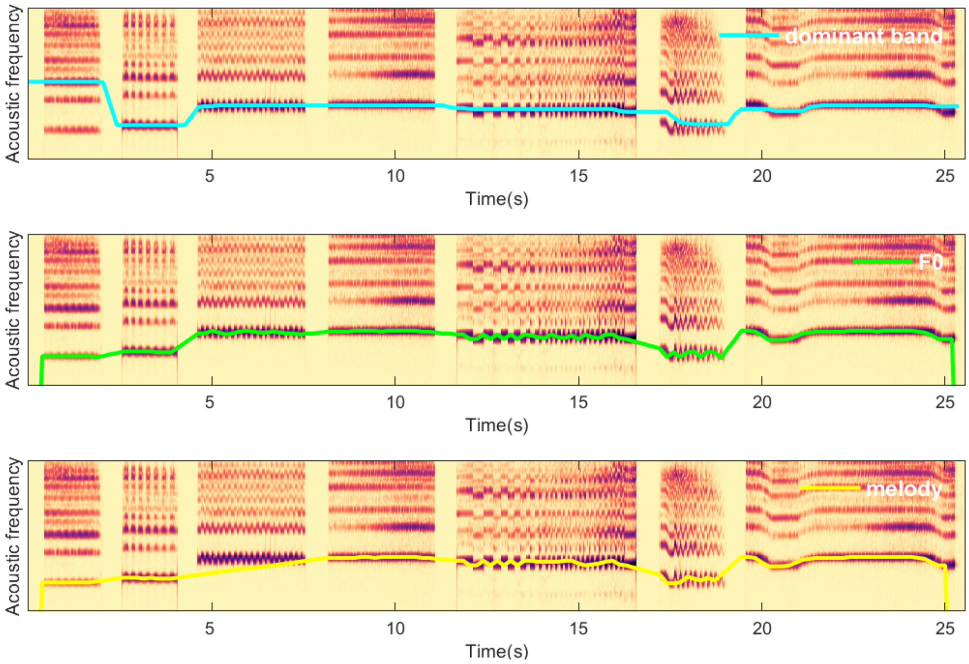


Figure 4.4: Comparison of three adaptive scattering decomposition trajectories in the scalogram of the PMT examples in Figure 4.2: dominant band, fundamental frequency (F0), and predominant melody (melody); the dominant band is preprocessed by interpolation and smoothing while the last two result from interpolation only.

temporal smoothness of the final F0 track using a hidden Markov model to track the candidates with their associated probabilities. The algorithm is commonly used for estimating the F0s of solo instrument playing or singing voice, which is the case of our dataset, comprising monophonic recordings of the Chinese bamboo flute (see Section 3.4).

Predominant melody The predominant melody refers to a sequence of frequency values corresponding to the pitch of the dominant melodic line in polyphonic music (Salamon and Gómez, 2012). The predominant melody is included here to explore its performance as a decomposition trajectory as compared to the dominant band and the F0. We extract the predominant melody using the salience-based melody estimation method proposed by Salamon and Gómez (2012). The main idea is to create pitch contours grouped from fundamental frequency candidates and then select the main contour based on a set of musical features. The features include pitch mean, pitch deviation, contour mean salience, contour total salience, contour salience deviation, contour length, and vibrato presence (true or false).

To localise F0 and predominant melody trajectories in the scalogram, we allocate the bands with the closest frequency values to both cases. Figure 4.3 middle and bottom display these two trajectories in the original form, respectively. To deal with zero frequency values and missed estimations, we also preprocess these trajectories by linear interpolation and obtain the corresponding trajectories shown in Figure 4.4 middle and bottom. We compare the performance of the adaptive scattering using the three decomposition trajectories for playing technique recognition in Subsection 4.4.5.

4.3 Playing Technique Recognition

In this section, we build an automatic system for detecting and classifying PMTs based on the proposed adaptive scattering representations: the adaptive time scattering (AdaTS), the adaptive time–rate scattering (AdaTRS), and their combination (AdaTS+AdaTRS). The system comprises four binary classifiers, each for one type of PMTs, i.e., vibrato, tremolo, trill, and flutter-tongue. This enables us to set the scattering hyperparameters according to the characteristics of each type of playing technique and to fine-tune the hyperparameters of the corresponding classifier. We investigate a multiclass classification scheme in Subsection 5.5.2 which recognises all playing techniques simultaneously.

4.3.1 Feature extraction

Figure 4.2 shows the extraction process of the adaptive scattering features of PMTs using dominant band as the example decomposition trajectory. Starting from a waveform, we first extract the decomposition trajectory from (b) the first-order time scattering transform. Localising the decomposition trajectory in (a) the scalogram and decomposing L frequency bands centred symmetrically around the dominant band trajectory, we obtain (c) the AdaTS. (d) the AdaTRS is obtained by applying frequency scattering along the modulation rate axis of (c). The AdaTS+AdaTRS is the frame-wise concatenation of (c) and (d).

Table 4.2 gives the adaptive scattering hyperparameters that capture discriminative information for PMT recognition. The averaging scale T is useful for discriminating modulations with large differences on the modulation rate, for example, distinguishing flutter-tongue from other low-rate PMTs. Averaging scales covering at least four unit patterns are recommended for reliable estimation of the modulation rate. According to the rate range of PMTs (see Table 4.1), we use $T = 2^{13}$ (in samples; corresponding to 186 ms at a sampling rate $F_s = 44.1$ kHz) for flutter-tongue, and $T = 2^{15}$ (743 ms) for the other three types of PMTs. The range M (in Hz) of the modulation rate narrows the adaptive scattering to the part that contains core information of the playing technique. An interval larger than the modulation rate range provides some harmonics in the modulation representation. For example, we set $M = [0, 150]$ Hz for flutter-tongue, and $M = [0, 100]$ Hz for vibrato, tremolo, and trill.

$Q_1^{(\mathrm{t})}$ is the filters per octave of the temporal filterbank in the first-order time scattering. Since the modulations discussed here are all oscillatory patterns, setting Q_1 should ensure that each of the modulations are not blurred in the first-order wavelet modulus transform. Here, we use $Q_1^{(\mathrm{t})} = 16$ to support subtly-modulated vibratos and tremolos, of which the modulation extent is less than one semitone. Higher $Q_1^{(\mathrm{t})}$ creates better frequency resolution but the support of the wavelets will have large overlaps in the time domain, providing less accurate temporal information and requiring higher computational cost. We set

Hyperparameter	Notation	Characteristics
Averaging scale	T	Modulation rate
Temporal filters per octave	$Q_1^{(\mathrm{t})}$	Modulation extent
Number of bands decomposed	L	$= 1$, temporal shape > 1, spectro-temporal shape
Feature dimension reduction	M	Modulation rate range

Table 4.2: Hyperparameters of the proposed adaptive scattering representations which capture discriminative information for pitch modulation-based techniques (PMTs).

$Q_1^{(\mathrm{t})} = 16$ for vibrato and tremolo detection according to a tradeoff between computational cost and accuracy. $Q_1^{(\mathrm{t})} = 12$ is applied to trill due to its note-level nature. Since the most distinct feature of flutter-tongue is the modulation rate, we set a small $Q_1^{(\mathrm{t})} = 4$ for computation saving.

L is the number of decomposed frequency bands symmetrically centred at the dominant band in the scalogram. For all modulations, we use $L = 7$ according to experimental results. $Q_2^{(\mathrm{t})}$ is the filters per octave of the temporal filter bank in the second-order time scattering. We use smaller $Q_2^{(\mathrm{t})}$ as compared to $Q_1^{(\mathrm{t})}$ due to the less oscillatory nature of the signals to be decomposed in the second-order, i.e., the frequency bands around the decomposition trajectory in the scalogram. All adaptive scattering representations operate with $Q_2^{(\mathrm{t})} = 1$ and $Q_2^{(\mathrm{t})} = 4$ filters per octave for flutter-tongue and the other three types of techniques, respectively. Besides the shared hyperparameters with the AdaTS above, the AdaTRS uses frequency scattering with $Q_1^{(\mathrm{f})} = 1$ filters per octave and an averaging scale corresponding to the entire modulation rate axis of the AdaTS.

As introduced in Section 2.5, the scattering coefficients are the results of convolving the wavelet modulus transform with a lowpass filter. The original frame size of the scattering coefficients equals the averaging scale T, i.e., 186 ms for flutter-tongue and 743 ms for the other three types of playing techniques. To compensate for the low temporal resolution resulting from the large averaging scales, we use an oversampling parameter α (Andén and Mallat, 2014) which introduces overlaps between averaging windows. The frame size h is then inversely log-proportional to the oversampling parameter by $h = T/(2^{\alpha} F_s)$. We set $\alpha = 2$ consistently for all classifiers, which results in the frame sizes for flutter-tongue, trill, vibrato, and tremolo of 46, 186, 186, and 186, respectively (all in ms), as shown in Table 4.3. Table 4.3 also lists the dimensionality of each adaptive scattering representation for each type of playing technique.

Representation	FT	Trill	Vibrato	Tremolo
AdaTS	42	133	133	133
AdaTRS	42	70	70	70
AdaTS+AdaTRS	84	203	203	203
Frame size (ms)	46	186	186	186

Table 4.3: Frame sizes and dimensionalities of the proposed adaptive scattering representations for flutter-tongue (FT), trill, vibrato, and tremolo in the binary classification scheme.

4.3.2 Recognition system

With the adaptive scattering features calculated, we build in this section a recognition system comprising four binary classifiers, each for one type of playing technique. We use support vector machines (SVMs) (Hastie et al., 2009) with Gaussian kernels as classifiers due to their good generalisability based on a limited amount of training data (Albu and Martinez, 1999). The SVM hyperparameters to be optimized during training are the error penalty parameter C and the width of the Gaussian kernel γ. We use consistent parameter grids of $2^{\{3:1:8\}}$ and $2^{\{-12:1:-7\}}$ for C and γ, respectively, for training all the classifiers, and select the best hyperparameters for testing. The classifiers take as input the adaptive scattering features and output frame-wise predictions of playing technique type. All features are z-score normalised.

In the recognition process, the CBFdataset (see Section 3.4) is split into training and test sets according to an 8:2 ratio by performers (performers are randomly initialised). We conduct 5 splits in a circular way, with no performer overlap between the test sets across splits and between the training-test sets in each split. Within each split, we run a 3-fold cross-validation, sampling on the training dataset in a way that ensures each fold includes approximately the same ratio of positive and negative class instances for a given playing technique. This is to avoid the cases that there is no instance or there are too few instances of a given playing technique class in the validation set if we further split the training set based on performer identity.

4.4 Evaluation

In this section, we present the evaluation dataset, metrics, baseline method, and results for recognising the four types of PMTs, i.e., vibrato, tremolo, trill, and flutter-tongue.

4.4.1 Dataset

To the author's knowledge, there is not yet any dataset of real-world performances with annotations covering all the four types of PMTs. We evaluate the proposed methodology on the CBFdataset introduced in Section 3.4. The CBFdataset comprises recordings of 20 full-length performances on the Chinese bamboo flute (CBF), isolated playing techniques, and annotations of seven types of playing techniques. Besides the four types of PMTs discussed in this chapter, there are also other three types of playing techniques which we will explore in Chapter 5. The total duration of the CBFdataset is 2.6 hours.

4.4.2 Metrics

Playing techniques are typically music events with certain durations. We conduct both frame- and event-based evaluation for the recognition results using the precision $\mathcal{P}$, recall $\mathcal{R}$, and F-measure $\mathcal{F}$, introduced in Subsection 2.2.3. In the frame-based evaluation, labels assigned by the classifier are compared to the ground truth in a frame-wise manner. The frame sizes are different from technique to technique. When evaluating for a specific technique over different methods, we resample the detection results to the same frame sizes that we use for CBF technique evaluation, as listed in Table 4.3.

In the event-based evaluation, we merge frame labels into events and evaluate each type of playing technique based on the onset and duration of its instances in the test set. The merging of frame labels into events is implemented using the Python library *mir_ eval* (Raffel et al., 2014). Considering the duration range of each type of playing technique, the events are postprocessed by minimum duration pruning and gap filling. We fill the gaps between neighbouring events when the gaps are shorter than the shortest event in the training set; and prune the events that have smaller duration than the minimum duration event in the training set. The minimum duration is automatically calculated subject to the technique and the training-test data split during recognition. Onsets of the obtained events are also evaluated using the Python library *mir_eval* (Raffel et al., 2014), which computes a maximum match between reference and estimated onsets, subject to a window constraint. An event is considered to be detected only when its onset falls within a 200 ms window of the ground truth and its duration is at least 50% of the ground truth duration.

4.4.3 Baseline

To the author's knowledge, there does not yet exist any general framework for detecting all the four types of PMTs in real-world performances. We compare the proposed system against the state-of-the-art approach originally proposed for vibrato detection: a system based on the filter diagonalisation method (FDM) (Yang, 2017). The system takes the frame-wise fundamental frequency estimated by the pYIN pitch detection algorithm (Mauch and Dixon, 2014) as input. The FDM feature is then computed and fed into a naive Bayes classifier for vibrato detection. Besides using the FDM as a baseline for vibrato detection in the CBFdataset (see Section 3.4), we also extend its application to the recognition of the other three types of PMTs: flutter-tongue, trill, and tremolo. To have fair comparisons, we resample the detection results into the same frame sizes as that we use for CBF playing technique recognition shown in Table 4.3. Different ranges for the hyperparameters, i.e., ranges of rate and extent, are experimented for the FDM according to PMT characteristics (see Table 4.1). The best frame-based F-measures obtained for flutter-tongue, trill, vibrato, and tremolo are 26.2%, 72.5%, 67.7%, and 38.8%; the corresponding event-based F-measures are 3.2%, 54.2%, 58.9% and 39.3%, respectively.

4.4.4 Comparative studies

Apart from the comparison with the baseline method, we also compare the performance of the proposed adaptive scattering representations using different decomposition trajectories; and compare the adaptive scattering with existing scattering representations introduced in Section 2.5 for PMT recognition: the frequency-averaged time scattering, the standard time scattering, and the standard time scattering with principal component analysis. We present each of these comparisons in the following paragraphs.

Decomposition trajectories We compare the proposed adaptive scattering representations—AdaTS, AdaTRS, and AdaTS+AdaTRS—calculated from the three decomposition trajectories introduced in Subsection 4.2.2: dominant band, F0, and predominant melody. To evaluate the performance of the preprocessing step on the dominant band trajectory, we also include the original dominant band trajectory in the comparison. The original dominant band trajectory is extracted directly from the first-order time scattering as shown in Figure 4.3 top. Replacing zero frequency values with linear interpolation and smoothing it by median filtering, we obtain the preprocessed dominant band trajectory (see Figure 4.4 top).

We implement F0 estimation using the Librosa Python package (McFee et al., 2015) and predominant melody extraction using the Essentia library (Bogdanov et al., 2013) with the same frame size of 3 ms. For these two trajectories, we fill zero frequency values and missed estimations also with a linear interpolation (see Subsection 4.2.2). To localise F0 and predominant melody trajectories in the scalogram, we allocate the bands with the closest frequency values to both cases. Since both trajectories have higher temporal resolution (3 ms) than that of the dominant band trajectory (46 ms for flutter-tongue and 186 ms for the other three playing techniques), we use the median value of a set of F0 and melody values (16 for flutter-tongue and 62 for the other three playing techniques). This enables consistent frame sizes for the adaptive scattering features based on different decomposition trajectories. We discuss the recognition results using these trajectories in Subsection 4.4.5.

Standard time scattering To evaluate the performance of the proposed adaptive scattering representations, we compare it with the standard time scattering (see Subsection 2.5.1). Since the standard time scattering has fewer hyperparameters as compared to the proposed adaptive scattering representations, we set the shared hyperparameters of the former the same to those of the latter, i.e., $T = 2^{13}, Q_1^{(\mathrm{t})} = 4, Q_2^{(\mathrm{t})} = 1$ for flutter-tongue; $T = 2^{15}, Q_1^{(\mathrm{t})} = 12, Q_2^{(\mathrm{t})} = 4$ for trill; and $T = 2^{15}, Q_1^{(\mathrm{t})} =$

$16, Q_2^{(\mathrm{t})} = 4$ for vibrato and tremolo. This results in the standard time scattering features with dimensionalities of 222, 1725, 2074, and 2074 for flutter-tongue, trill, vibrato, and tremolo, respectively; and the corresponding frame sizes are 46 ms, 186 ms, 186 ms, and 186 ms, as shown in Table 4.4. We also explore the effectiveness of the standard time scattering for identifying performers in polyphonic orchestral music in Section 6.1.

Standard time scattering with principal component analysis Considering the redundancy of the standard time scattering which decomposes all frequency bands in the scalogram, we conduct a principal component analysis (PCA) (Bishop, 2006) on the coefficients before classification and denote the resulting representation as *standard time + PCA*. This is to compare the performance of the standard time scattering with redundancy reduced in two different ways: a linear dimension reduction using PCA and a nonlinear one achieved by the adaptive operation. The latter case is the adaptive time scattering, which calculates the second-order time scattering only for a small set of frequency bands in the scalogram, i.e., the frequency bands around the decomposition trajectory. We take dimensionality into account in the representation comparison in that a representation with a lower dimensionality extracts the core characteristic information of the playing techniques but also reduces the computational cost of the classification stage, which we discuss in detail in Subsection 4.4.5.

We set the amount of variance to be explained by PCA as 95%, which results in the feature dimensionalities of 94, 603, 760, and 760 for flutter-tongue, trill, vibrato, and tremolo, respectively, as shown in Table 4.4. Although the dimensionalities are still higher than the proposed adaptive scattering representations, they are reduced by more than half as compared to the original dimensionalities of the standard time scattering. The corresponding frame sizes are the same as those of the proposed adaptive scattering representations (see Table 4.4).

Frequency-averaged time scattering Apart from the invariance to local time-shifts and time-warps, the frequency-averaged time scattering processes the second-order time scattering coefficients with a lowpass filter along the log-frequency axis (see Subsection 2.5.2), which introduces frequency-transposition invariance. Therefore, besides the averaging along the entire log-frequency axis, the frequency-averaged time scattering has the same set of hyperparameters as the standard time scattering, i.e., $T = 2^{13}, Q_1^{(\mathrm{t})} = 4, Q_2^{(\mathrm{t})} = 1$ for flutter-tongue; $T = 2^{15}, Q_1^{(\mathrm{t})} = 12, Q_2^{(\mathrm{t})} = 4$ for trill; and $T = 2^{15}, Q_1^{(\mathrm{t})} = 16, Q_2^{(\mathrm{t})} = 4$ for vibrato and tremolo. This returns the lowest feature dimensionalities among all the considered scattering representations: 10, 33, 32, and 32 for flutter-tongue, trill, vibrato, and tremolo, respectively, as shown in Table 4.4. The corresponding frame sizes are the same to those of the other scattering representations.

Representation	FT	Trill	Vibrato	Tremolo
AdaTS	42	133	133	133
AdaTRS	42	70	70	70
AdaTS+AdaTRS	84	203	203	203
Frequency-averaged	10	33	32	32
Standard time	222	1725	2074	2074
Standard time + PCA	94	603	760	760
Frame size (ms)	46	186	186	186

Table 4.4: Frame sizes and dimensionalities of the proposed adaptive scattering and the existing scattering representations for CBF flutter-tongue (FT), trill, vibrato, and tremolo.

4.4.5 Results

In this section, we analyse the recognition results of the four CBF PMTs using the adaptive scattering representations based on different decomposition trajectories and those using the existing scattering representations presented in Subsection 4.4.4. To provide an intuition of the computational cost of the recognition system, we compare the runtime of the feature extraction and classification using the trill technique as an example. The effect of SVM hyperparamters at the classification stage is also discussed.

Comparison of decomposition trajectories Table 4.5 and Table 4.6 display the frame- and event-based results of our proposed recognition system for the four types of PMTs using different decomposition trajectories: original dominant band, smoothed dominant band, F0, and predominant melody. We compare these results from four fronts: the overall performance of all decomposition trajectories, the performance of the three adaptive scattering representations (AdaTS, AdaTRS, and AdaTS+AdaTRS), the original and the smoothed dominant band trajectories, and the performance of these trajectories on different types of playing techniques. For all decomposition trajectories, the F0 achieves the best overall performance measured by the average F-measure over the four playing techniques. It yields the highest average F-measure of 73.9%, as compared to 73.5%, 72.6%, and 70.8%, the best ones obtained by the other three decomposition trajectories, i.e., original dominant band, smoothed dominant band, and predominant melody, respectively, in the frame-based evaluation. Similar trends take place in the event-based evaluation.

With regard to the comparison among the three adaptive scattering representations, the AdaTS+AdaTRS returns the highest average F-measure as compared to the AdaTS only and the AdaTRS only, although the AdaTS exhibits comparable performance as the AdaTS+AdaTRS, for both evaluation methods. This verifies our analysis in Subsection 4.2.1 that the AdaTRS provides additional information for the recognition of PMTs. Comparing the performance of the AdaTS+AdaTRS using the original and the smoothed dominant band, we notice that the former outperforms the latter for all the four playing techniques, with F-measure increase of 0.6%, 1.2%, 1.2%, and 0.4% for flutter-tongue, trill, vibrato, and tremolo, respectively, in the frame-based evaluation; and the respective F-measure improvement in the event-based evaluation are 0.8%, 2.8%, 1.2%, and 2.0%. This may be attributed to the fact that the preprocessing step also flattens short note changes with PMTs when smoothing out octave jumps. For example, it moves the trajectory below and above the true dominant bands at around 20 sec and 25 sec, respectively, in Figure 4.4 top, where the flutter-tongue techniques are both performed on two short notes. When such note changes are more frequent than the octave jumps, it is expected that the smoothing operation of the dominant band trajectory fails to improve the overall performance of the adaptive scattering representations.

Examining the performance of all trajectories on different playing techniques, we see that the original dominant band achieves the highest F-measures for flutter-tongue and trill recognition while the F0 outperforms all the other representations for vibrato and tremolo detection in the

Trajectory	Representation	FT	Trill	Vibrato	Tremolo	Average
Dominant band original	AdaTS	88.8	89.8	71.2	42.6	73.1
	AdaTRS	80.6	83.6	60.6	30.0	63.7
	AdaTS+AdaTRS	**89.2**	**90.4**	72.0	42.2	73.5
Dominant band smoothed	AdaTS	88.8	88.2	70.2	40.6	72.0
	AdaTRS	81.0	82.8	60.0	30.4	63.6
	AdaTS+AdaTRS	88.6	89.2	70.8	41.8	72.6
Fundamental frequency	AdaTS	88.2	89.6	72.2	43.8	73.5
	AdaTRS	80.8	84.4	62.0	32.4	64.9
	AdaTS+AdaTRS	88.2	90.0	**72.2**	**45.2**	**73.9**
Predominant melody	AdaTS	85.8	85.2	66.8	45.0	70.7
	AdaTRS	77.2	77.4	57.2	31.6	60.9
	AdaTS+AdaTRS	85.6	85.6	67.4	44.6	70.8

Table 4.5: Comparison of frame-based recognition results for flutter-tongue (FT), trill, vibrato, and tremolo techniques in the CBFdataset (see Section 3.4) using different decomposition trajectories: original dominant band, smoothed dominant band, fundamental frequency, and predominant melody. All numbers are F-measure scores (%). “Average” refers to the average F-measure over the four playing techniques.

Trajectory	Representation	FT	Trill	Vibrato	Tremolo	Average
Dominant band original	AdaTS	**72.0**	71.5	50.1	25.2	54.7
	AdaTRS	54.3	56.2	35.9	16.7	40.8
	AdaTS+AdaTRS	71.3	**73.3**	50.1	25.9	55.2
Dominant band smoothed	AdaTS	71.3	68.5	47.2	24.7	52.9
	AdaTRS	55.6	52.5	34.2	18.3	40.2
	AdaTS+AdaTRS	70.5	70.5	48.9	23.9	53.5
Fundamental frequency	AdaTS	70.9	72.0	51.8	27.1	55.5
	AdaTRS	55.6	60.8	38.6	17.1	43.0
	AdaTS+AdaTRS	70.1	72.9	**53.0**	31.3	**56.8**
Predominant melody	AdaTS	68.7	60.5	48.7	31.7	52.4
	AdaTRS	50.2	45.7	35.8	15.6	36.8
	AdaTS+AdaTRS	67.8	63.5	47.3	**34.4**	53.3

Table 4.6: Comparison of event-based recognition results for flutter-tongue (FT), trill, vibrato, and tremolo techniques in the CBFdataset (see Section 3.4) using different decomposition trajectories: original dominant band, smoothed dominant band, fundamental frequency, and predominant melody. All numbers are F-measure scores (%).

frame-based evaluation. This also happens to the event-based evaluation except the highest F-measure of tremolo detection which is obtained by the predominant melody trajectory. According to the performance comparison above, we use the F0 as the decomposition trajectory for the proposed adaptive scattering representations in the following sections of this chapter and in Chapter 5 due to its best overall performance.

Comparison of different representations Table 4.7 and Table 4.8 show the frame- and event-based binary classification results in terms of F-measure for flutter-tongue, trill, vibrato, and tremolo recognition using the proposed adaptive scattering, three existing scattering representations, and the FDM baseline. The adaptive scattering representations—AdaTS, AdaTRS, and AdaTS+AdaTRS—are all decomposed from the frequency bands around the F0 trajectory in the scalogram. We compare these results from four perspectives: the overall performance of all representations, recognition results of different playing techniques, the three adaptive scattering representations, and pair-wise comparison of the proposed adaptive scattering with the existing scattering representations. As can be seen, among all representations, the proposed AdaTS+ AdaTRS achieves the best overall performance in both frame- and event-based evaluation, yielding average F-measure scores of 73.9% and 56.8%, respectively. All three existing scattering representations, i.e., the frequency-averaged time scattering, the standard time scattering, and the standard time scattering with PCA, exhibit comparable overall performance and outperform the FDM baseline in terms of average F-measure score for both evaluation methods.

Comparing the recognition results of the same playing technique across different methods, one can observe that although the AdaTS+AdaTRS attains the highest average F-measure score, it does not outperform all the other representations for all the playing techniques. In the frame-based evaluation, it achieves the highest F-measures for trill and vibrato recognition while underperforming the frequency-averaged time scattering and the standard time scattering with PCA for flutter-tongue detection, and have poorer performance on tremolo detection as compared to the frequency-averaged time scattering. In the event-based evaluation, the AdaTS+AdaTRS considerably outperforms all the other representations for trill recognition. However, it underperforms

Method	FT	Trill	Vibrato	Tremolo	Average
AdaTS	88.2	89.6	72.2	43.8	73.5
AdaTRS	80.8	84.4	62.0	32.4	64.9
AdaTS+AdaTRS	88.2	**90.0**	**72.2**	45.2	**73.9**
Frequency-averaged	**92.0**	82.8	63.8	**50.8**	72.4
Standard time	91.6	85.6	65.0	48.8	72.8
Standard time + PCA	**92.0**	85.6	66.0	49.2	73.2
FDM	26.2	72.5	67.7	38.8	51.3

Table 4.7: Frame-based performance comparison of binary classification for flutter-tongue (FT), vibrato, tremolo, and trill in the CBFdataset (see Section 3.4) using the proposed adaptive scattering representations (AdaTS, AdaTRS, AdaTS+AdaTRS), three existing scattering representations (frequency-averaged time scattering, standard time scattering, standard time scattering with PCA), and the baseline method (FDM). "Average" refers to the average F-measure over all playing techniques.

Method	FT	Trill	Vibrato	Tremolo	Average
AdaTS	70.9	72.0	51.8	27.1	55.5
AdaTRS	55.6	60.8	38.6	17.1	43.0
AdaTS+AdaTRS	70.1	**72.9**	53.0	31.3	**56.8**
Frequency-averaged	83.5	33.6	22.6	16.1	39.0
Standard time	**84.5**	42.7	32.0	25.1	46.1
Standard time + PCA	84.0	44.9	34.5	24.5	47.0
FDM	3.2	54.2	**58.9**	**39.3**	38.9

Table 4.8: Event-based performance comparison of binary classifications for flutter-tongue (FT), vibrato, tremolo, and trill in the CBFdataset (see Section 3.4) using the proposed adaptive scattering representations (AdaTS, AdaTRS, AdaTS+AdaTRS), three existing scattering representations (frequency-averaged time scattering, standard time scattering, standard time scattering with PCA), and the baseline method (FDM). "Average" refers to the average F-measure over all playing techniques.

the frequency-averaged time scattering, standard time scattering, and standard time scattering with PCA, for flutter-tongue detection; and obtains lower F-measures than the FDM baseline for both vibrato and tremolo detection.

Narrowing the scope with the proposed adaptive scattering representations, the AdaTS+AdaTRS exhibits better overall performance than the AdaTS only and the AdaTRS only in both frame- and event-based evaluation. For all playing techniques, the AdaTS+AdaTRS yields comparable or better results as compared to the AdaTS while the AdaTRS returns the poorest results. This indicates that applying frequency scattering along the modulation rate axis of the AdaTS provides extra information for the recognition of PMTs only when the resulting representation is combined with the AdaTS. The AdaTRS alone exhibits high information loss.

We provide more insights on the recognition results in Table 4.7 and Table 4.8 by comparing the proposed adaptive scattering representations with the existing ones in a pair-wise manner. The first pair is the AdaTS and the frequency-averaged time scattering, two representations with only frequency-transposition invariance introduced on top of the standard time scattering. The frequency-transposition invariance of the AdaTS is achieved by the adaptive operation and that of the frequency-averaged time scattering is realised by applying frequency scattering to the second-order time scattering. We observe that the former outperforms the latter, with the average F-measure improved by 1.1% and 16.5% in the frame- and event-based evaluation, respectively. Another pair of representations in the comparison is the AdaTS and the standard time scattering with PCA, both of which are the standard time scattering with a dimensionality reduction. The dimensionality of the former is reduced by the adaptive operation while that of the latter is lowered using the PCA. As can be seen, the former beats the latter with the average F-measure increased by 0.3% and 8.5% in the frame- and event-based evaluation, respectively.

Cross checking the detection results with the original audio in the CBFdataset (see Section 3.4), we find three types of errors: rapid pitch changes, co-articulations, and play techniques that exhibit similar patterns as non-techniques. Figure 4.5 top shows the log-frequency spectrogram of an excerpt in the performance of *Morning* by Player 9 in the CBFdataset (see Section 3.4); the middle and bottom subfigures display the detected flutter-tongue events before and after postprocessing, i.e., gap filling and minimum duration pruning, and the ground truth. False negatives mostly happen at rapid note changes. The stable pitch regions are correctly detected while there are missed frames at note changes such as the gaps at 0.5 sec, 0.9 sec, and 1.2 sec. Although these three gaps are not filled due to their long duration, the F-measure score for this excerpt increases 2% after filling the events with duration less than the minimum duration event in the training set, i.e., the ones at around 0.7 sec and 2.8 sec. We discuss the other two types of errors with examples in Subsection 5.5.2.

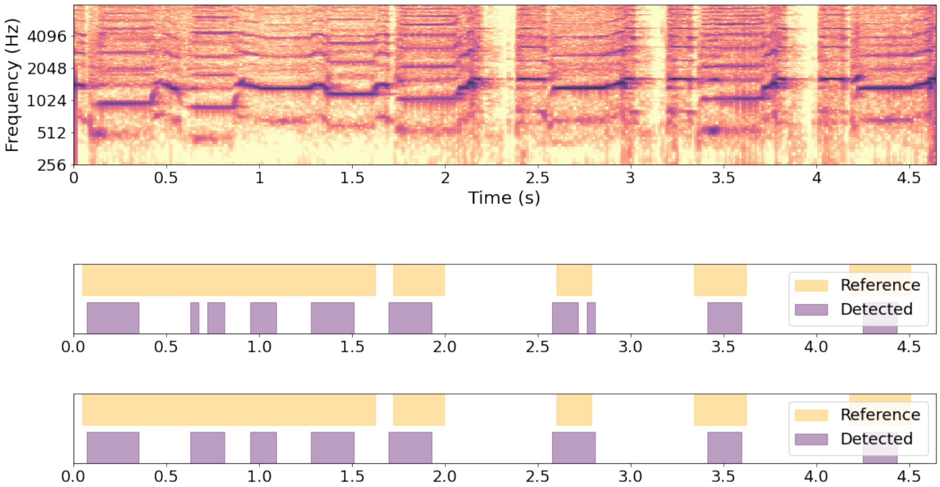


Figure 4.5: Flutter-tongue detection result for an excerpt in the performance of *Morning* by Player 9 using the proposed AdaTS+AdaTRS feature based on the F0 trajectory. Top: log-frequency spectrogram; middle: comparison between the ground truth and frame-based classification output (frame-based $\mathcal{P}$=100%, $\mathcal{R}$=55%, $\mathcal{F}$=71%); bottom: comparison between the ground truth and obtained events after gap filling and minimum duration pruning (frame-based $\mathcal{P}$=100%, $\mathcal{R}$=58%, $\mathcal{F}$=73%; event-based $\mathcal{P}$=60%, $\mathcal{R}$=64%, $\mathcal{F}$=62%). The $\mathcal{P}$, $\mathcal{R}$, $\mathcal{F}$ values above are the results on this example.

Computational cost comparison Apart from the comparison of the recognition performance, we also compare the scattering representations discussed above in terms of computational expense. Table 4.9 shows the dimensionality and the runtime (in hours) of both the feature extraction stage and the classification stage of the proposed adaptive scattering and the three existing scattering representations for trill recognition on the same machine. The representations are listed by an ascending order of their dimensionalities. It is expected that the extraction of the three existing scattering representations—frequency-averaged time scattering, standard time scattering with PCA, and standard time scattering—exhibit similar runtimes. This is because the first two representations are both derived from the standard time scattering by additional operations with low computational cost, i.e., averaging along the log-frequency axis and PCA. For the proposed AdaTS+AdaTRS, ideally, the feature extraction computation is lower than that of the standard time scattering since the former only decomposes L out of λ frequency bands from the scalogram while the latter decomposes all the λ frequency bands. The longest runtime of 5.73 h obtained (see Table 4.9) for extracting the AdaTS+AdaTRS may be attributed to our current implementation which is different from that of the standard scattering transform calculation. We regard implementation optimisation for the adaptive scattering feature extraction as future work.

Representation	Dimensionality	Runtime (hours)		
		Extraction	Classification	Total
Frequency-averaged	33	4.50	0.08	4.58
AdaTS+AdaTRS	203	5.73	0.31	6.04
Standard time+PCA	603	4.61	1.20	5.81
Standard time	1725	4.61	3.04	7.65

Table 4.9: Dimensionality, feature extraction runtime, classification runtime, and total runtime of the proposed adaptive scattering and the three existing scattering representations for trill recognition.

With regard to the runtime of the classification stage, we notice that it positively correlates with the representation dimensionality. The frequency-averaged scattering which has the lowest feature dimensionality (33) obtains the shortest runtime (0.08 h) while the standard time scattering with the highest dimensionality (1725) exhibits the longest runtime (3.04 h). Therefore, for the proposed recognition system, a lower dimensional representation is preferable when there is not much degradation on the recognition performance.

Effect of SVM hyperparameters To investigate the influence of SVM hyperparameters on the recognition performance, we plot the F-measure scores obtained by different combinations of the error penalty parameter C and the width of the Gaussian kernel γ from the hyperparameter grids $C \in 2^{\{3:1:8\}}$ and $\gamma \in 2^{\{-12:1:-7\}}$ (see Subsection 4.3.2). Figure 4.6 shows the F-measure scores yielded on the training set during the cross-validation in the first split for trill recognition using the AdaTS+AdaTRS, where the recordings of performers 4 and 8 construct the test set and the remaining recordings form the training set. One may observe that the results obtained by these hyperparameter grids are relatively stable. The combination of $C = 8$ and $\gamma = 2^{-7}$ generates the highest averaged F-measure of 94% for trill recognition during the cross-validation process on the training set and the corresponding F-measure on the test set is 96%.

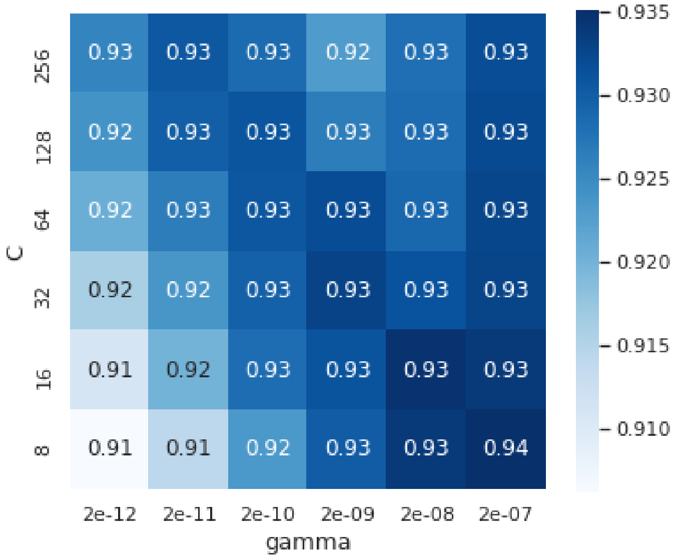


Figure 4.6: Effect of SVM hyperparameters on the training set during cross-validation (numbers are F-measure scores). The F-measure score on the test set is 96% with $C = 8$ and $\gamma = 2^{-7}$.

4.5 Conclusions

In this chapter, we have proposed a variant of the scattering transform, the adaptive scattering, which provides representations that are compact and invariant to large frequency-transpositions for pitch modulation-based techniques (PMTs). Three adaptive scattering representations, i.e., the adaptive time scattering (AdaTS), the adaptive time–rate scattering (AdaTRS), and the combination of these two operators (AdaTS+AdaTRS), are investigated. We compare the performance of each representation using three decomposition trajectories: dominant band, fundamental frequency (F0), and predominant melody. To mitigate the missed estimations and instabilities of the decomposition trajectories, we preprocess them with linear interpolation (and smoothing). Comparing the three preprocessed trajectories and the original dominant band trajectory, the F0 exhibits the best overall performance and is used as the decomposition trajectory for the final recognition systems in this Chapter and in Chapter 5 for representing PMTs.

To gain more insights on the adaptive scattering, we also compare its performance on PMT recognition with that of three existing scattering representations: frequency-averaged time scattering, standard time scattering, and standard time scattering with principal component analysis. Once trained on the proposed adaptive scattering representations, support vector machine (SVM) classifiers achieve comparable or better results on PMT recognition in the CBFdataset (see Section 3.4) as compared to the existing scattering representations and the filter diagonalisation baseline. Besides the recognition performance, we further compare the scattering representations in terms of computational expense. The findings are that the extraction of the adaptive scattering exhibits longer runtime than the existing scattering representations for the current implementation and that a lower dimensionality reduces the computational cost of the classification stage.

Despite the promising results obtained, there are some limitations regarding the work conducted in this chapter. Firstly, we have proposed the adaptive scattering in this chapter for recognising PMTs in monophonic music, which is based on the assumption that the instrument has PMTs in its playing technique repertoire, i.e., purely harmonic instruments or pitched percussive instruments. In both cases, we could form a decomposition trajectory when the pitch is stable. For pitched percussive instruments such as piano and guitar, the dominant band strategy may pick up some irrelevant bands to the fundamental frequency or its harmonics at the transient stage, whichever is short as compared to the sustain and decay stages. This assumption may be violated in the case of polyphonic music, which we will explore in Section 6.1.

On the other hand, the hyperparameters of the scattering transform are not comprehensively tuned together with those of the classifiers. There are two sets of hyperparameters in our proposed recognition system, the scattering transform hyperparameters and the SVM hyperparameters. We tuned the latter using a validation set in the SVM training process. For the former, we use hyperparameters motivated by music theory, which offers clear information on what characteristics of each playing technique have been captured by the proposed scattering representations. Tuning the hyperparameters of the scattering transform and the SVM jointly would potentially improve the recognition results.

Potential feature work includes possible applications of the proposed playing technique recognition system and further investigations of the proposed adaptive scattering representations. Section 6.1 of this thesis is an example in the first direction, in which we propose a performer identification system using vibratos detected by the playing technique recognition system proposed in this chapter. The local invariance to time-shifting, time-warping, and frequency-transposition of the adaptive scattering transforms may also be attractive to other music signal analysis tasks, such as music structure analysis, genre recognition, instrument recognition, and music transcription. Motivated by the observation in Figure 4.2 that the second-order scattering transform carries information on the modulation rate, one may use the scattering transform as a tool for playing technique modelling.

Playing technique recognition and modelling may also greatly help music synthesis systems generate realistic sounds that account for acoustic variations due to the exercise of a variety of instrumental or vocal techniques. A music style transformer (Dai et al., 2018) or note ornamentor is also possible since playing techniques carry important information regarding musical styles. Remodelling a straight note based on a playing technique or articulation of a professional player or synthesising playing techniques that go beyond real instrument limitations present other attractive directions for further exploration, for example creating a flutter-tongue effect for piano.

Chapter 5

Joint Scattering for Pitch Evolution-based Technique Recognition

In Chapter 4, we proposed the adaptive scattering for pitch modulation-based techniques (PMTs). These playing techniques include vibratos, tremolos, trills, and flutter-tongue, all exhibiting periodic modulations in the time–frequency domain. As with the other group of playing techniques introduced in Section 3.2, the pitch evolution-based techniques (PETs) which contain monotonic pitch changes, the adaptive scattering may not be applicable. The adaptive scattering decomposes only a small set of frequency bands of the scalogram, which loses the spectral variation information that are important to PET recognition. In this chapter, we modify the joint time–frequency scattering (JTFS) into a direction-invariant representation, the dJTFS, for PET recognition. Two recognition systems are built: one detects PETs using binary classifiers with the hyperparameters fine-tuned for each type of playing technique; the other classifies PMTs and PETs simultaneously based on a multiclass classification scheme which takes the concatenation of the adaptive scattering and the dJTFS coefficients as input. Both systems are evaluated on the CBFdataset (see Section 3.4). To test the generalisability of the proposed methodology, we further verify the system over three additional datasets with a variety of instrumental and vocal techniques.

We introduce the characteristics of PETs in Section 5.1. Section 5.2 presents the dJTFS and gives a different interpretation of the JTFS as compared to the original one defined in Subsection 2.5.2. Two recognition systems are then built in Section 5.3, one with binary classifiers and the other using a multiclass classifier. Section 5.4 proposes a baseline method for glissando detection, followed by evaluation of both recognition systems in Section 5.5. Section 5.6 tests the generalisability of the proposed methodology on three additional datasets and Section 5.7 concludes the chapter. This chapter is extended from the associated publications Wang et al. (2020a) and Wang et al. (submitted) except Section 5.4, which was published in Wang et al. (2019b) (see Section 1.4 for publication details).

5.1 Characteristics of Pitch Evolution-based Techniques

Table 5.1 displays the characteristic information of acciaccatura, portamento, and glissando, three typical examples of PETs we consider in this chapter. The duration range is derived from the CBFdataset (see Section 3.4), while the other three characteristics are based on the musical definition of the playing techniques (see Section 3.2). Each type of these playing techniques has a specific duration range: 0.1-0.4 sec for acciaccatura, 0.2-1.2 sec for portamento, and 0.2-1.1 sec for glissando. For temporal variations, although all three types of playing techniques contain monotonic pitch changes over time, portamento exhibits smooth pitch changes while the pitch changes within acciaccatura and glissando are both at the note level. Acciaccatura contains only one note change, while glissando spans a series of note changes. For spectral variations, acciaccatura has a noisy attack while glissando and portamento exhibit clear harmonic structures. The possible directions of their pitch changes are different: acciaccatura in Chinese bamboo flute (CBF) playing only occurs downwards, while the other two techniques can exhibit both upward and downward directions.

5.2 Direction-invariant Joint Time–Frequency Scattering

Different from the separable scattering (see Subsection 2.5.2) which calculates time and frequency scattering in separable steps, the joint time–frequency scattering (JTFS) applies them jointly. The interaction

Characteristics	Acciaccatura	Portamento	Glissando
Duration (s)	0.1-0.4	0.2-1.2	0.2-1.1
Temporal variation	One note change	Smooth pitch changes	Consecutive note changes
Spectral variation	Noisy attack	Harmonic	Harmonic
Pitch direction	↘	↗ or ↘	↗ or ↘

Table 5.1: Characteristics of pitch evolution-based techniques (PETs).

of the two types of wavelet convolutions captures the joint activation of temporal and spectral variations. Motivated by the recognition task for PETs, we interpret the definition of the JTFS introduced in Subsection 2.5.2 from a new perspective. Rather than formulating a two-dimensional (2-D) mother wavelet, we consider the temporal and spectral wavelet convolutions in a sequential manner. This is more precise in terms of the computations performed and provides explicit information of what has been captured at each step.

Following the notations in Section 2.5, we denote $\boldsymbol{\psi}(t)$ and $\boldsymbol{\psi}(\lambda)$ as mother wavelets along the time and the log-frequency axes, respectively, with time $t \in \mathbb{R}$ and log-frequency variable $\lambda \in \mathbb{R}$. $\boldsymbol{\psi}_{v_\mathrm{t}}(t)$ and $\boldsymbol{\psi}_{v_\mathrm{f}}(\lambda)$ are temporal and spectral wavelet filterbanks, dilated from the mother wavelets $\boldsymbol{\psi}(t)$ and $\boldsymbol{\psi}(\lambda)$ by the respective scaling factors 2^{-v_t} and 2^{-v_f}. v_t and v_f are the log-frequency variables of $\boldsymbol{\psi}_{v_\mathrm{t}}(t)$ and $\boldsymbol{\psi}_{v_\mathrm{f}}(\lambda)$, and measures the temporal and spectral variabilities, respectively. An orientation variable $\theta = \pm 1$ is introduced to reflect the oscillation direction (up or down) of the spectro-temporal pattern. $\theta = -1$ flips the centre frequency of wavelet $\boldsymbol{\psi}(\lambda)$ from 2^λ to -2^λ. The resulting temporal and spectral wavelet filterbanks are respectively:

$$\boldsymbol{\psi}_{v_\mathrm{t}}(t) = 2^{v_\mathrm{t}}\boldsymbol{\psi}(2^{v_\mathrm{t}}t) \quad \text{and} \tag{5.1}$$

$$\boldsymbol{\psi}_{v_\mathrm{f},\theta}(\lambda) = 2^{v_\mathrm{f}}\boldsymbol{\psi}(\theta 2^{v_\mathrm{f}}\lambda). \tag{5.2}$$

The joint wavelet transform of $\mathbf{X}(t, \lambda)$ computes convolutions of the form:

$$\Big((\mathbf{X} \overset{t}{*} \boldsymbol{\psi}_{v_\mathrm{t}}) \overset{\lambda}{*} \boldsymbol{\psi}_{v_\mathrm{f},\theta}\Big)(t,\lambda) = \Big(\mathbf{X} \overset{t,\lambda}{*} (\boldsymbol{\psi}_{v_\mathrm{t}} \otimes \boldsymbol{\psi}_{v_\mathrm{f},\theta})\Big)(t,\lambda), \tag{5.3}$$

where the operator $\otimes$ denotes the outer product between two one-dimensional (1-D) wavelets, returning a 2-D wavelet. In practice, we implement the joint time–frequency convolution via the left-hand side of the equation above, that is, by a sequence of two 1-D convolutions. This two-step factorised procedure is more efficient than the one-step 2-D convolution, described on the right-hand side. However, the right-hand side of Eq. (5.3) is useful for the theoretical understanding of the JTFS as involving a joint convolutional operator in the time–frequency domain. Indeed, we may view the outer product between the temporal wavelet $\boldsymbol{\psi}_{v_{\mathrm{t}}}(t)$ and the spectral wavelet $\boldsymbol{\psi}_{v_{\mathrm{f}},\theta}(\lambda)$ as the factorisation of a joint time–frequency wavelet

$$\boldsymbol{\Psi}_{v_{\mathrm{t}},v_{\mathrm{f}},\theta}(t,\lambda) = \boldsymbol{\psi}_{v_{\mathrm{t}}}(t)\boldsymbol{\psi}_{v_{\mathrm{f}},\theta}(\lambda), \tag{5.4}$$

which captures the local spectro-temporal modulations of $\mathbf{X}(t,\lambda)$ around time t and log-frequency λ in terms of the temporal variability v_{t}, the spectral variability v_{f}, and the orientation θ.

For a specific recognition task at hand, we typically focus on a spectro-temporal pattern smaller than a "time–frequency box" restricted by some time scale T in samples and frequency interval F in octaves. To ensure local time-shifting invariance, time-warping stability, frequency-transposition invariance, and frequency-warping stability, we take the modulus of the output of Eq. (5.3) and average it by a 2-D lowpass filter $\boldsymbol{\Phi}_{T,F}(t,\lambda)$. Following Andén et al. (2019), we define the *joint time–frequency scattering* of $\mathbf{X}(t,\lambda)$ according to Eqs. (5.3) and (5.4) as:

$$\mathbf{S}_2^{\mathrm{JTFS}}\boldsymbol{x}(t,\lambda,v_{\mathrm{t}},v_{\mathrm{f}},\theta) = \left(\left|\mathbf{X} \overset{t}{*} \boldsymbol{\psi}_{v_{\mathrm{t}}} \overset{\lambda}{*} \boldsymbol{\psi}_{v_{\mathrm{f}},\theta}\right| \overset{t,\lambda}{*} \boldsymbol{\Phi}_{T,F}\right)(t,\lambda). \tag{5.5}$$

A diagram of the JTFS calculation process for a glissando example is shown in Figure 5.1. (b) is the temporal wavelet transform (temporal WT) operation, i.e., convolving (a) the scalogram $\mathbf{X}(t,\lambda)$ with $\boldsymbol{\psi}_{v_{\mathrm{t}}}(t)$. The obtained wavelet transform $\mathbf{X} \overset{t}{*} \boldsymbol{\psi}_{v_{\mathrm{t}}}(t)$ mainly captures the temporal variations of each frequency band. To capture correlations across frequency bands, we apply wavelet convolutions with $\boldsymbol{\psi}_{v_{\mathrm{f}},\theta}(\lambda)$ along the log-frequency axis of $\mathbf{X} \overset{t}{*} \boldsymbol{\psi}_{v_{\mathrm{t}}}(t)$, which is (c) the spectral wavelet transform (spectral WT) operation, and obtain $\mathbf{X} \overset{t}{*} \boldsymbol{\psi}_{v_{\mathrm{t}}} \overset{\lambda}{*} \boldsymbol{\psi}_{v_{\mathrm{f}},\theta}$. Taking complex modulus of (c) and averaging it by the lowpass filter $\boldsymbol{\phi}_{T,F}(t,\lambda)$, we derive (d) the JTFS, $\mathbf{S}_2^{\mathrm{JTFS}}\boldsymbol{x}(t,\lambda,v_{\mathrm{t}},v_{\mathrm{f}},\theta)$. According to Eq. (5.5), for each "time–frequency" box around (t,λ), the $\mathbf{S}_2^{\mathrm{JTFS}}\boldsymbol{x}(t,\lambda,v_{\mathrm{t}},v_{\mathrm{f}},\theta)$ is a three-dimensional tensor with respect to $(v_{\mathrm{t}}, v_{\mathrm{f}}, \theta)$, which captures the joint activation of temporal and spectral variations, and its direction, as shown in (d). The energy of the JTFS concentrates on the $\theta = -1$ side due to the upward direction of the glissando example.

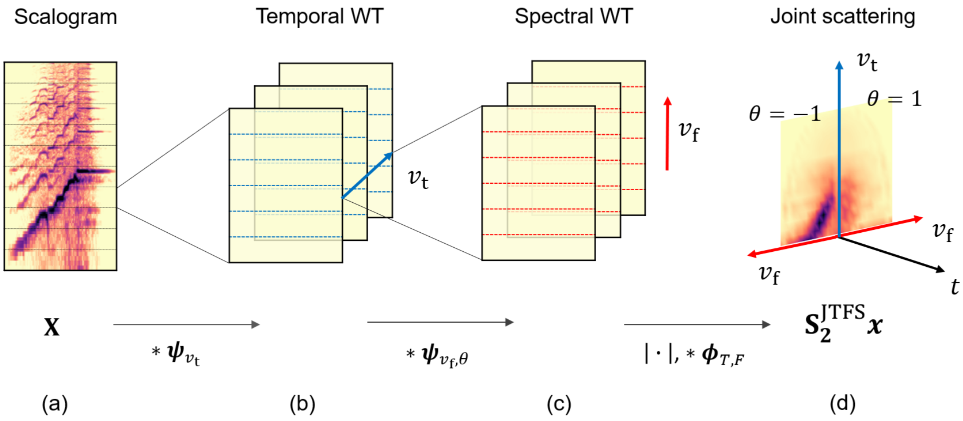


Figure 5.1: Calculating the joint time–frequency scattering (JTFS) for a glissando example: (a) scalogram; (b) temporal wavelet transform (temporal WT) operation by convolving with temporal filterbank $\boldsymbol{\psi}_{v_{\mathrm{t}}}$; (c) spectral wavelet transform (spectral WT) by applying spectral filterbank $\boldsymbol{\psi}_{v_{\mathrm{f}},\theta}$; (d) the JTFS, result of complex modulus and averaging with a 2-D lowpass filter $\boldsymbol{\phi}_{T,F}$.

Figure 5.2 compares the JTFS of acciaccatura, portamento, and glissando: (a) is the spectrogram; (b), (c), and (d) are the 2-D joint activations for each respective type of PET. Here we use a spectrogram in (a) rather than a scalogram to clearly visualise the spectro-temporal patterns of the playing techniques. As observed, although both acciaccatura and glissando have high-energy regions in the JTFS, their energy distributions along the variation scales are different. From (b) and (d), noisy attacks show as diffused energy in the JTFS, and the time and frequency regularity of glissando results in clear slopes.

As discussed in Section 5.1, each PET exhibits one direction of pitch change, while according to Eq. (5.5), we obtain information for both directions. For recognising the type of PETs only, we modify Eq. (5.5) into the *direction-invariant joint time-frequency scattering* (dJTFS), which introduces a pooling operation on the direction variable θ of the JTFS. This can be either a max-pooling or an average-pooling. We define the former case as *dJTFS-max*, extracting only the JTFS coefficients

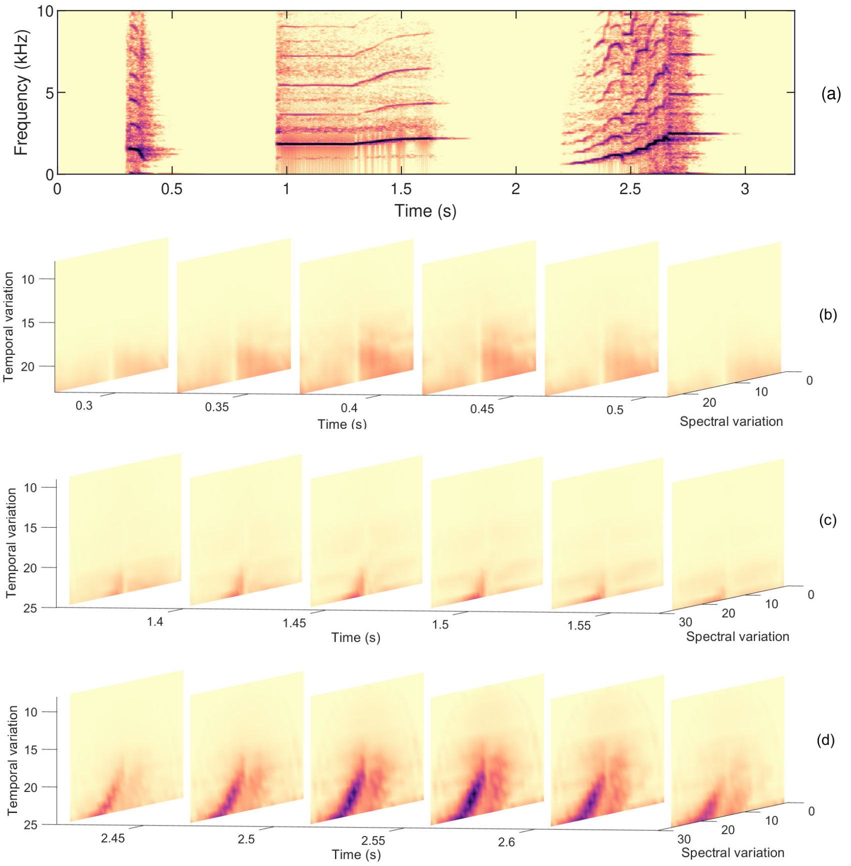


Figure 5.2: Visualisation of the joint time–frequency scattering (JTFS) for pitch evolution-based techniques (PETs). (a) Spectrogram showing acciaccatura, portamento, and glissando; (b), (c), and (d) are the corresponding JTFS plots for each case.

corresponding to $\theta_{\max}(t)$:

$$\mathbf{S}_2^{\mathrm{dJTFS-max}}\boldsymbol{x}(t,\lambda,\upsilon_{\mathrm{t}},\upsilon_{\mathrm{f}}) = \mathbf{S}_2^{\mathrm{JTFS}}\boldsymbol{x}(t,\lambda,\upsilon_{\mathrm{t}},\upsilon_{\mathrm{f}},\theta_{\max}), \tag{5.6}$$

where $\theta_{\max}(t)$ is the direction with maximum spectro-temporal energy:

$$\theta_{\max}(t) = \arg\max_{\theta=1,-1}\Big(\sum_{\lambda,\upsilon_{\mathrm{t}},\upsilon_{\mathrm{f}}}\mathbf{S}_2^{\mathrm{JTFS}}\boldsymbol{x}(t,\lambda,\upsilon_{\mathrm{t}},\upsilon_{\mathrm{f}},\theta)\Big). \tag{5.7}$$

In the latter case, we average the JTFS coefficients over both directions, i.e., $\theta = 1$ and $\theta = -1$, and define the resulting representation as *dJTFS-avg*:

$$\mathbf{S}_2^{\mathrm{dJTFS-avg}}\boldsymbol{x}(t,\lambda,\upsilon_{\mathrm{t}},\upsilon_{\mathrm{f}}) = \frac{1}{2}\sum_{\theta=1,-1}\mathbf{S}_2^{\mathrm{JTFS}}\boldsymbol{x}(t,\lambda,\upsilon_{\mathrm{t}},\upsilon_{\mathrm{f}},\theta). \tag{5.8}$$

We compare the performance of dJTFS-max and dJTFS-avg on PET recognition in Subsection 5.5.1.

Similarly to Section 4.2, we normalise $\mathbf{S}_2^{\mathrm{dJTFS}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f})$, i.e., either $\mathbf{S}_2^{\mathrm{dJTFS-max}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f})$ or $\mathbf{S}_2^{\mathrm{dJTFS-avg}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f})$, over the first-order time scattering coefficients $\mathbf{S}_1\boldsymbol{x}(t,\lambda)$ to capture only the temporal and spectral variations regardless of the absolute energy of the waveform. We then take the logarithm of the normalised coefficients (Andén and Mallat, 2014) to mimic auditory perception (see Subsection 2.5.1) and derive the *log-normalised dJTFS*:

$$\widetilde{\mathbf{S}}_2^{\mathrm{dJTFS}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f}) = \log_2\left(\frac{\mathbf{S}_2^{\mathrm{dJTFS}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f})}{\mathbf{S}_1\boldsymbol{x}(t,\lambda)+\varepsilon}\right), \tag{5.9}$$

where ε is a small additive offset whose role is to avoid division by zero. Since $\widetilde{\mathbf{S}}_2^{\mathrm{dJTFS}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f})$ is simply a log-normalisation of $\mathbf{S}_2^{\mathrm{dJTFS}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f})$, thereafter we also refer to the former, the one we actually use in the experiments of this thesis, as the dJTFS.

5.3 Playing Technique Recognition

To develop a general framework for recognising playing techniques, we investigate two classification schemes in this chapter:

1. A recognition system with three binary classifiers, each detecting one type of PET, which is similar to the recognition system proposed in Section 4.3 for recognising PMTs. Each classifier takes as input the dJTFS coefficients with the hyperparameters fine-tuned for each type of playing technique.

2. A recognition system with a multiclass classifier which detects all seven playing techniques simultaneously: vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando. The first four are PMTs explored in Chapter 4 and the last three are PETs discussed in this chapter. The system takes the concatenation of the adaptive scattering and the dJTFS coefficients as input.

The binary classification scheme uses features with a lower dimensionality and is capable of detecting co-articulations, such as the combination of tremolo and trill, or glissando co-articulated with flutter-tongue. The multiclass one provides a confusion matrix between playing techniques and yields results that are comparable across different methods for different datasets.

5.3.1 Feature extraction

For the binary classification scheme, we set the hyperparameters of the dJTFS representations according to the characteristics of each type of PET, as shown in Table 5.2. The averaging scale T carries duration information via setting T equivalent to the maximum duration of each playing technique. According to the duration range of PETs in Table 5.1, we use $T = 2^{13}$ (in samples; corresponding to 186 ms at a sampling rate of $F_s = 44.1$ kHz) for acciaccatura, and $T = 2^{14}$ (372 ms) for portamento and glissando. The temporal filters per octave in the first-order time scattering $Q_1^{(\mathrm{t})}$ are useful for distinguishing note changes from smooth pitch changes. For acciaccatura and glissando, we set $Q_1^{(\mathrm{t})} = 12$ due to their note-change property. To capture the smooth pitch changes within portamento, $Q_1^{(\mathrm{t})} > 12$ is required. Higher $Q_1^{(\mathrm{t})}$ creates better frequency resolution while the support of the wavelets would have large overlaps in the time domain, providing less accurate temporal information. We set $Q_1^{(\mathrm{t})} = 16$ for portamento detection according to a tradeoff between computational cost and accuracy. We use $Q_2^{(\mathrm{t})} = 2$ filters per octave in the second-order time scattering due to the less oscillatory nature of audio signals at this order of decomposition.

Hyperparameter	Notation	Characteristics
Averaging scale	T	Duration
Temporal filters per octave	$Q_1^{(\mathrm{t})}$	Pitch change
Spectral filters per octave	$Q_1^{(\mathrm{f})}$	Spectral variation
Orientation variable	θ	Direction of pitch change
Modulation range	M	Dimension reduction

Table 5.2: Hyperparameters of the proposed direction-invariant joint time–frequency scattering (dJTFS) which capture discriminative information for pitch evolution-based techniques (PETs).

One may observe from Figure 5.2 (a) the different harmonic structures between the selected PETs. This timbral information can be captured by applying frequency scattering with $Q_1^{(\mathrm{f})}$ spectral filters per octave along the log-frequency axis. The spectral averaging scale, F (in octave units), depends on the frequency-transposition invariance requirement of the task. Here we use $Q_1^{(\mathrm{f})} = 2$ filters per octave and a spectral averaging scale covering the entire log-frequency axis. To extract coefficients which contain most of the spectro-temporal modulation energy, we narrow the coefficients corresponding to a temporal modulation range of $M = [0, 50]$ Hz. We then obtain the dJTFS feature of PETs for each time frame according to Eqs. (5.6) or (5.8), depending on the direction-invariance strategy used.

The evolutionary nature of PETs suggests the importance of temporal context. Here we calculate mean and standard deviation of 5 frames centred at the current frame to represent context information. Similarly to the recognition of PMTs in Section 4.3, we set the oversampling parameter $\alpha = 2$ consistently for all classifiers, which results in the frame sizes of 46 ms, 93 ms, and 93 ms, for acciaccatura, portamento, and glissando recognition, respectively. The corresponding feature dimensionalities are 366, 410, and 498, as shown in Table 5.3.

Representation	Acciaccatura	Portamento	Glissando
dJTFS	366	410	498
JTFS	1490	1486	1770
JTFS+PCA	67	23	63
Frame size (ms)	46	93	93

Table 5.3: Dimensionalities and frame sizes of the proposed dJTFS, the JTFS, and the JTFS+PCA for acciaccatura, portamento, and glissando in the binary classification scheme.

We also compare the dJTFS with the JTFS for recognising PETs. The JTFS is computed using the same hyperparameters as that we use for dJTFS calculation, i.e., $T = 2^{13}, Q_1^{(\mathrm{t})} = 12$ for acciaccatura; $T = 2^{14}, Q_1^{(\mathrm{t})} = 16$ for portamento; $T = 2^{14}, Q_1^{(\mathrm{t})} = 12$ for glissando; and $Q_2^{(\mathrm{t})} = 2, Q_2^{(\mathrm{f})} = 2, \alpha = 2$ and a spectral averaging scale covering the entire log-frequency axis for all three playing techniques. Therefore for a specific playing technique, the frame size of the JTFS is the same as that of the dJTFS; while the dimensionality of the former is over 3 times that of the latter, as shown in Table 5.3. Similarly to Subsection 4.4.4, we also apply principal component analysis (PCA) to the JTFS before feeding it into the classifier and denote the resulting representation as JTFS+PCA. The PCA operation significantly reduces the dimensionalities of the JTFS features, leading to dimensionalities of

67 for acciaccatura, 23 for portamento, and 63 for glissando, which are also lower than those of the dJTFS for the corresponding techniques (see Table 5.3). We show in Subsection 5.5.1 how the dimensionality influences the computational cost of the classification stage.

As with the multiclass classification scheme, we use the concatenation of the proposed AdaTS+AdaTRS (see Section 4.2) and dJTFS features as the input. We calculate the AdaTS+AdaTRS using one set of hyperparameters: $T = 2^{15}$, $Q_1^{(\mathrm{t})} = 16$, $Q_2^{(\mathrm{t})} = 4$, $Q_1^{(\mathrm{f})} = 1$, $\alpha = 2$, $M = [0, 100]$ Hz, and an averaging scale covering the entire modulation rate axis for the AdaTRS component; the dJTFS is computed with another set of hyperparameters: $T = 2^{14}$, $Q_1^{(\mathrm{t})} = 16$, $Q_2^{(\mathrm{t})} = 2$, $Q_1^{(\mathrm{f})} = 2$, $\alpha = 2$, $M = [0, 50]$ Hz, and a spectral averaging scale covering the whole log-frequency axis. Due to the different averaging scales, $T = 2^{15}$ for the AdaTS+AdaTRS and $T = 2^{14}$ for the dJTFS, we duplicate the AdaTS+AdaTRS before concatenation to have the same number of frames as the dJTFS, with a frame size of $h = T/(2^{\alpha}F_s) = 93$ ms. The dimensionality of the concatenated feature is 613, higher than those in the binary classification schemes displayed in Table 4.3 and Table 5.3.

5.3.2 Recognition system

For both classification schemes, we use support vector machines (SVMs) (Hastie et al., 2009) with Gaussian kernels as classifiers due to their good generalisability based on a limited amount of training data (Albu and Martinez, 1999). The SVM hyperparameters to be optimized are the error penalty parameter C and the width of the Gaussian kernel γ. We use consistent parameter grids of $2^{\{3:1:8\}}$ and $2^{\{-12:1:-7\}}$ for C and γ, respectively, for the classification of all playing techniques during training, and select the best ones for testing. The classifiers take as input the JTFS or the concatenation of the AdaTS+AdaTRS and the JTFS, and output frame-wise predictions of playing technique type. All features are z-score normalised.

In the recognition process, the CBFdataset (see Section 3.4) is split into training and test sets according to an 8:2 ratio by performers (performers are randomly initialised). We conduct 5 splits in a circular way, with no performer overlap between the test sets across splits and between the training-test sets in each split. Within each split, we run a 3-fold cross-validation, sampling on the training dataset in a way that ensures each fold includes approximately the same ratio of positive and negative class instances for a given playing technique. This is to avoid the cases that there is no example or there are too few examples of a given playing technique class in the validation set if we further split the training set based on performer identity.

5.4 Baselines

5.4.1 Baselines for binary classification scheme

To the author's knowledge, there is not yet any prior work on the recognition of all three types of PETs, i.e., acciaccatura, portamento, and glissando. In the binary classification scheme, we compare the proposed dJTFS approach with the state-of-the-art method introduced by Yang (2017) for detecting portamenti. The method is based on hidden Markov models (HMMs) (Rabiner, 1989; Xing et al., 2010) and takes the frame-wise fundamental frequency (F0) estimated by the pYIN pitch detection algorithm (Mauch and Dixon, 2014) as input. We also extend the application of this method to acciaccatura recognition. For fair comparisons, we resample the detection results of this method into the same frame sizes as that we use for CBF acciacatura and portamento detection based on the dJTFS features, i.e., 46 and 93 ms, as shown in Table 5.3. Frame-based F-measures of 25.0% and 30.0%, and event-based F-measures of 23.5% and 22.4% are obtained for acciacatura and portamento detection in the CBFdataset (see Section 3.4).

Motivated by the consecutive note changes within glissandi, we propose in this thesis a baseline method for glissando detection which is also based on HMMs. The proposed glissando detection method includes two stages:

1. Rule-based segmentation: A set of rules informed by the characteristics of glissandi (see Section 5.1) is introduced to extract segments with consecutive note changes in the same direction as glissando candidates.

2. HMM-based identification: A glissando HMM (G-HMM) is trained using all ground truth glissandi in the training set. Different from traditional binary classification, the false positives of the segmentation stage, which exhibit similar pitch evolution and

duration as the ground truth, are used to train a non-glissando HMM (NG-HMM). Glissando candidates in the test set are finally identified by the two HMMs.

HMMs enable the decoding of note evolution while smoothing outlier variations within glissandi. We introduce each stage in detail.

Rule-based segmentation

To obtain glissando candidates from the full-piece recordings, we introduce a rule-based segmentation stage using F0 with a hop size of 6 ms as input (see Figure 5.3). The pitch is first smoothed to exclude noisy variations and quantised to the nearest notes in 12-tone equal temperament scale, resulting in 16 notes in the CBF tonal range: G4-A6 for the C flute, and D5-E7 for the G flute (we assume that flute types are known for the current system). Frames with F0s less than 250 Hz and waveform amplitude less than -20 dB are marked as silence. The sign of note change is extracted to represent note change direction. Consecutive note changes in the same direction are then extracted as glissando candidates, which are further pruned by constraints on note numbers (at least 4 for both upward and downward glissandi) and duration (at least 0.2 sec for upward glissandi and 0.15 sec for downward glissandi) based on consultations with the professional performers of the CBF.

HMM-based Identification

Since all glissando candidates extracted in the previous stage share similar pitch evolution characteristics, the input to the HMMs must possess sufficient discriminative power to distinguish glissandi from non-glissandi. Considering the pitch discreteness and long duration of glissandi, we use a feature set consisting of both short-term (average pitch change, average intensity, average intensity change) and long-term (note number, note duration, note range) features (Li et al., 2015; Peeters, 2004). All features are statistics (mean and standard deviation) of pitch and intensity with variations on window and hop sizes. Hop size variations range from 10 to 20 ms at intervals of 2 ms, while window sizes depend on the glissando direction.

Short-term features To capture pitch and intensity change, the short-term window varies from 100 to 200 ms at intervals of 20 ms for

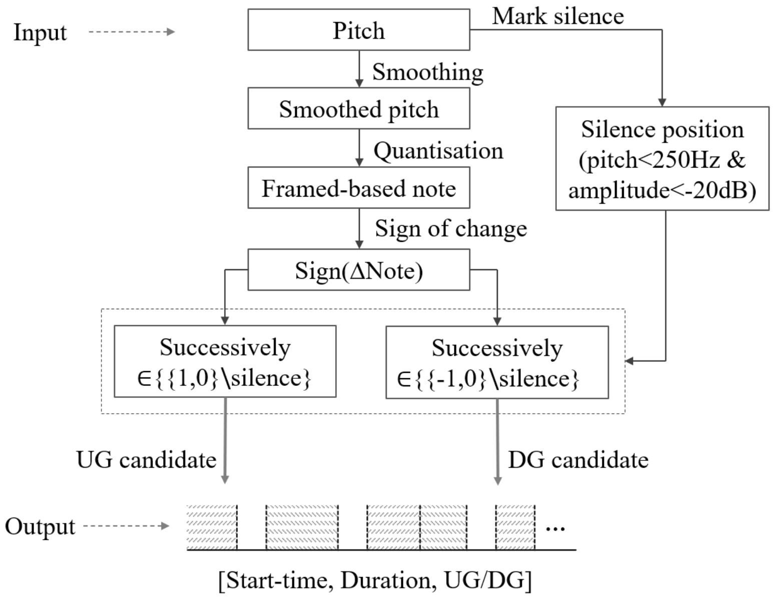

Input
Pitch
Mark silence
Smoothing
Smoothed pitch
Quantisation
Framed-based note
Silence position (pitch<250Hz & amplitude<-20dB)
Sign of change
Sign(ΔNote)
Successively ∈{{1,0}\silence}
Successively ∈{{-1,0}\silence}
UG candidate
DG candidate
Output
...
[Start-time, Duration, UG/DG]

Figure 5.3: Diagram of rule-based segmentation for the proposed glissando detection baseline (UG=upward glissando; DG=downward glissando).

the following three features.

- Average pitch change:

$$\Delta p_i = \frac{1}{W} \sum_{k=1}^{W} \Big[p_i(k) - p_{i-1}(k) \Big], \tag{5.10}$$

  where $p_i(k)$ is the k-th pitch value within the window centered at the i-th time frame, and W is the window length.

- Average intensity (amplitude in dB scale) (Abeßer and Schuller, 2017):

$$I_i = \frac{1}{W} \sum_{k=1}^{W} \Big[20 \cdot log_{10} A_i(k) \Big], \tag{5.11}$$

  where $A_i(k)$ is the amplitude of the k-th sample within the window centered at the i-th time frame, and I_i is average intensity of this window.

- Average intensity change: $\Delta I_i = I_i - I_{i-1}$.

Long-term features To capture the discreteness of pitch evolution in glissandi, note-level features with long windows are calculated. The window sizes vary from 200 to 400 ms at intervals of 50 ms for downward glissandi with shorter duration, and from 200 to 600 ms at the same intervals for upward glissandi which have longer duration. The calculation process for one upward glissando example is shown in Figure 5.4. With a 400 ms window sliding forward, the number of notes N is 8 (one more than the number of peaks, highlighted by the red circles) and the note range (note change between start and end notes) R equals 7. Note durations D, which refer to the intervals between two note change peaks, are {80,40,60,40,40,60} ms.

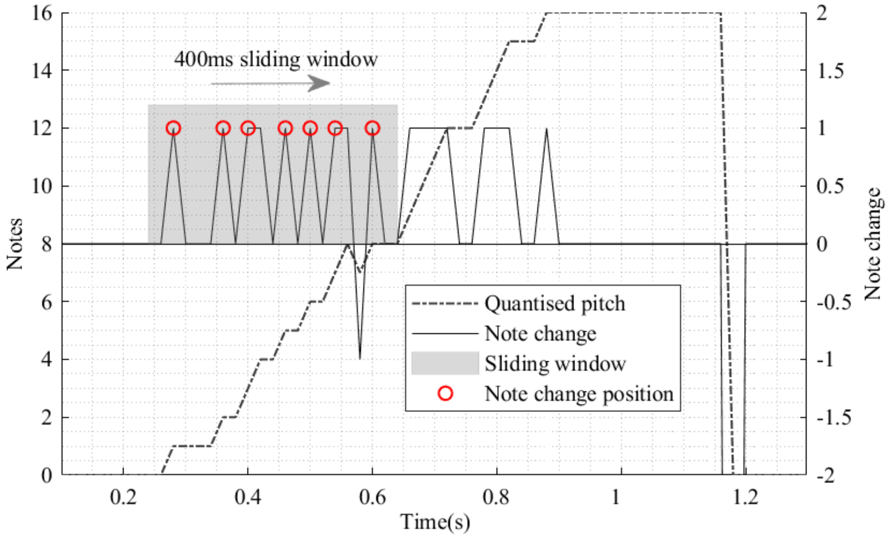


Figure 5.4: Long-term feature calculation for an upward glissando example in the proposed glissando detection system.

As shown in Figure 5.5, two HMMs with Gaussian mixture emissions are trained on the training set, with k-means initialisation and iterative parameterisation by the Expectation-Maximisation algorithm (Murphy, 2012). During the training process, model parameters—the number of HMM latent states, number of Gaussian mixture components, and window-hop sizes—are varied and the model with the best performance on the validation set is chosen as the final one for testing. The emission used is a Gaussian mixture distribution (Murphy, 2012):

$$p(\boldsymbol{x}_i|\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}) = \sum_{m=1}^{M} \pi_m \mathcal{N}(\boldsymbol{x}_i|\mu_m,\Sigma_m), \tag{5.12}$$

where $\boldsymbol{x}_i$ is the observed feature vector of the i-th frame; π_m, μ_m and Σ_m are the prior, mean and covariance of the m-th mixture component, respectively; and $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ are the model parameters, each of which is an M-dimensional vector corresponding to π_m, μ_m, and Σ_m respectively.

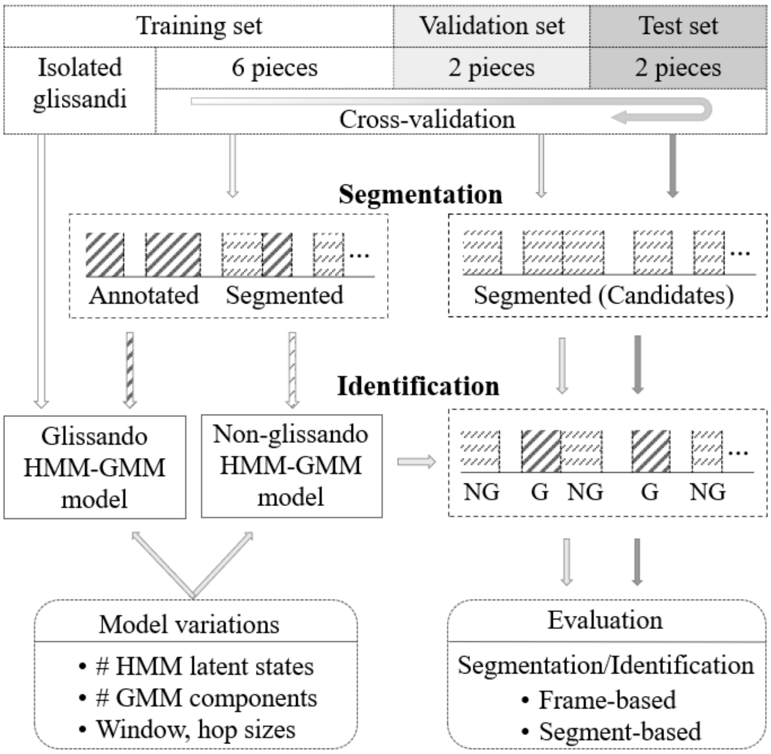


Figure 5.5: Diagram of HMM-based glissando detection baseline (G=glissando; NG=non-glissando).

Evaluation of glissando detection baseline

We evaluate the proposed glissando detection baseline on a subset of the CBFdataset (see Section 3.4), which we named as the *CBF-glissDB* in Wang et al. (2019b). The CBF-glissDB uses only the recordings with glissandi from the CBFdataset, i.e., isolated glissandi and 10 full-length performances of *Busy Delivering Harvest* «扬鞭催马运粮忙» and *Jolly Meeting* «喜相逢» by 10 performers. The total duration of the CBF-glissDB is 51 minutes. The data is subdivided into three subsets, namely, training (all isolated glissandi and 6 full pieces), validation (2 full pieces), and test (2 full pieces), as shown in Figure 5.5. The segmentation stage is applied to full-piece recordings in all three subsets, but to different ends. For the training set, segmentation is carried out conservatively, promoting recall over precision, in order to extract all relevant segments, but leading also to the extraction of false positive segments. False positive segments are then used to train an NG-HMM. In the validation and test stages, the extracted segments serve as candidates to be assigned glissando (G) or non-glissando (NG) labels by comparing the log-likelihoods calculated by the two HMMs. Since the HMMs are applied directly to the candidate segments, the absolute position of glissandi in the pieces does not influence the result. The ten full-piece recordings are randomly allocated to the training, validation, and test sets in a 6:2:2 ratio at the beginning of experiment. A five-fold cross-validation is then conducted.

To investigate the influence of automatic pitch detection on glissando detection, we build two systems: an *F0-informed system* using the pitch ground truth as input; and a *fully-automated system* using pitch automatically estimated by pYIN pitch detection algorithm (Mauch and Dixon, 2014) as input. The former is to assess the performance of the proposed glissando detection system independently of the pitch estimation performance. The pitch ground truth is created by the author of this thesis using Sonic Visualiser (Cannam et al., 2010) via a manual correction of the errors of the F0s estimated by pYIN (Mauch and Dixon, 2014).

Because glissando length ranges approximately from 200 to 1100 ms, for each system, frame- and segment-based evaluations are implemented. The frame size used in frame-based evaluation is 20 ms. Segment-based evaluation compares detected and ground truth glissandi in short-time, non-overlapping segments (Mesaros et al., 2016). A segment length of 100 ms is adopted. True positives are segments which have overlaps with both ground truth and detected glissandi; false positive segments overlap only with detected glissandi; and, false negatives intersect with ground truth only.

We evaluate the segmentation and detection results using the frame-based precision $\mathcal{P}$, recall $\mathcal{R}$, and F-measure $\mathcal{F}$ introduced in Subsection 2.2.3 as metrics. Table 5.4 lists segmentation and detection results for both upward and downward glissandi in the F0-informed detection system. As can be seen, the segmentation stage performs a conservative selection of candidate segments with high recall and low precision. The large number of false positives obtained for NG-HMM training benefits the data balance in our system. The better identification performance of upward glissandi over downward ones can be attributed to their more regular patterns. As can be seen, the identification F-measure increases by approximately 60% as compared to the segmentation F-measure, which verifies our intuition that consecutive pitch changes can be captured by HMMs.

Stage	Glissando direction	Frame-based (%)			Segment-based (%)		
		$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
Rule-based segmentation	Upward	3.1	93.4	5.9	3.1	92.8	6.0
	Downward	4.9	83.1	9.0	5.1	86.9	9.9
HMM-based Identification	Upward	73.4	75.4	73.4	72.0	74.0	72.0
	Downward	65.4	67.6	63.2	64.4	70.2	64.2

Table 5.4: Evaluation results of the F0-informed glissando detection system on the CBF-glissDB based on annotated pitch.

After verifying the glissando detection stage independently, we then use the automatically estimated pitch to evaluate the fully-automated glissando detection system. Due to the influence of breathing, some parts in the CBF recordings have high intensity but no detected pitch. Thus silence cannot be determined only by pitch presence, and we define silence segments as parts having both no pitch and intensity below -20 dB. Correctly detected frames are the voiced parts with pitch intervals less than half a semitone between the ground truth and the detected pitch. Pitch estimation accuracy refers to the percentage of correctly detected frames over all voiced frames. Table 5.5 shows the estimated pitch detection results for both full-piece recordings and ground truth glissando segments within these pieces. The poorer pitch estimation performance on glissando segments shows that pYIN works less well on rapid pitch evolution progressions.

Type	Full-length pieces		Glissando segments	
	Southern	Northern	Upward	Downward
Accuracy (%)	80.2	79.5	72.0	74.8

Table 5.5: Pitch estimation accuracy for full-length pieces and glissando segments in the CBF-glissDB.

The fully-automated glissando detection results are shown in Table 5.6. Considering the pitch evaluations shown above, it is reasonable to expect worse performance when using automatically estimated pitch as input. Pitch is a main discriminative feature in the proposed glissando detection method. The presence of undetected pitches or octave errors within glissandi hinders the G-HMM to capture the consecutive note evolution. Thus false positives, which exhibit similar pitch evaluation as the ground truth glissandi and have higher pitch estimation accuracy, may be assigned with G labels. This is verified by the better identification performance on downward glissandi over upward ones with lower pitch estimation results.

Stage	Glissando direction	Frame-based (%)			Segment-based (%)		
		$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
Rule-based segmentation	Upward	2.1	84.8	4.1	2.1	86.2	4.4
	Downward	3.3	67.3	5.9	3.6	75.0	7.1
HMM-based identification	Upward	36.4	63.2	44.6	36.8	63.4	45.0
	Downward	58.2	48.4	50.4	58.0	51.8	52.6

Table 5.6: Evaluation results of the fully-automated glissando detection system on the CBF-glissDB based on estimated pitch.

Note that the proposed baseline method is tested on a subset of CBFdataset (see Section 3.4), i.e., CBF-glissDB, while the proposed dJTFS approach (see Section 5.3) is evaluated on the complete CBF-dataset. For fair comparisons, we reimplement this baseline method on the CBFdataset and resample the detection result into the same frame size as that we use for CBF glissando evaluation using the dJTFS, i.e., 93 ms (see Table 5.3). Much lower F-measure scores are obtained: 14.0% for upward glissando and 11.6% for downward glissando detection. This may be attributed to the hand-crafted rules introduced in this method which may not be robust when the dataset is expanded with noisy information only. The expanded part of CBFdataset as compared to the CBF-glissDB comprises recordings without any glissando examples. Our proposed recognition system based on the dJTFS aims at developing a general framework for playing technique recognition rather than on glissando only. Therefore we use the average F-measure score of upward and downward glissandi as the final glissando detection results, i.e., 12.8%, when comparing with the dJTFS, as shown in Table 5.7. Although this result is low as compared to that of the proposed dJTFS method, it is still higher than a random baseline which would have been the duration ratio of all glissandi over the whole CBFdataset, i.e., 5.0 min/2.6 hours ≈ 3%.

5.4.2 Baselines for multiclass classification scheme

To detect all seven types of playing techniques simultaneously via the multiclass classification scheme (see Subsection 5.5.2), we compare the proposed representations with commonly used features such as mel-frequency cepstral coefficients (MFCCs) (Abeßer et al., 2010) and modulation power spectrum (MPS) (Thoret et al., 2017) on the CBFdataset (see Section 3.4). The MPS is also a representation capable of capturing spectro-temporal modulation information in audio signals and has been applied to instrument recognition (Thoret et al., 2017) while not yet been used for playing technique recognition. Frame-based macro F-measures obtained by the MFCCs and the MPS for recognising the seven CBF playing techniques are 35.9% and 52.0%, respectively.

5.5 Evaluation

In this section, we evaluate the binary and multiclass classification schemes on the CBFdataset (see Section 3.4) using the same evaluation metrics introduced in Subsection 2.2.3, i.e., precision $\mathcal{P}$, recall $\mathcal{R}$, and F-measure $\mathcal{F}$ scores. Similarly to Subsection 4.4.2, we consider both frame- and event-based evaluation for the binary classification scheme. The former evaluation method compares the predictions with the ground truth in a frame-wise manner while the latter one merges frame labels into events and postprocess the events by minimum duration pruning and gap filling. We fill the gaps between neighbouring events when the gaps are shorter than the shortest event in the training set; and prune the events that have smaller duration than the minimum duration event in the training set. The minimum duration is automatically calculated subject to the technique and the training-test data split during recognition. An event is considered to be detected only when its onset falls within a 200 ms window of the ground truth and its duration is at least 50% of the ground truth duration. For the multiclass classification scheme, only frame-based evaluation is used. This is because the single-label multiclass classification scheme is not capable of detecting co-articulations, where merging frame labels into events is not accurate.

5.5.1 Binary classification scheme

Table 5.7 and Table 5.8 display the frame- and event-based binary classification results, respectively, for acciaccatura, portamento, and glissando using the proposed dJTFS based on max-pooling (dJTFS-max), dJTFS based on average-pooling (dJTFS-avg), the JTFS, the JTFS+PCA, and the HMMs baseline. We compare these results from three perspectives: the overall performance of all methods, the results on different playing techniques, and the performance of two dJTFS representations. Among all the methods, the dJTFS-avg achieves the best overall performance measured by the average F-measure over the three playing techniques, i.e., 75.7% and 75.6% in the frame- and event-based evaluation, respectively. The JTFS performs the second best, with the average F-measure 1.2% and 0.9% lower than those obtained by the dJTFS-avg for the two evaluation methods, respectively. All the four joint scattering representations significantly outperform the HMMs baseline.

For the recognition results on specific playing techniques, the highest F-measure scores are obtained by different methods: 66.4% for portamento detection using the dJTFS-max, 74.8% for acciaccatura recognition using the dJTFS-avg, and 86.8% on glissando detection using the JTFS, respectively. The above scores are frame-based F-measures; a similar trends appear in the event-based evaluation, as shown in Table 5.8. For both evaluation methods, all the scattering representations—dJTFS-max, dJTFS-avg, JTFS, and JTFS+PCA—considerably outperform the HMMs baseline.

Comparing the two types of dJTFS representations, i.e., dJTFS-max and dJTFS-avg, we notice that the latter outperforms the former for recognising both acciaccatura and glissando while for portamento detection, the latter underperforms the former. This may be attributed to the instability of the dJTFS-max when the direction with maximum energy, $\theta_{\max}$, oscillates between the upward and downward directions. This may be the case for acciaccatura and glissando due to their noisy note changes within the technique. Take the glissando technique in Figure 5.2 (a) for example, although the direction of the glissando is upward, downward note changes exist inside the playing technique, e.g. the note change at around 2.5 sec. For such cases, the dJTFS-max feature may oscillate between $\theta = 1$ and $\theta = -1$ within the playing technique and is less stable than the dJTFS-avg. In contrast, the portamento technique comprises smooth pitch changes where a direction change within the playing technique is less likely to happen.

Method	Acciaccatura	Portamento	Glissando	Average
dJTFS-max	70.4	**66.4**	81.4	72.7
dJTFS-avg	**74.8**	66.0	86.4	**75.7**
JTFS	73.0	63.6	**86.8**	74.5
JTFS+PCA	70.4	54.6	85.6	70.2
HMMs	25.0	30.0	12.8	22.6

Table 5.7: Frame-based performance comparison of binary classification for acciaccatura, portamento, and glissando in the CBFdataset (see Section 3.4) using the proposed dJTFS-max and dJTFS-avg, the JTFS, the JTFS+PCA, and the HMM-based glissando detection baseline. All numbers are F-measures (%). "Average" refers to the average F-measure over the three playing techniques.

Method	Acciaccatura	Portamento	Glissando	Average
dJTFS-max	74.1	**65.7**	75.6	71.8
dJTFS-avg	**78.2**	65.6	83.1	**75.6**
JTFS	76.1	63.5	**84.6**	74.7
JTFS+PCA	73.6	52.0	84.3	70.0
HMMs	23.5	22.4	14.8	20.2

Table 5.8: Event-based performance comparison of binary classification for acciaccatura, portamento, and glissando in the CBFdataset (see Section 3.4) using the proposed dJTFS-max and dJTFS-avg, the JTFS, the JTFS+PCA, and the HMM-based glissando detection baseline. All numbers are F-measures (%).

Cross checking the detection results with the original audio, we find another two types of errors besides the repaid note changes described in Subsection 4.4.5: co-articulation and playing techniques exhibit similar spectro-temporal patterns as those of non-techniques, for example, short portamento and note change. Figure 5.6 top shows the log-frequency spectrogram of an excerpt in performance of *Busy Delivering Harvest* by Player 3 in the CBFdataset (see Section 3.4); the bottom subfigure displays the ground truth portamento and frame-based classification output. The false negative at around 9 sec is an example of portamento and flutter-tongue co-articulation. In such cases, portamento is no longer smooth but modulated with small ripples, making it hard to detect, even with 16 filters per octave in the first-order scattering transform. The false negative at 12.5 sec is an instance of the system misclassifying a note change into a portamento.

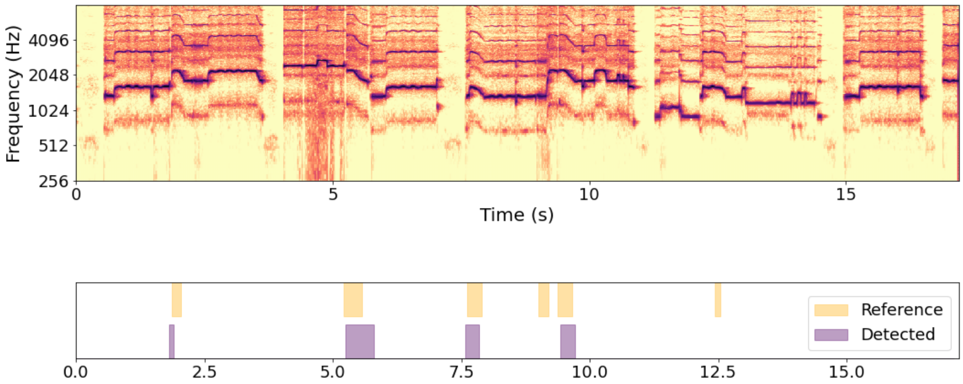


Figure 5.6: Portamento detection result for an excerpt in Player 3's performance of *Busy Delivering Harvest* using the dJTFS-avg. Top: log-frequency spectrogram; bottom: comparison between the ground truth (upper half) and frame-based classification output (lower half). For this example, frame-based $\mathcal{P}$=85%, $\mathcal{R}$=52%, $\mathcal{F}$=65%.

Besides the recognition performance, we also compare the scattering representations—the proposed dJTFS, the JTFS, and the JTFS+PCA—in terms of computational expense. Since both the dJTFS and the JTFS+PCA are derived from the JTFS, where the direction selection operation, either max-pooling or average-pooling, and the PCA operation are negligible as compared to the calculation of the JTFS, we compare only the computation cost at the classification stage. Table 5.9 lists the dimensionalities and the runtimes of these three representations for recognising the three types of PETs. It can be seen that for all three playing techniques, the classification using the JTFS is much more expensive than the other two representations due to its high dimensionality; and the runtime is positively correlated with the dimensionality of the representation. This matches the finding in Subsection 4.4.5 that for the proposed recognition system, when the performance is guaranteed, a lower dimensional representation is preferable for computation saving.

Representation	Dimensionality/runtime		
	Acciaccatura	Portamento	Glissando
JTFS+PCA	67/0.44	23/0.20	63/0.29
dJTFS	366/1.58	410/1.59	498/1.07
JTFS	1490/7.04	1486/4.36	1770/4.50

Table 5.9: Dimensionality and classification runtime (in hours) of the proposed dJTFS, the JTFS, and the JTFS+PCA for recognising acciaccatura, portamento, and glissando in the CBFdataset (see Section 3.4).

5.5.2 Multiclass classification scheme

Although the binary classifiers above detect co-articulations, cases of co-articulation form only a small portion of the CBFdataset (see Section 3.4). In the multiclass classification scheme, we discard all co-articulation samples. This enables us to generate a confusion matrix between techniques, but also to provide comparable results for different datasets across different benchmark methods. The system recognises all seven playing techniques simultaneously using the concatenation of the proposed AdaTS+AdaTRS (see Section 4.2) and the dJTFS-avg (see Section 5.2) features as input. The dJTFS-avg is used because of its better overall performance for PET recognition as compared to the dJTFS-max (see Subsection 5.5.1). We include an extra class 'other' to account for frames that are none of the discussed seven playing techniques.

Figure 5.7 (a) shows the frame-based F-measures of multiclass classification for the seven types of playing techniques, i.e., flutter-tongue, trill, vibrato, tremolo, acciaccatura, portamento, and glissando, using the proposed scattering representations, the MFCCs baseline, and the MPS baseline. The macro F-measures over the techniques obtained using these three representations are 79.5%, 35.9%, and 52.0%. As can be seen, the proposed scattering representations outperform both baselines for all playing techniques; the MPS also exhibits better performance than the MFCCs. Figure 5.8 (a) shows the confusion matrix with the number of frames detected for each class using the proposed scattering representations, where the 'other' cases form the majority of the CBFdataset (see Section 3.4). This is because playing techniques are rare events in real-world performances. Additionally, among the seven types of playing techniques, the number of samples for each class is highly

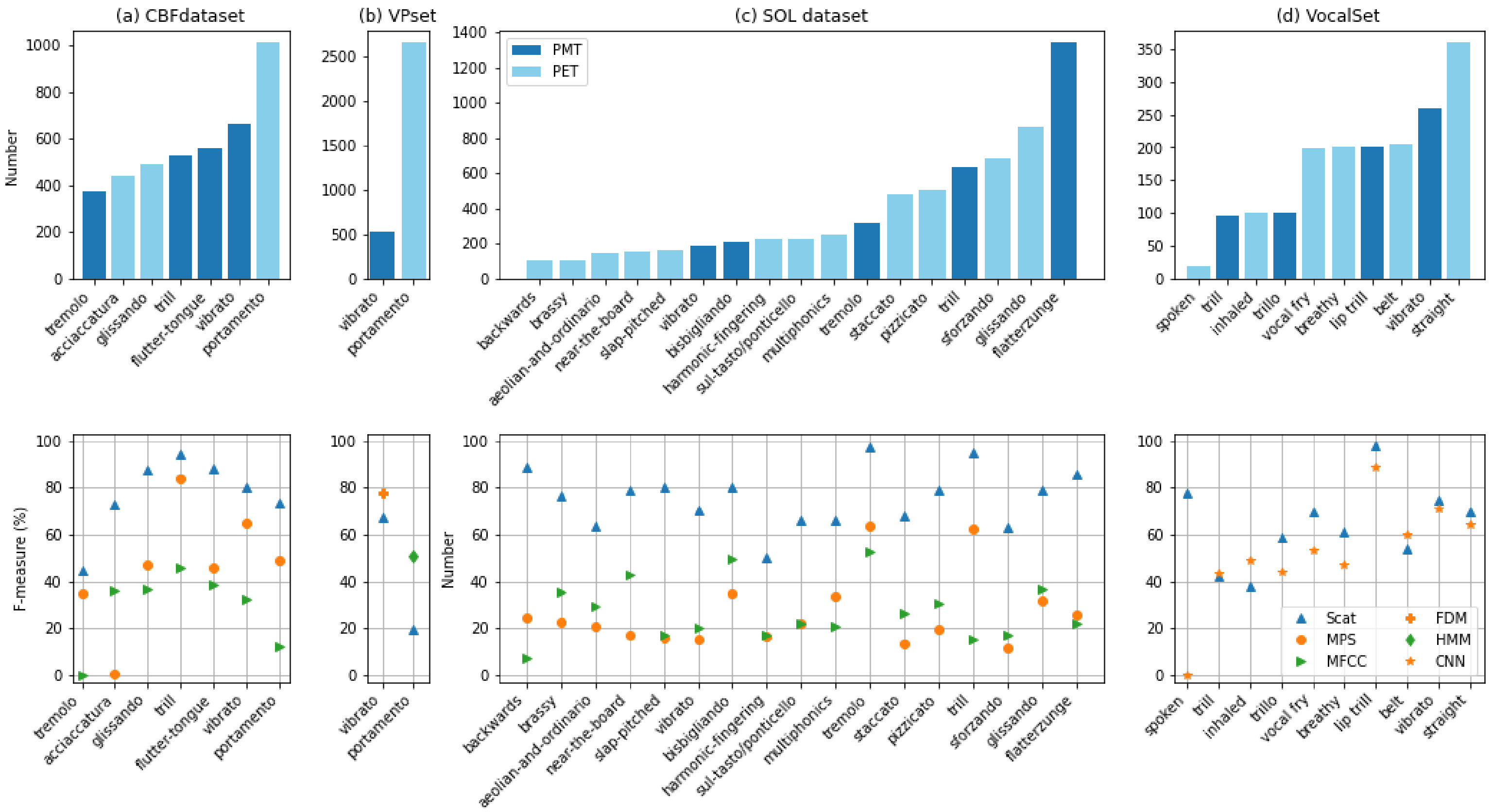


Figure 5.7: Number of playing techniques and recognition results on the CBFdataset, VPset, SOL dataset, and VocalSet. Top: the list of playing techniques and number of samples in each dataset. The techniques are grouped into pitch modulation-based techniques (PMTs, in blue) and pitch evolution-based techniques (PETs, in light blue). Bottom: recognition results by multiclass classification for each dataset (binary classification for the VPset). The blue triangles are the F-measure scores obtained by the proposed scattering representation while others are those of the baseline methods.

imbalanced, according the statistical information of the techniques we obtained in Section 3.4. We then normalise the detection result over the number of instances per technique class and obtain the normalised confusion matrix in Figure 5.8 (b). The confusion between vibrato and tremolo is expected since frequency variations are commonly accompanied with amplitude modulations and vice versa. In the case of CBF, such co-articulations are common because of the instrument gestures of vibrato and tremolo. Vibratos can be generated by fingering or tonguing, while tremolos are commonly produced by breath variations. Performers also frequently add tremolo effects on top of other playing techniques for expressivity.

Figure 5.9 and Figure 5.10 show the confusion matrices obtained for multiclass classification of the seven playing techniques using the MFCCs and the MPS features on the CBFdataset (see Section 3.4), respectively. High confusion between the playing techniques and the 'other' class is observed for both baseline features. Both features achieve the best score for trill recognition. The MFCCs detect none of the tremolos while the MPS returns the lowest score for acciaccatura recognition. The confusion between vibrato and tremolo is also found from the recognition results obtained by the MPS.

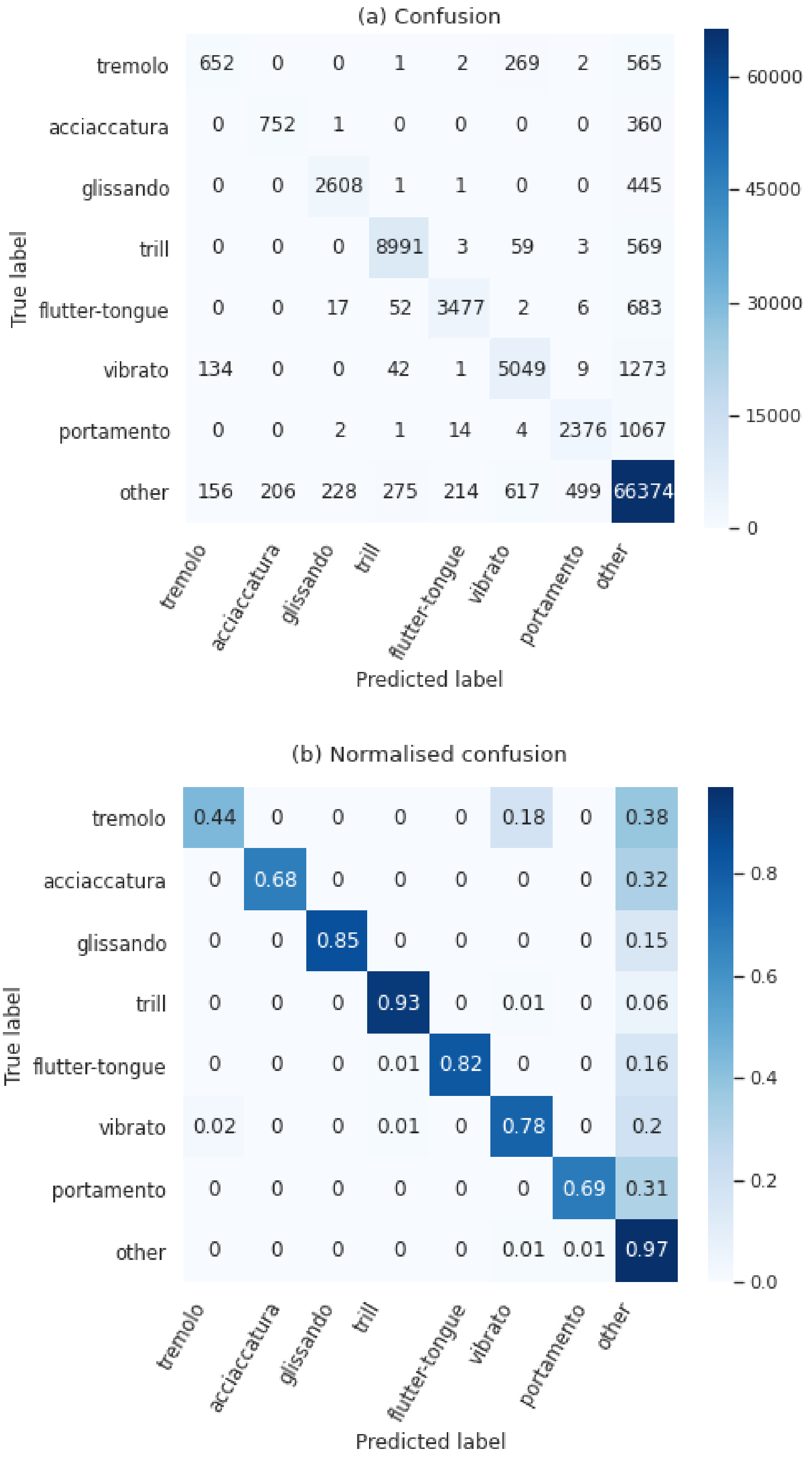
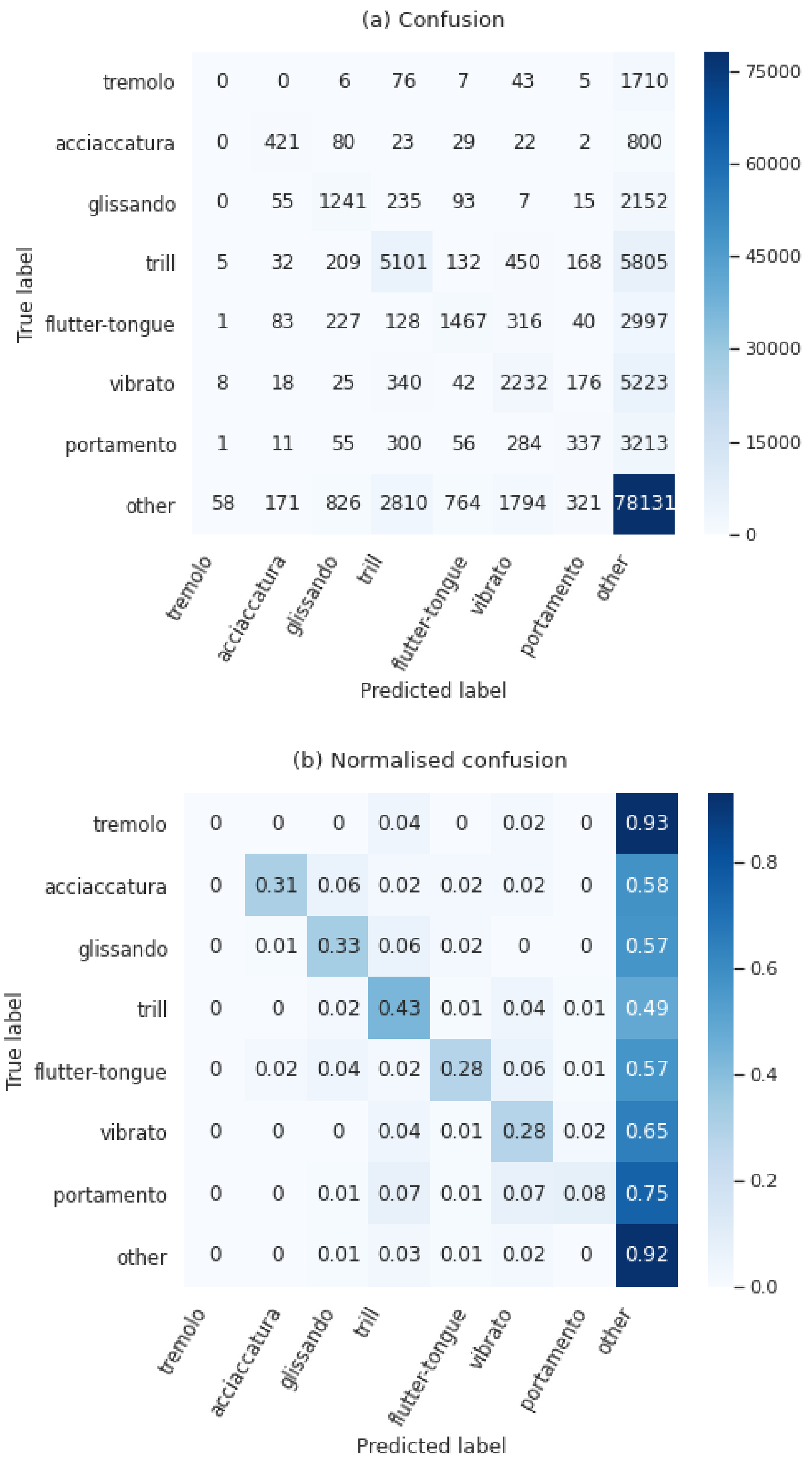


Figure 5.8: Confusion matrices obtained for multiclass classification of the seven CBF playing techniques using the proposed scattering representations. (a) confusion matrix with number of frames detected; (b) normalised confusion matrix with (a) divided by the number of samples per technique.



(a) Confusion

(b) Normalised confusion

Figure 5.9: Confusion matrices obtained for multiclass classification of the seven CBF playing techniques using the MFCCs. (a) confusion matrix with number of frames detected; (b) normalised confusion matrix with (a) divided by the number of samples per technique.

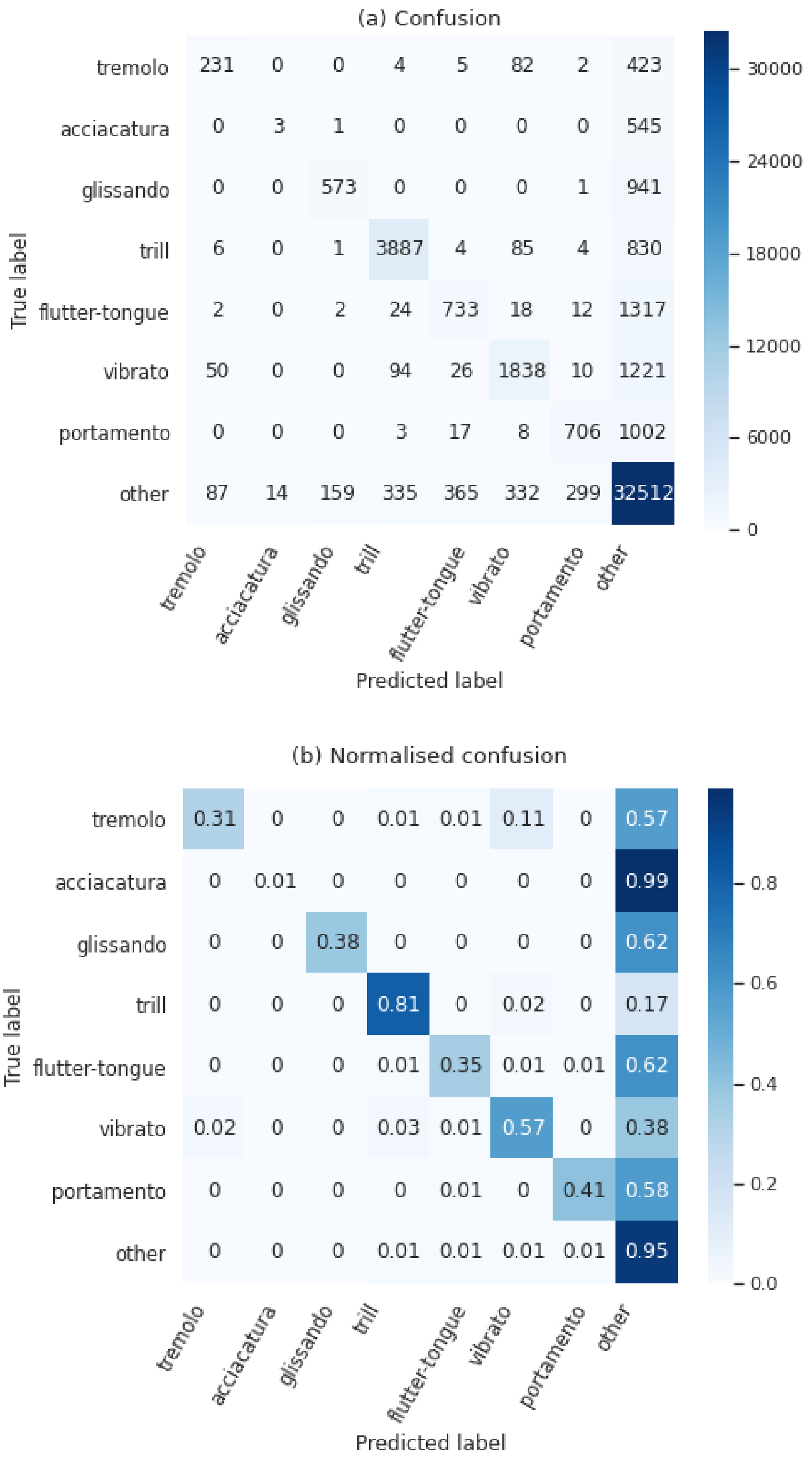


Figure 5.10: Confusion matrices obtained for multiclass classification of the seven CBF playing techniques using the MPS. (a) confusion matrix with number of frames detected; (b) normalised confusion matrix with (a) divided by the number of samples per technique.

5.6 Additional Evaluations

To verify the generalisability of the proposed framework for playing technique recognition, we test it on three existing datasets with a variety of playing techniques: vibrato/portamento dataset (VPset) (Yang, 2017), Studio On Line (SOL) dataset (Lostanlen et al., 2018), and vocal technique dataset (VocalSet) (Wilkins et al., 2018). We call these three datasets the *additional datasets* and provide details of the content of each dataset in Subsection 2.2.2. The types of playing techniques and number of samples in each dataset are summarised in the top subfigures of Figure 5.7 (b), (c), and (d), respectively. Besides the frame- and event-based evaluation used in Section 5.5, we consider also the clip-based evaluation introduced in Subsection 2.2.3 for the playing technique recognition on the SOL dataset, which comprises only short audio clips with one technique per clip.

For the additional datasets, we conduct binary classification for the VPset and multiclass classification for the SOL dataset and the VocalSet. This is because vibrato and portamento techniques are from two separate subsets in the VPset (see Subsection 2.2.2). All experiments for the additional datasets use the same settings as the CBF binary or multiclass classification, i.e., hyperparameters of the proposed scattering representations and hyperparameter grids of the SVMs classifiers. However, the ways of splitting data vary according to dataset. For the VPset, rather than cross-validating within the same performance as Yang (2017), we take into account the performer identity. We use one performer's playing for testing and the remaining three recordings for training, and repeat this for all four performers. The final result is the average of frame-based F-measure scores over all performers. After removing silence from the recordings, a random split ratio of 8:2 between training and test sets is used for the SOL dataset due to a lack of performer identity information. Five splits are then conducted in a circular way. For the VocalSet, we keep the training-test split as that in the original work (Wilkins et al., 2018). Silence is also removed before the scattering feature extraction. All samples from 15 singers are placed in the training set and the remaining 5 singers in the test set.

Similarly to the training process (see Subsection 5.3.2) on the CBF-dataset, for the additional datasets, we also optimise the SVM hyperparameters via a 3-fold cross-validation on the training set. This is done by sampling on the training set in a way that ensures each fold includes approximately the same ratio of positive and negative class instances for a given playing technique. For the VPset, the frame sizes and feature dimensionalities of the scattering representations, i.e., the AdaTS+AdaTRS for recognising vibratos and the dJTFS-avg for detecting portamenti, are the same as those we use for CBF vibrato (186 ms and 203) and portamento (93 ms and 410) recognition in the binary classification scheme, as shown in Table 4.3 and Table 5.3, respectively. For the SOL dataset and the VocalSet, the frame sizes and feature dimensions are the same as those in the multiclass classification of CBF playing techniques, i.e., 93 ms and 613 (see Subsection 5.3.1).

The baseline methods for the additional datasets are also different according to dataset. We compare the proposed system with the filter diagonalisation method (FDM, see Subsection 4.4.3) for vibrato detection and the hidden Markov models (HMMs, see Subsection 5.4.1) for portamento recognition in the VPset; and with convolutional neural networks (CNNs, see Section 2.5) for detecting vocal techniques in the VocalSet. This is because these methods were originally used for detecting playing techniques in the corresponding datasets. Frame-based F-measures for vibrato recognition using the FDM and for portamento detection using the HMMs are 77.7% and 50.6%, respectively, on the VPset. CNNs were used in Wilkins et al. (2018) for vocal technique classification with a frame size of 3 seconds. Macro averaged F-measure of 65.2% for the 10 techniques were reported. For the SOL dataset, we compare the proposed scattering representations with the MFCCs and the MPS (see Subsection 5.4.2). Macro averaged F-measures for detecting the 17 playing techniques in the SOL dataset using MFCCs and MPS are 27.1% and 26.6%, respectively. All the F-measure scores above are obtained by resampling the recognition results from the baseline methods into the frame size of the multiclass classification of CBF playing techniques for fair comparisons, i.e., 93 ms, except that of the vocal technique recognition, where the frame size (3 seconds) is much larger than this value.

The bottom subfigures of Figure 5.7 (b), (c), and (d) display frame-based F-measures for recognising each type of playing technique in the VPset, SOL dataset, and VocalSet, respectively. Note that these results are based on the same scattering hyperparameters that we use for the CBFdataset. Parameter tuning for each dataset could potentially improve the recognition results. In the VPset, the proposed method achieves comparable results to the FDM on vibrato detection while considerably underperforms the HMMs for portamento detection. The most frequent errors found are note changes being detected as portamenti, which is consistent with the detection errors of CBF portamenti. Our recognition system achieves comparable overall performance as the CNNs, with a macro F-measure score of 64.5% against 65.2%. However, the F-measures from the proposed system over playing techniques is more stable than those from the CNNs, where the latter failed to recognise any of the 20 spoken techniques.

A macro F-measure score of 84.4% is obtained for recognising playing techniques in the SOL dataset. The much better overall performance of the proposed system than the baselines may be attributed to the SOL dataset structured with short clips of individual techniques. No technique is based on the same note of the same instrument, which offers high intra-class variability for playing techniques. Figure 5.11 and Figure 5.12 show the confusion matrices obtained for multiclass classification of the playing techniques on the SOL dataset and the VocalSet, respectively. The recognition results on the SOL dataset present few confusions between techniques. For playing techniques in the VocalSet, the lip trill obtains the best score while the inhale technique obtains the lowest. The straight technique exhibits high confusion with other techniques.

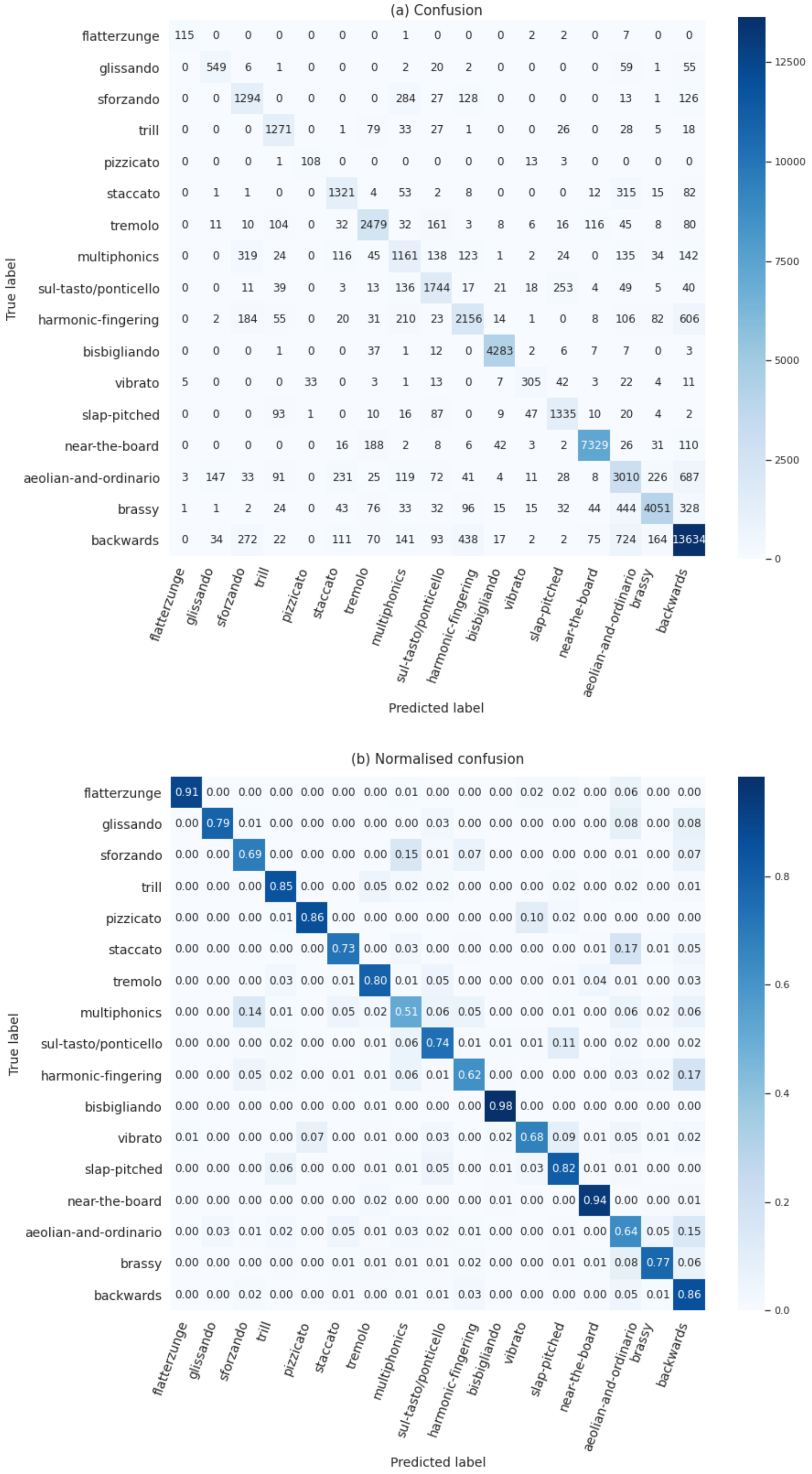
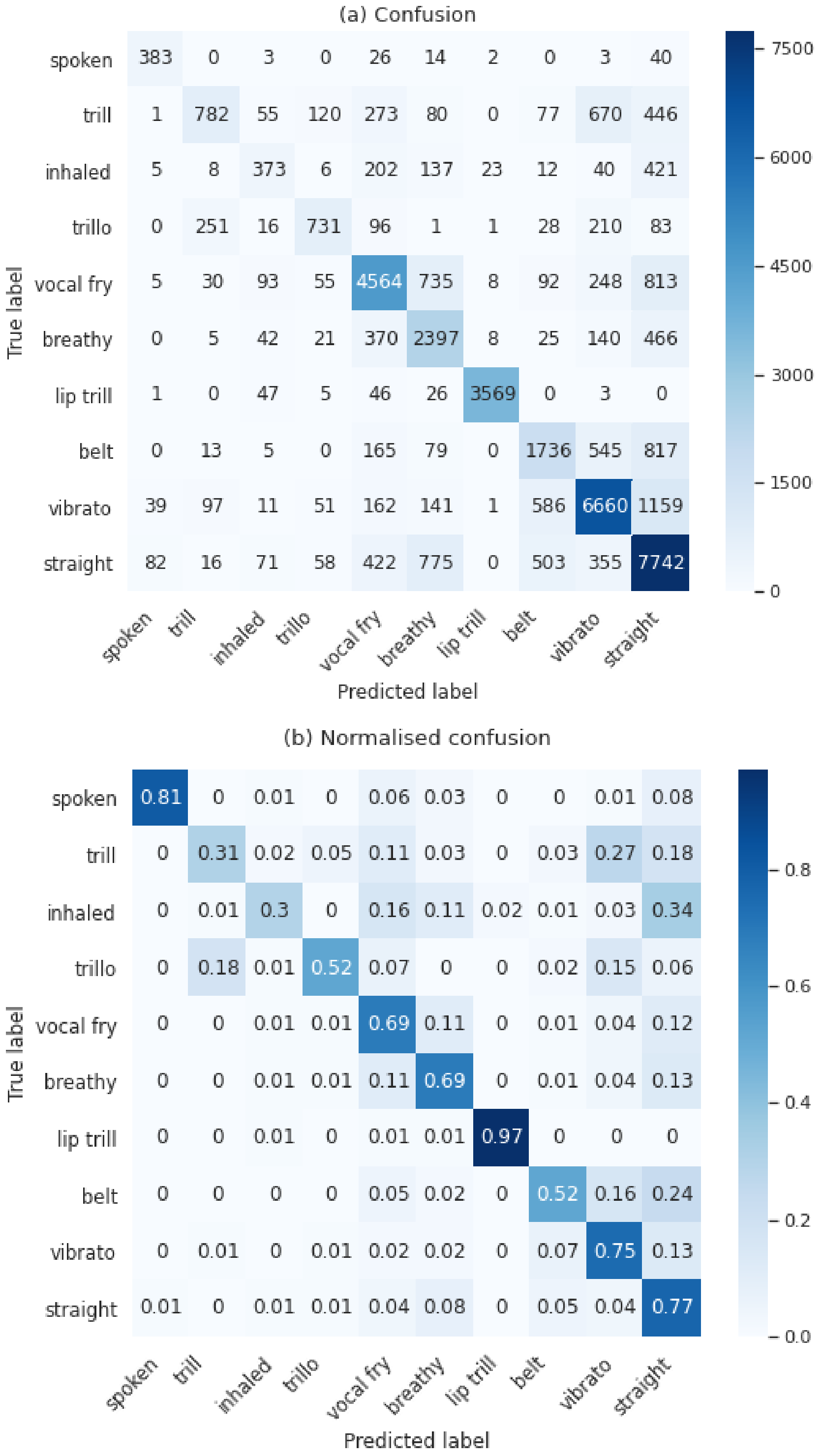


Figure 5.11: Confusion matrices obtained for multi-class classification of playing techniques in the SOL dataset. (a) confusion matrix with number of frames detected; (b) normalised confusion matrix with (a) divided by the number of samples per technique.



(a) Confusion

	spoken	trill	inhaled	trillo	vocal fry	breathy	lip trill	belt	vibrato	straight
spoken	383	0	3	0	26	14	2	0	3	40
trill	1	782	55	120	273	80	0	77	670	446
inhaled	5	8	373	6	202	137	23	12	40	421
trillo	0	251	16	731	96	1	1	28	210	83
vocal fry	5	30	93	55	4564	735	8	92	248	813
breathy	0	5	42	21	370	2397	8	25	140	466
lip trill	1	0	47	5	46	26	3569	0	3	0
belt	0	13	5	0	165	79	0	1736	545	817
vibrato	39	97	11	51	162	141	1	586	6660	1159
straight	82	16	71	58	422	775	0	503	355	7742

(b) Normalised confusion

	spoken	trill	inhaled	trillo	vocal fry	breathy	lip trill	belt	vibrato	straight
spoken	0.81	0	0.01	0	0.06	0.03	0	0	0.01	0.08
trill	0	0.31	0.02	0.05	0.11	0.03	0	0.03	0.27	0.18
inhaled	0	0.01	0.3	0	0.16	0.11	0.02	0.01	0.03	0.34
trillo	0	0.18	0.01	0.52	0.07	0	0	0.02	0.15	0.06
vocal fry	0	0	0.01	0.01	0.69	0.11	0	0.01	0.04	0.12
breathy	0	0	0.01	0.01	0.11	0.69	0	0.01	0.04	0.13
lip trill	0	0	0.01	0	0.01	0.01	0.97	0	0	0
belt	0	0	0	0	0.05	0.02	0	0.52	0.16	0.24
vibrato	0	0.01	0	0.01	0.02	0.02	0	0.07	0.75	0.13
straight	0.01	0	0.01	0.01	0.04	0.08	0	0.05	0.04	0.77

Figure 5.12: Confusion matrices obtained for multi-class classification of the ten vocal techniques in the VocalSet. (a) confusion matrix with number of frames detected; (b) normalised confusion matrix with (a) divided by the number of samples per technique.

5.7 Conclusions

In this chapter, we have presented the direction-invariant joint time–frequency scattering (dJTFS), a variant of the joint time–frequency scattering (JTFS), for representing pitch evolution-based techniques (PETs). To detect PETs regardless of their directions, we have investigated two ways of creating the direction invariance, i.e., max-pooling and average-pooling, which result in the dJTFS-max and dJTFS-avg representations. With the dJTFS features extracted, we build a recognition system with three binary classifiers, each for detecting on type of PETs, i.e., acciaccatura, protamento, and glissando. The results show that all the scattering representations, i.e., the proposed dJTFS-avg and dJTFS-max, the JTFS, and the JTFS with dimensionality reduced by principal component analysis (PCA), considerably outperform the hidden Markov model (HMMs) baseline. Among these scattering representations, the dJTFS-avg achieves the best overall results in terms of the average F-measure over all playing techniques. One possible reason for the better performance of dJTFS-avg over the dJTFS-max may be the instability of the dJTFS-max when the directions within the playing techniques oscillates between upward and downward directions.

To detect both pitch modulation-based techniques (PMTs) and PETs simultaneously, we have developed another recognition system which takes the concatenation of the AdaTS+AdaTRS (see Section 4.2) and the dJTFS-avg as input. As compared to the mel-frequency cepstral coefficients (MFCCs) and the modulation power spectrum (MPS) baseline features, the proposed scattering representations outperforms both for all the seven types of playing techniques we have investigated in Chapter 4 and in this chapter: vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando. To test the generalisability of the proposed methodology, we apply this recognition system to three additional datasets with a variety of instrumental and vocal playing techniques and obtain promising results.

We list some limitations of the study conducted in this chapter and present possible directions for improvement. We set the hyperparameters of the proposed scattering representations, i.e., the adaptive scattering and the dJTFS, based on the characteristics of the playing techniques (see Table 4.1 and Table 5.1). Among these characteristics, only the averaging scale T is grounded on the duration range of the playing techniques in the CBFdataset (see Section 3.4). Yet, to the author's knowledge, these averaging scales are applicable to the same playing techniques of other instruments or singing voice. For example, we use averaging scale $T = 2^{15}$ (in samples, equivalent to 743 ms at a sampling rate 44.1 kHz) for CBF vibrato recognition. This corresponds to a modulation rate of 1.35 Hz. All vibratos with modulation rate above this value can be potentially detected, such as singing vibratos ranging from 5 to 8 Hz (Prame, 1994), erhu vibratos between 4 and 8 Hz (Yang, 2017), and violin vibratos from 2 to 15 Hz (Zhao et al., 2021). The other hyperparameters of the adaptive scattering are motivated by the characteristics of the playing techniques in general, for instance, $Q_1^{(\mathrm{t})} = 12$ to account for note-level pitch changes and and $Q_1^{(\mathrm{t})} = 16$ for capturing subtle modulations or smoothed pitch changes.

The motivation of setting the scattering hyperparamters based on playing technique characteristics in general is to develop an explainable and generalisable playing technique recognition framework that can be applied to other instruments and datasets directly without retraining. The performance of the proposed presentations on the three additional datasets in Section 5.6 verifies the generalisability of the hyperparameters that we use on the CBFdataset (see Section 3.4). However, if our target is to obtain an optimal performance on a specific dataset, automatically tuning the hyperparameters of the scattering representations and the classifier jointly would be the best choice.

Since co-articulations form a small portion of the CBFdataset (see Section 3.4), we conduct single-label multiclass classification during the recognition. This is done by discarding the samples with more than one label. In practice, a user may expect a recognition system to detect all playing technique components in a co-articulation. In this case, a multilabel classifier should be the most appropriate choice. In this chapter, we use the mean and standard deviation of a fixed number of frames centred at the current frame for calculating the dJTFS features. One may apply recurrent classifiers such as long short-term memory units (Warrick et al., 2019) to account for temporal context.

The experiments in this chapter all operate on the original datasets directly without any data augmentation. However, no evidence is found that the detection result relies highly on the number of samples (see

Figure 5.7). The techniques with many more samples do not achieve better results, e.g. portamento in the CBFdataset (see Section 3.4) and straight in the VocalSet. This points to a robustness of the proposed representations to learn quickly the discriminative information based on a small number of examples. For playing techniques across different datasets, we currently follow the taxonomy of the original datasets. Yet, playing techniques are musical patterns, which vary over instruments, regions, styles, and performers and may require a consistent yet flexible taxonomy. The same technique may exist under a different name in the context of another instrument or genre. For example, portamento in the VPset corresponds to glissando in the SOL dataset. The definition of playing technique may also overlap depending on the player or singer performing it, e.g. trill and vibrato in the VocalSet (Wilkins et al., 2018).

Chapter 6

Scattering Transform Applications

We proposed the adaptive scattering in Chapter 4 for recognising pitch modulation-based techniques (PMTs), and modified the joint time–frequency scattering (JTFS) in Chapter 5 for recognising pitch evolution-based techniques (PETs), respectively. In this chapter, we go beyond playing technique recognition and music signal analysis, to further verify the applicability of the scattering representations to other problems and domains. We apply the proposed adaptive time scattering (AdaTS, see Section 4.2) and the standard time scattering (StaTS, see Subsection 4.4.4) of detected vibratos for performer identification in polyphonic orchestral music in Section 6.1. This is motivated by the finding in Chapter 4 that these scattering representations capture the characteristic information of vibratos. Section 6.2 investigates the JTFS for detecting and classifying chick calls, which is inspired by the observation that chick calls exhibit similar spectro-temporal patterns as the PETs investigated in Chapter 5. Section 6.1 is an improvement of the paper Zhao et al. (2021) and Section 6.2 is an extension of our recently submitted work Wang et al. (submitted) (see Section 1.4 for publication details).

6.1 Adaptive and Standard Time Scattering for Performer Identification

Identifying performers from polyphonic music is a challenging task in music information retrieval (MIR). Being ubiquitous in both singing and harmonic instrument playing, vibratos have been used for singer identification (Kroher and Gómez, 2014; Nwe and Li, 2007) while being less explored for performer identification in instrumental music.

The idea of this section is inspired by our recently published work Zhao et al. (2021), which developed two violinist identification systems based on vibrato features: the AdaTS system and the VF-DS system. The former takes the adaptive time scattering (AdaTS) features as input to a support vector machine classifier and the latter system uses vibrato feature distribution similarity (VF-DS), for violinist identification. From the comparison of the identification results, we find that both methods outperform a random baseline, with accuracy improvement of 19.8% and 13.8%, respectively. The VF-DS method is more explainable than the AdaTS one in that the former uses four low-level features, i.e., average vibrato rate, average vibrato extent, standard deviation of vibrato rate, and standard deviation of vibrato extent, while the feature dimensionality of the latter is 154. On the flip side, the VF-DS method requires a certain number of vibrato notes for feature distribution calculation and is based on the assumption that vibrato segments are available. The AdaTS method is more flexible as long as there is vibrato in the music and can be potentially fully automatic because the AdaTS itself is a vibrato detector. This is the main idea that we investigate in this section: automating the AdaTS performer identification system in Zhao et al. (2021) using the vibrato detection system that we developed in Chapter 4.

Specially, there are five improvements of the performer identification system proposed in this section as compared to that developed in Zhao et al. (2021): (1) The system in this section is fully automatic as we first detect vibratos from full-length recordings and then use the AdaTS features of these detected vibratos for performer identification; (2) We compare different decomposition trajectories for the AdaTS besides the predominant melody trajectory; (3) We created annotations for all vibratos that are present in the recordings apart from the vibrato notes used in Zhao et al. (2021); (4) We explain in detail the reason for the poor performance of the identification systems in Zhao et al. (2021); (5) We explore also the standard time scattering (StaTS) (see Subsection 2.5.1) for this task and compare its performance with that of the AdaTS.

In summary, this section proposes an automatic system using detected vibratos for violinist identification from commercial orchestral recordings. We detect vibratos using the proposed AdaTS and the StaTS presented in Section 4.2 and Subsection 4.4.4, respectively. The scattering features of these vibratos are then used as the input to a machine learning classifier for identifying violinists. To the author's knowledge, this is the first attempt to identify violinists in polyphonic music based on the features of automatically detected vibratos only; and the first work to apply the StaTS for performer identification.

6.1.1 Background

The diversity of musical expression depends highly on the performers' interpretation, which may come from playing techniques, articulation, tempo variation, dynamics, and timbre (Juslin and Laukka, 2003). Jung (2007) analysed playing styles of three famous violinists, and argued for instance that the performances of Jascha Heifetz can be described as "unemotional" and "cold", whereas that of David Oistrakh always make listeners feel "warm" and rich in emotion. Among these factors, vibratos play an essential role in the performance of singing voice, flute and bowed-string instruments, and are frequently used to enhance selected notes and make them more prominent (Palmer and Hutchins, 2006). The information contained in vibratos provides useful clues of the performer's identity.

Most prior research on violinist identification used features such as pitch, timing, energy or vibrato amounts in a music piece, without considering detailed vibrato characteristics. Ramirez et al. (2011) built a Celtic violinist classifier taking as input the extracted pitch, timing, and amplitude features that represent both note-level characteristics and broader musical context. Molina-Solana et al. (2010) proposed an approach for identifying violinists in monophonic audio recordings using a musical trend-based model. Shih et al. (2017) used articulation and energy features to compare different playing styles of Jascha Heifetz and David Oistrakh. Additionally, previous research work on vibrato analysis focused mostly on monophonic music (Yang, 2017; Pang and Yoon, 2005). In this thesis, we develop an automatic system to identify violin performers using the proposed AdaTS and the StaTS features discussed in Section 4.2 and Subsection 4.4.4, which can be potentially applied to polyphonic scenarios.

6.1.2 Dataset

The dataset studied is a subset of the Violin dataset presented in Zhao et al. (2021), which comprises a collection of violin concertos from commercial CDs. A concerto is a musical work that focuses on a solo instrument, such as the violin or piano, accompanied by an orchestra. The complete Violin dataset has 45 recordings with a total duration of 26 hours. The recordings are 5 concertos written by five well-known composers: Ludwig van Beethoven, Johannes Brahms, Felix Mendelssohn, Jean Sibelius, and Pyotr Ilyich Tchaikovsky. Each concerto has 3 movements and is performed by 9 master violinists: Jascha Heifetz, Anne Sophie Mutter, David Oistrakh, Itzhak Perlman, Pinchas Zukerman, Isaac Stern, Salvatore Accardo, Yehudi Menuhin and Maxim Vengerov. Table 6.1 lists the composer and the name of each concerto in the Violin dataset.

Composer	Concerto name
Ludwig van Beethoven	Violin Concerto in D major, Op.61
Johannes Brahms	Violin Concerto in D major, Op.77
Felix Mendelssohn	Violin Concerto in E minor, Op.64
Pyotr Ilyich Tchaikovsky	Violin Concerto in D major, Op.35
Jean Sibelius	Violin Concerto in D minor, Op.47

Table 6.1: Composer and name of each concerto in the Violin dataset.

As a proof-of-concept study in this thesis, we use only the second movement of each concerto for computation saving, which forms a subset of 6 hours. Hereafter, we refer to this subset as the *Violin-II dataset*. To evaluate vibrato detection performance prior to the performer identification stage, all the vibratos in the Violin-II dataset were annotated by the author of the thesis. The total duration of the annotated vibratos is 1.7 hours. Figure 6.1 displays the number of annotated vibratos for each performer in each of the 45 concertos in the Violin-II dataset. As can be seen, the number of vibratos for each performer in the same piece are approximately balanced.

6.1.3 Vibrato detection

In Chapter 4, we compared the proposed adaptive scattering with other scattering representations for PMT recognition. The adaptive scattering achieves the best overall performance in the CBFdataset

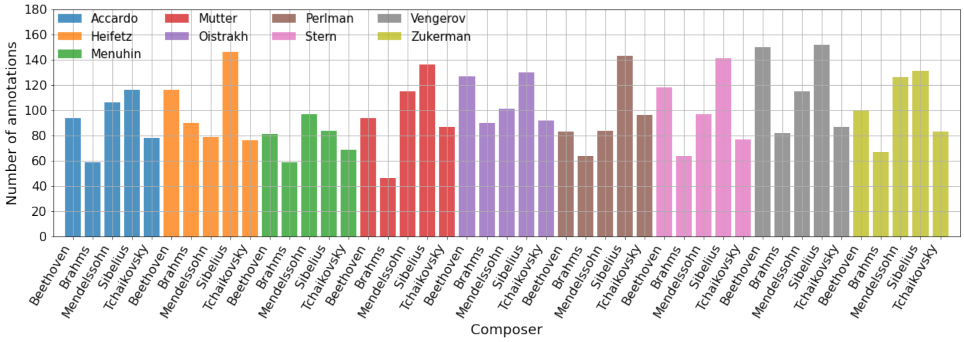


Figure 6.1: Number of of annotated vibratos in each recording for each performer in the Violin-II dataset. The correspondence of the concerto name to each composer is shown in Table 6.1.

(see Section 3.4), which comprises monophonic Chinese bamboo flute recordings. For the polyphonic music investigated in this section, the error-prone stage of decomposition trajectory extraction motivates us to explore other scattering representations for vibrato detection and performer identification.

Adaptive and standard time scattering

Similarly to Subsection 4.2.2, for the adaptive scattering, we investigate the performance of three decomposition trajectories: dominant band, predominant melody, and fundamental frequency (F0). Specifically, we explore in this section the adaptive time scattering (AdaTS) which achieved F-measure scores similar to that of the AdaTS+AdaTRS for PMT recognition (see Subsection 4.4.5) where the former is half in dimensionality to the latter. Decomposing frequency bands around these trajectories should capture the characteristic information of vibratos: rate, extent, and shape. These are useful cues for both detecting vibratos and characterising performers.

We extract the dominant band trajectory, i.e., the frame-wise frequency bands with maximum acoustic frequency energy, from the first-order scattering transform, and localise it to the scalogram. During the extraction process, we limit all trajectories to the frequency range of the violin: G3 to A7 (196 to 3520 Hz). This is to increase the possibility that the decomposition trajectories will correspond to one of the harmonic partials of the violin, rather than to frequencies produced by the accompanying instruments, which may have a different range of frequencies. To localise predominant melody and F0 trajectories in the scalogram, we allocate the bands with the closest frequency values to both cases. Since these two trajectories have higher temporal resolution (3 ms) than that of the dominant band trajectory (186 ms), we use the median value of the predominant melodies and F0s per 62 frames. This enables a consistent frame size of 186 ms for the AdaTS features calculated from different decomposition trajectories.

The top subfigures of Figure 6.2 to Figure 6.4 show the extracted dominant band, predominant melody, and F0 trajectories, respectively, in the log-frequency spectrogram of an example excerpt from the second movement of *Violin Concerto in D minor, Op.47* performed by Isaac Stern. As can be seen, the dominant band trajectory suffers from octave jumps (frequency switches between harmonic partials) while the other two trajectories exhibit missing estimations (false negatives), likewise the observations in Subsection 4.2.2. We preprocess all decomposition trajectories by linearly interpolating the missed estimations using the values of neighbouring frames and also smooth the dominant band trajectory by a median filter to reduce the effect of octave jumps. The blue solid line in Figure 6.2 top is the preprocessed dominant band trajectory, which is much more stable than the non-preprocessed one in yellow dotted line. Because the predominant melody and fundamental frequency trajectories are preprocessed only by linear interpolation, for better visualisation, we show these two trajectories in the original form, in Figure 6.3 top and Figure 6.4 top, respectively.

As observed from the top subfigures of Figure 6.2 to Figure 6.4, all three decomposition trajectories are not accurate even after limiting them to the tonal range of violin and preprocessing the trajectories by smoothing and interpolation. A typical error is that the frequency partials of accompaniment are incorrectly detected as the decomposition trajectory of the violin, for example, the regions from 0 to 4 sec and from 14 to 17 sec for all trajectories. Such erroneous trajectory estimations would substantially influence the downstream tasks, i.e., vibrato detection and performer identification, where the features used are both decomposed from the extracted trajectories. Even when we use the extracted trajectory of annotated vibratos, for example, the vibrato segments of the two regions above, the AdaTS features calculated provide misleading information to our performer identification system

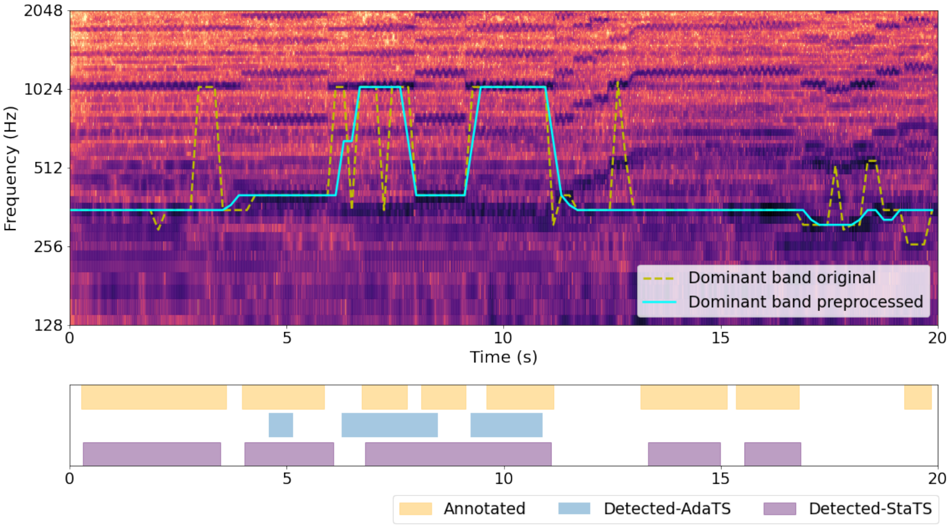


Figure 6.2: Vibrato detection result for an excerpt in the second movement of *Violin Concerto in D minor, Op.47* performed by Isaac Stern. Top: original and preprocessed dominant band trajectories of the adaptive time scattering (AdaTS) in the log-frequency spectrogram. Bottom: comparison of annotated vibrato segments with those detected by the AdaTS and the standard time scattering (StaTS, see Subsection 4.4.4).

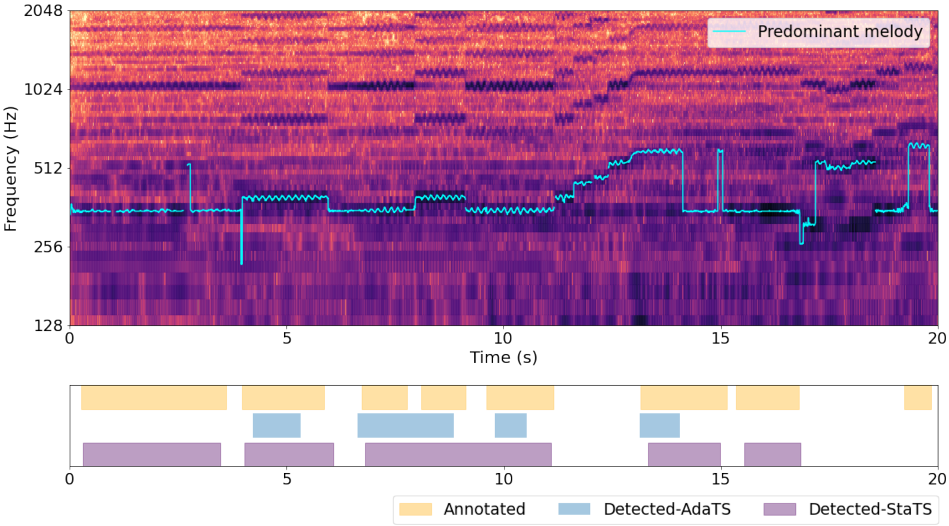


Figure 6.3: Vibrato detection result for the same excerpt in Figure 6.2. Top: predominant melody trajectory of the adaptive time scattering (AdaTS) in the log-frequency spectrogram. Bottom: comparison of annotated vibrato segments with those detected by the AdaTS and the standard time scattering (StaTS).

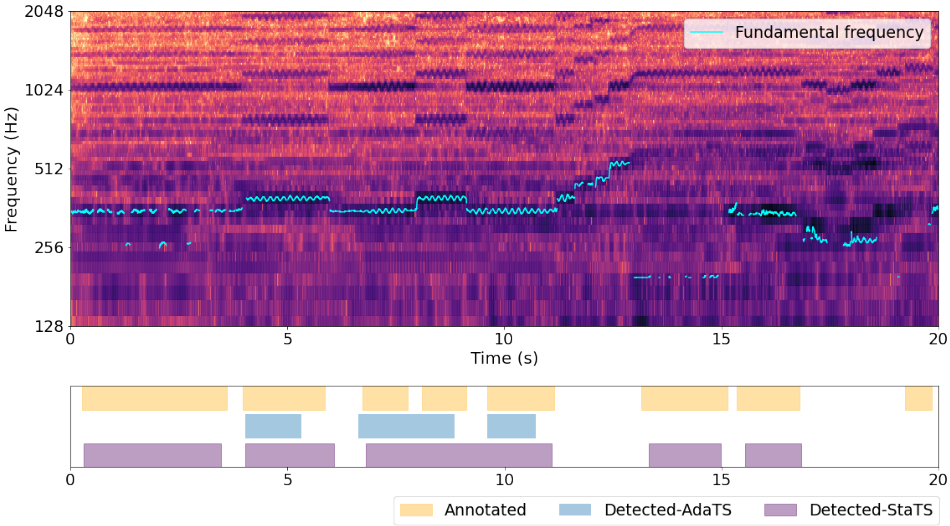


Figure 6.4: Vibrato detection result for the same excerpt in Figure 6.2. Top: fundamental frequency trajectory of the adaptive time scattering (AdaTS) in the log-frequency spectrogram. Bottom: comparison of annotated vibrato segments with those detected by the AdaTS and the standard time scattering (StaTS).

because of the incorrect decomposition trajectories.

The problem of the error-prone decomposition trajectory extraction motivates us to consider other scattering representations discussed in Subsection 4.4.4 that do not depend on a decomposition trajectory: the frequency-averaged time scattering, the StaTS, and the StaTS with principal component analysis. These representations decompose all frequency bands in the scalogram at the expense of having a higher feature dimensionality. We compare in this section the performance of the StaTS with that of the AdaTS. Although the StaTS underperforms the proposed adaptive scattering for detecting vibratos from monophonic Chinese bamboo flute recordings, for the polyphonic music explored in this section, it skips the erroneous stage of decomposition trajectory extraction and may achieve better results than the AdaTS.

Vibrato detection

We extract the AdaTS features by setting hyperparameters according to the characteristic information of vibratos discussed in Section 4.1: rate, extent, and shape of the modulation unit. Since the vibrato rate in violin music ranges from 2 to 15 Hz (Zhao et al., 2021), we use an averaging scale $T = 2^{15}$ (in samples; corresponding to 743 ms at a sampling rate of $F_s = 44.1$ kHz). $Q_1^{(\mathrm{t})}$ are the filters per octave of the temporal filterbank in the first-order time scattering; $Q_1^{(\mathrm{t})} = 16$ filters per octave are applied to capture modulations smaller than one semitone. Filters per octave in the second-order time scattering $Q_2^{(\mathrm{t})} = 4$ are used due to the less oscillatory nature of the signal to be decomposed at this order. $L \geqslant 1$ is the number of frequency bands symmetrically centred at the decomposition trajectory in the scalogram. We use $L = 7$ according to experimental results. The range M (in Hz) of the modulation rate is useful to extract the core part of the scattering coefficients that contain characteristic information of vibrato. Setting M into an interval, the AdaTS extracts only the coefficients corresponding to this range. An interval larger than the modulation rate range provides some harmonics in the modulation representation. We use $M = [0, 100]$ together with $Q_1^{(\mathrm{t})} = 16$ and $Q_1^{(\mathrm{t})} = 4$, which results in a feature dimensionality of 133. The frame size h (in samples) is inversely log-proportional to the oversampling parameter α (see Subsection 4.3.1), whereby $h = T/(2^{\alpha} F_s)$. We use $\alpha = 2$ for all the experiments which corresponds to a frame size of $h = 186$ ms. The StaTS feature is extracted in the same way but with fewer hyperparameters, i.e., $T = 2^{15}, Q_1^{(\mathrm{t})} = 16, Q_2^{(\mathrm{t})} = 4, \alpha = 2$. The resulting frame size of the StaTS is the same as that of the AdaTS while the feature dimensionality of the former is 2074, higher than that of the latter.

To detect vibratos from full-length recordings, we build a binary classifier as introduced in Subsection 4.3.2. The classifier takes as input the AdaTS or the StaTS features and outputs frame-wise labels of vibratos or non-vibratos. We fuse neighbouring frames with the same labels into vibrato events and postprocess the events using gap filling and minimum duration pruning. The scattering features of these detected vibratos are then reused for identifying violinists in Subsection 6.1.4. For both vibrato detection and performer identification stages, we use support vector machines (SVMs) (Hastie et al., 2009) with Gaussian kernels as classifiers due to their good generalisability based on a limited amount of training data (Albu and Martinez, 1999). The hyperparameters to be optimized are the error penalty parameter C and the width of the Gaussian kernel γ. We use consistent parameter grids of $10^{\{0:1:2\}}$ and $10^{\{-4:1:-2\}}$ for C and γ, respectively, during training, and select the best

SVM hyperparameters for testing. All features are z-score normalised.

Both at the vibrato detection and the performer identification stages, we use piece-informed data splitting. We divide the Violin-II dataset (see Subsection 6.1.2) into training and test sets by leaving one piece out and evaluate 5 splits. This guarantees that there is no overlap of pieces in the training and test sets in each split. Within each split, we run a 3-fold cross-validation, sampling on the training dataset in a way that ensures each fold includes approximately the same ratio of positive and negative class instances for vibrato or for a given performer identity. This is to avoid the cases that there is no example or there are too few examples of a given playing technique or performer class in the validation set if we further split the training set based on piece.

The bottom subfigures of Figure 6.2 to Figure 6.4 display the annotated vibrato segments and the detected ones by the system using the AdaTS and the StaTS features. The blue segments in the middle of each bottom subfigure are the segments detected by the AdaTS using the decomposition trajectory (blue line) in the corresponding top subfigure. As can be observed, for the AdaTS, only when the decomposition trajectory is correctly estimated, is there a possibility for the vibrato to be detected. For the StaTS, the vibrato detection is more accurate because it does not depend on a decomposition trajectory and potentially preserves the vibrato information in the representation.

We evaluate the vibrato detection system using frame-based precision $\mathcal{P}$, recall $\mathcal{R}$, and F-measure $\mathcal{F}$ introduced in Subsection 2.2.3. Table 6.2 shows the vibrato detection results in the Violin-II dataset using the AdaTS and the StaTS features. The StaTS improves the F-measure by 36%, 24%, and 28% as compared to the AdaTS calculated from the three decomposition trajectories, i.e., F0, predominant melody, and dominant band, respectively. The additional vibratos detected by the StaTS provide the classifier an opportunity to learn the characteristics of each violinist's performance. We compare the identification performance of the AdaTS and the StaTS in Subsection 6.1.4.

6.1.4 Performer identification

Taking the AdaTS and the StaTS features of the detected vibratos as input, we build in this subsection a performer identification system that outputs performer identity. We split each recording in the test set into

Method	Trajectory	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
AdaTS	F0	58	25	35
	Predominant melody	67	36	47
	Dominant band	61	34	43
StaTS	/	**82**	**63**	**71**

Table 6.2: Results of vibrato detection in the Violin-II dataset using the proposed adaptive time scattering (AdaTS) and the standard time scattering (StaTS). The former is calculated from three decomposition trajectories for comparison: fundamental frequency (F0), predominant melody, and dominant band. $\mathcal{P}$, $\mathcal{R}$, and $\mathcal{F}$ are frame-based precision, recall, and F-measure (all in %), respectively.

3 segments and use the detected vibratos in each segment to identify violinists. This is because our dataset is short of piece diversity with only 5 different concertos. Splitting the full recordings into segments increases the number of samples for each performer in the test set, which may improve the robustness of the system. Segments from different recordings may vary in duration due to the unequal length of the recordings. SVMs take the AdaTS or the StaTS features of the detected vibratos as input and output frame wise performer identity. To obtain segment level labels, each segment is labelled based on the majority vote of the frame labels of the detected vibratos. To analyse the influence of vibrato detection accuracy on the identification performance, we compare the identification results based on detected vibratos and annotated vibratos (ground truth). We evaluate the performer identification result using frame-based macro precision $\mathcal{P}$, recall $\mathcal{R}$, F-measure $\mathcal{F}$, and accuracy $\mathcal{A}$ introduced in Subsection 2.2.3. To quantify the variation of the results over performers, we also calculate the standard deviation $\mathcal{S}$ of the F-measures of all performers.

Baseline

To the author's knowledge, there is not yet any prior work using only the features of automatically detected vibratos for violinist identification in polyphonic music. We compare the proposed system with the vibrato feature distribution similarity (VF-DS) method in Zhao et al. (2021), which was applied to the Violin dataset (see Subsection 6.1.2). Note that there are three differences on the data used in this section and that in

Zhao et al. (2021). Firstly, the annotations in Zhao et al. (2021) include only stable vibrato notes while in this section, we annotate also the vibratos performed on variable pitches. Additionally, the annotations in Zhao et al. (2021) do not cover all the vibratos in the Violin dataset. Specifically for Violin-II dataset (see Subsection 6.1.2), the total duration of the annotated vibrato notes in Zhao et al. (2021) is 28.3 minutes, less than one third of the duration (1.7 hours) of the vibratos annotated in this thesis. Finally, as a pilot study, we consider only the second movement of each concerto in the thesis while Zhao et al. (2021) used the complete Violin dataset. This is because the computation of the latter was conducted only on the annotated vibrato notes, which forms a small part of the Violin dataset. In this section, we run the feature extraction to the full recordings for both vibrato detection and performer identification.

Different from the proposed performer identification system, the VF-DS method identified violinists by calculating the feature distribution similarity of annotated vibrato notes performed by different performers using the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951). The KL divergence is an asymmetric measure, quantifying the information loss when using the probability distribution Q to approximate the probability distribution P. Taking the extracted predominant melody (see Subsection 6.1.3) as input, the VF-DS method (Zhao et al., 2021) first calculated 4 vibrato features for each annotated vibrato note: average vibrato rate, average vibrato extent, standard deviation of vibrato rate, and standard deviation of vibrato extent. Vibratos were then selected by thresholding these features according to the characteristics of violin vibratos, for example, keeping only the vibratos with rate in the range of 2-15 Hz and extent of 10-50 cents (Zhao et al., 2021) . Histograms are then used to model the feature distributions of these vibrato notes performed by each violinist in the training set. In the test set, the feature distributions were modelled per movement (or concerto). After computing the KL divergence of the feature distributions of each movement (or concerto) with that of each performer in the training set, each test movement (or concerto) was labelled with the performer identify with the minimum KL divergence.

In this section, we use the VF-DS method as a baseline, applying it to both the annotated and the detected vibratos in the Violin-II dataset, and compare its performance with our proposed performer identification system. Note that the VF-DS method itself cannot detect vibratos. When applying it to detected vibratos, we use the same vibrato segments detected by the AdaTS based on the predominant band trajectory, which achieves the best vibrato detection performance among the three decomposition trajectories.

Results

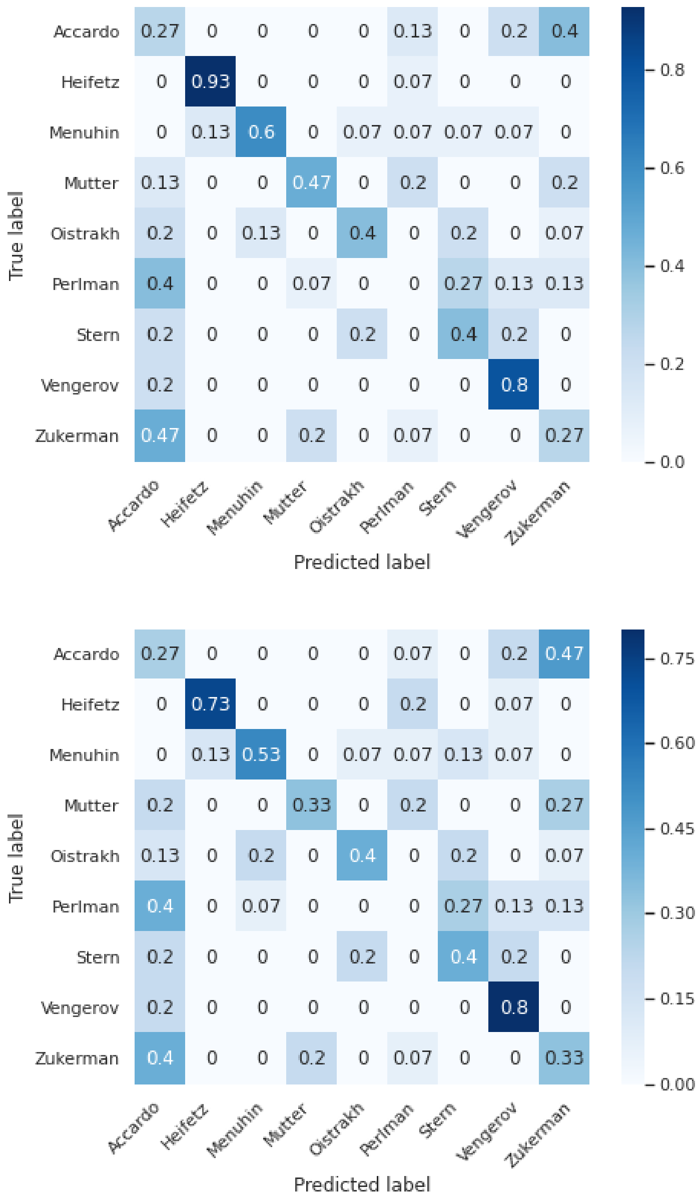
Table 6.3 displays the results of the proposed performer identification system using the AdaTS and the StaTS features, and that of the VF-DS baseline. The AdaTS features are computed from the three decomposition trajectories: F0, predominant melody, and dominant band. We compare the identification results from four fronts: performance of different methods, different decomposition trajectories of the AdaTS, results using annotated and detected vibratos, and the stability of the results over all performers for each method. Among the three methods—AdaTS, StaTS, and VF-DS—the StaTS outperforms the other two, with accuracy improved by more than 10% for all the cases: using either annotated or detected vibrato segments, or the AdaTS with different decomposition trajectories. As compared to the VF-DS method, the AdaTS achieves higher accuracy in both cases, i.e., using annotated and detected vibratos.

Comparing the results of the AdaTS features calculated from the three decomposition trajectories, we notice that the dominant band achieves the best performance using either ground truth or detected vibratos. The lowest accuracy of the AdaTS based on F0 trajectory is expected because the pYIN pitch estimation algorithm (Mauch and Dixon, 2014) that we use for F0 extraction was proposed for monophonic music signals, as discussed in Subsection 6.1.3. The StaTS and the VF-DS method exhibit better performance when using the feature calculated from annotated vibratos than those computed from detected vibratos. In contrast, the AdaTS yields higher accuracy when using the feature computed from detected vibratos. This may be attributed to the inaccurate decomposition trajectory extraction which may provide misleading information at the identification stage, as we investigated in Subsection 6.1.3. Although the StaTS achieves the highest macro F-measure, the standard deviation of the F-measures over performers are higher than other cases. This indicates that this method may be less stable as comparable to the AdaTS and the VF-DS methods.

Method	Decomposition trajectory	Vibrato annotated					Vibrato detected				
		$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{S}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{A}$	$\mathcal{S}$
AdaTS	F0	29	23	20	23	11	24	24	21	24	10
	Predominant melody	25	26	22	26	14	28	30	26	30	18
	Dominant band	36	27	26	27	12	30	32	27	32	19
StaTS	/	**48**	**46**	**46**	**46**	26	**45**	**42**	**43**	**42**	23
VF-DS	/	21	21	21	21	11	15	13	13	13	9

Table 6.3: Results of performer identification in the Violin-II dataset using the proposed adaptive time scattering (AdaTS), the standard scattering (StaTS), and the VF-DS baseline of detected and annotated vibratos. The AdaTS are calculated from the three decomposition trajectories for comparison: fundamental frequency (F0), predominant melody, and dominant band. $\mathcal{P}$, $\mathcal{R}$, $\mathcal{F}$, $\mathcal{A}$, and $\mathcal{S}$ are precision, recall, F-measure, accuracy, and the standard deviation of $\mathcal{F}$ across all performers (all in %), respectively. '/' means not applicable.

As discussed in Subsection 6.1.3, the main problem of the proposed system using the AdaTS features lies in the inaccurate decomposition trajectory extraction. Indeed, there are three stages in our proposed system: decomposition trajectory extraction, vibrato detection, and performer identification. We have evaluated the vibrato detection and the performer identification stages separately without assessing the first stage in the case of AdaTS. This is because we do not have ground truth for any of these trajectories. This may point out the importance of separating the target track prior to its decomposition trajectory extraction when applying the AdaTS to polyphonic music. Extracting the decomposition trajectory directly from the mixture would produce a large amount of false negatives, verified by the low recall scores in Table 6.2. Figure 6.5 top and bottom show the confusion matrices obtained by the StaTS computed from the annotated and the detected vibratos, respectively. High confusion between Salvatore Accardo and other performers is observed for both cases. Performer identity of no segments is recognised as Itzhak Perlman.



	Accardo	Heifetz	Menuhin	Mutter	Oistrakh	Perlman	Stern	Vengerov	Zukerman
Accardo	0.27	0	0	0	0	0.13	0	0.2	0.4
Heifetz	0	0.93	0	0	0	0.07	0	0	0
Menuhin	0	0.13	0.6	0	0.07	0.07	0.07	0.07	0
Mutter	0.13	0	0	0.47	0	0.2	0	0	0.2
Oistrakh	0.2	0	0.13	0	0.4	0	0.2	0	0.07
Perlman	0.4	0	0	0.07	0	0	0.27	0.13	0.13
Stern	0.2	0	0	0	0.2	0	0.4	0.2	0
Vengerov	0.2	0	0	0	0	0	0	0.8	0
Zukerman	0.47	0	0	0.2	0	0.07	0	0	0.27

	Accardo	Heifetz	Menuhin	Mutter	Oistrakh	Perlman	Stern	Vengerov	Zukerman
Accardo	0.27	0	0	0	0	0.07	0	0.2	0.47
Heifetz	0	0.73	0	0	0	0.2	0	0.07	0
Menuhin	0	0.13	0.53	0	0.07	0.07	0.13	0.07	0
Mutter	0.2	0	0	0.33	0	0.2	0	0	0.27
Oistrakh	0.13	0	0.2	0	0.4	0	0.2	0	0.07
Perlman	0.4	0	0.07	0	0	0	0.27	0.13	0.13
Stern	0.2	0	0	0	0.2	0	0.4	0.2	0
Vengerov	0.2	0	0	0	0	0	0	0.8	0
Zukerman	0.4	0	0	0.2	0	0.07	0	0	0.33

Figure 6.5: Normalised confusion matrices of performer identification in the Violin-II dataset based on the standard time scattering (StaTS) using ground truth vibratos (top) and automatically detected vibratos (bottom).

6.1.5 Discussion and conclusions

This section has proposed a fully automatic system for identifying performers in polyphonic orchestral music using the proposed adaptive time scattering (AdaTS) and the standard time scattering (StaTS) of detected vibratos. The system first detects vibratos from full-length recordings and the scattering features of these detected vibratos are then reused for performer classification. Comparing the performance of these two representations with the vibrato feature distribution similarity (VF-DS) baseline, the StaTS improves the accuracy by more than 10% than the other two, using either annotated or detected vibratos. The poor performance of the AdaTS results from its dependency on the extracted decomposition trajectory, an error-prone stage in the case of polyphonic music. This points out the limitation of applying the AdaTS directly to polyphonic music, which may suffer from extracting the harmonic partial of the target track from the mixture. However, the classification accuracy does demonstrate that the overall identification performance of the AdaTS is better than that of the VF-DS baseline. Accuracy values below 50% for all methods may be attributed to the lack of piece diversity of the current dataset, with only 5 pieces in total.

As future work to mitigate tracking difficulties, we will apply source separation techniques (Cano et al., 2018; Stöter et al., 2019) or use multi-pitch detection and instrument recognition methods that would assign a pitch to a specific instrument (Giannoulis and Klapuri, 2013) to obtain the target track prior to the AdaTS feature extraction. This is to ensure that the decomposition trajectory always corresponds to a harmonic partial of the target instrument. We could also make use of the score to improve the extraction accuracy of the decomposition trajectory (Devaney et al., 2012). Finding the solo performance of the same piece by the same violinists and comparing the performer identification results with that obtained in this section are also potential ways to verify the proposed methodology.

For the scattering representations that do not rely on a decomposition trajectory discussed in Subsection 4.4.4, we have only explored the StaTS. Another direction would be comparing the performance of the other scattering representations, such as the frequency-averaged time scattering and the StaTS with principal component analysis, with those used in this section for identifying performers in polyphonic music. Besides the abundance of vibratos, other pitch modulation-based techniques, such as tremolos and trills, are also frequently used in violin music. We can also extend the framework to performer identification using scattering features of multiple types of playing techniques. We currently label the performer identity at segment level by majority vote of its frame labels, which may not capture the temporal changes of vibrato rate and extent. For example, we observe that some performers start vibrato with a low rate, gradually increase it, and finally slow down. In this case, a majority vote of frame labels ignores such variations.

6.2 Joint Scattering for Chick Call Recognition

Similarly to the application of the time scattering representations to performer identification, this section investigates the applicability of the joint time–frequency scattering (JTFS) for recognising chick calls. This is motivated by the observation that chick calls exhibit similar spectro-temporal patterns as the pitch evolution-based techniques (see Section 5.1).

6.2.1 Background

Livestock farming is central for human sustainment. As farming technologies are booming, the large-scale and breeding-intensive poultry industries necessitate systems to automatically monitor the welfare of animals. Livestock vocalisations play a crucial part in such systems, for example, assessing laying hens' thermal comfort (Du et al., 2020), finding avian influenza-infected chickens to prevent the spread of diseases (Cuan et al., 2020), and detecting abnormal sound of broilers (Liu et al., 2020) as an early warning tool. Vocalisations produced by animals contain important information about their health, emotion, and behaviour (Briefer, 2012).

To the author's knowledge, there is limited computational research on analysing chick vocalisations and not yet any work on developing fully automatic systems for detecting and classifying chick calls. Collias and Joos (1953) grouped calls produced by young chicks into pleasure calls and contact calls (also known as distress calls), analysed the characteristics of these calls through displaying them on the spectrogram, and explored the common features of the sound signals that stimulate the production of each type of call. It was reported that contact calls are composed of descending frequencies only, are much louder, reach to lower frequencies, and are given at a slower rate; while pleasure calls perform the opposite, i.e., are composed of ascending frequencies, are much softer, start from higher frequencies, and are produced at a higher rate. Marx et al. (2001) analysed variations of chick sound patterns under successive changes in social isolation. Four types of chick calls were all labelled manually by visual inspection of the spectrogram: contact calls, short peeps, warbles, and pleasure calls.

The work in this section is a pilot study towards a fully automated system for real-time chick-robot interaction, where the robot interactively plays sounds appropriate to the context, for example, to attract, reassure, or calm down the chicks. It is a collaborative work between the author of the thesis and the Comparative Cognition Lab at the School of Biological and Behavioural Sciences of Queen Mary University of London. There are four aims with this project: (1) creating a dataset of chick call recordings and call type annotations; (2) developing an offline automatic system to detect chick calls in the dataset; (3) improving the detection system to automatically recognise the type of each chick call; (4) making the offline systems real-time, i.e., recognising chick calls and producing back calls to interact with the chick, and analysing the chick-robot interaction from the biological perspective. The author of this thesis is responsible for tasks (2) and (3), i.e., developing automatic systems for chick call detection and classification, which we introduce in Subsection 6.2.3 and Subsection 6.2.4, respectively. Since the content of the dataset provides useful context for the work, we describe the data collection and annotations in Subsection 6.2.2.

6.2.2 Dataset

As a pilot study[1], we collected data in laboratory conditions with a schematic shown in Figure 6.6. The arena is an open plastic box of size 60 cm × 92 cm × 52 cm. The walls of the arena are lined with white plastic, and the floor is lined with paper towels. We divided the arena into 5 regions—start, far, centre, close, and touch—and place the chick in the middle of the start region. A test stimulus (an imprinting object or a robot), was placed in the region furthest from the chick, as shown by the blue circle in the figure. The experiments used chicks from the Ross 308 strain of the species *Gallus gallus*. Chicks were hatched in darkness and in individual boxes so that they had no visual or tactile experience prior to the experiment. We placed one chick at a time in the arena within 12 hours after their hatching. The chick was transferred from the hatchery to the arena using a box of size 15 cm × 15 cm × 15 cm.

For each chick, we recorded their movements and sounds for around 10 minutes by a Microsoft LifeCam Studio Webcam and an AKG P170 microphone with a Behringer U-Phoria UMC204HD audio interface.

[1] All experiments in this study were approved by the Animal Welfare and Ethical Review Body (AWERB) committee at QMUL.

The camera was placed approximately 1 m above the centre of the arena; and the microphone was 1 m above the outer wall of the arena. All data was recorded at a sampling rate of 44.1kHz/16bits and we collected one recording for one chick at a time.

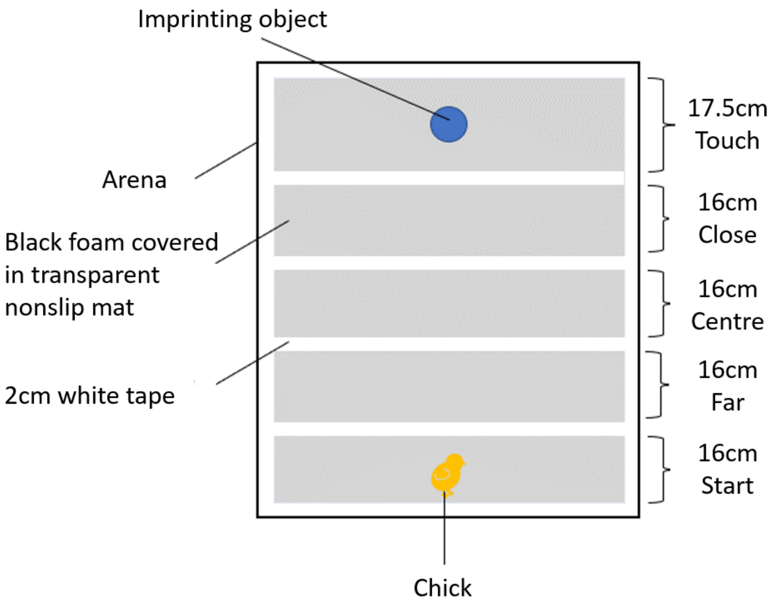


Figure 6.6: Schematic of the laboratory conditions for chick sound collection. This figure is contributed by the Comparative Cognition Lab, our collaborator in this project.

In this thesis, we develop a chick call detection system using the data collected from 12 chicks and a chick call recognition system based on the data of 4 chicks. This is due to the available annotations we have currently: start time and end time for 12 chicks and call type annotations for 4 of them. All the annotations were created by 2 experts experienced with chick sounds from the Comparative Cognition Lab at Queen Mary University of London. Three types of chick calls were annotated: pleasure, contact, and uncertain calls. Figure 6.7 top displays examples of the recorded pleasure and contact calls. As can be observed, pleasure calls are characterised by upward frequency changes, low energy, and short duration while contact calls exhibit the opposite, i.e., downward frequency changes, high energy, and long duration, which matches the findings in (Collias and Joos, 1953). Uncertain calls are those calls the annotators are not certain about. The whole repertoire of chick vocalisations is beyond the scope of this work. We introduce how the rapid frequency changes of pleasure and contact calls can be captured by the JTFS in Subsection 6.2.4.

6.2.3 Chick call detection

The number of calls is an indicator of the state of chicks (Marx et al., 2001). To facilitate real-time implementation, we develop separately a casual system for extracting chick call segments and for counting the total number of calls. The system first detects the onsets of chick calls and then extracts call segments by removing silence within each inter-onset interval.

Onset detection

The onsets of chick calls are detected using *SuperFlux* (Böck and Widmer, 2013), an algorithm outperforming the benchmark spectral flux method (Masri, 1996) for onset detection. The latter calculates the difference per frequency band in the magnitude spectrogram, sums up all positive changes over all bands, and selects the final onsets using peak-picking; while the former adds a maximum filter along the frequency axis before summing up the positive changes. This reduces the number of false positives originated from frequency modulations without missing onsets.

The implementation of the algorithm is based on the Librosa Python package (McFee et al., 2015), which takes the log-melspectrogram as input. We use 8 mel bands, ranging from 2048 to 6000 Hz. This is motivated by the observation in Figure 6.7 that the energy of both pleasure and contact calls concentrate over this frequency range. We evaluate the onset detection results using the *mir_ eval* Python library (Raffel et al., 2014). An onset is correctly detected when it falls in a tolerance window of 150 ms around that of the ground truth. Figure 6.7 top also displays the detected onsets (dotted lines) of chick calls in the spectrogram. The onset detection results for the 12 chicks, 7 males and 5 females, are shown in Table 6.4 with a total number of 8599 calls. We achieve recall scores above 90% for the onset detection of all chicks. The average precision, recall, and F-measure scores obtained are 84.3%, 97.0%, and 90.0%, respectively.

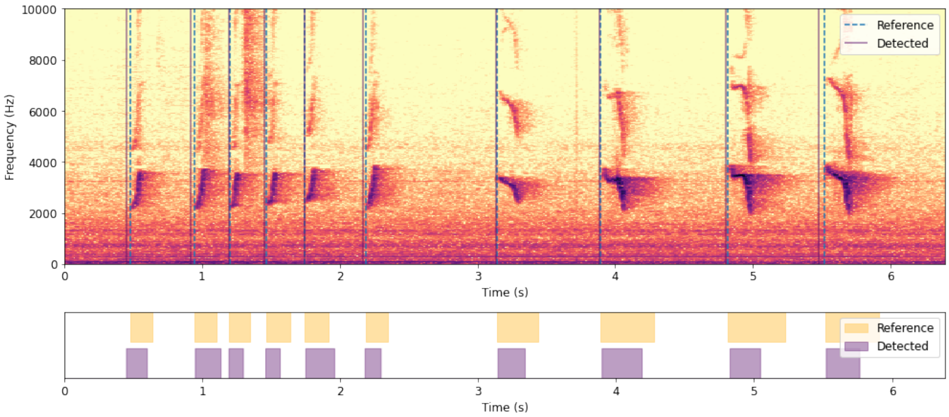


Figure 6.7: Example visualisation of chick call onset detection and segmentation results. Top: spectrogram with reference onsets in solid lines and detected onsets in dashed lines; bottom: comparison of reference call segments and detected call segments. For this example, frame-based $\mathcal{P}$=96%, $\mathcal{R}$=67%, and $\mathcal{F}$=79%.

Chick ID	#calls	Onset detection (*mir_eval*)			Segmentation (frame-based)			Segmentation (event-based)		
		$\mathcal{P}$(%)	$\mathcal{R}$(%)	$\mathcal{F}$(%)	$\mathcal{P}$(%)	$\mathcal{R}$(%)	$\mathcal{F}$(%)	$\mathcal{P}$(%)	$\mathcal{R}$(%)	$\mathcal{F}$(%)
85M	442	74	94	83	56	80	66	62	77	69
87M	1255	95	98	97	75	70	72	68	67	67
91M	525	77	97	86	61	86	71	65	81	72
21M	967	72	95	82	57	62	59	33	41	36
70M	647	79	98	87	58	62	60	50	61	55
32M	748	81	98	89	88	70	78	51	58	54
39M	987	98	99	99	97	79	87	94	95	94
89F	789	88	99	93	87	82	84	78	86	81
34F	707	88	97	92	45	46	46	46	47	47
41F	1052	92	93	93	78	76	77	80	80	80
48F	294	86	99	92	46	36	40	31	35	33
72F	186	81	98	89	35	48	40	42	47	44
T/A	8599	84.3	97.0	90.0	65.3	66.4	65.0	58.3	64.4	61.1

Table 6.4: Onset detection and segmentation results for the audio recording of each chick in the chick call dataset (M=male; F=female; T/A=total or average).

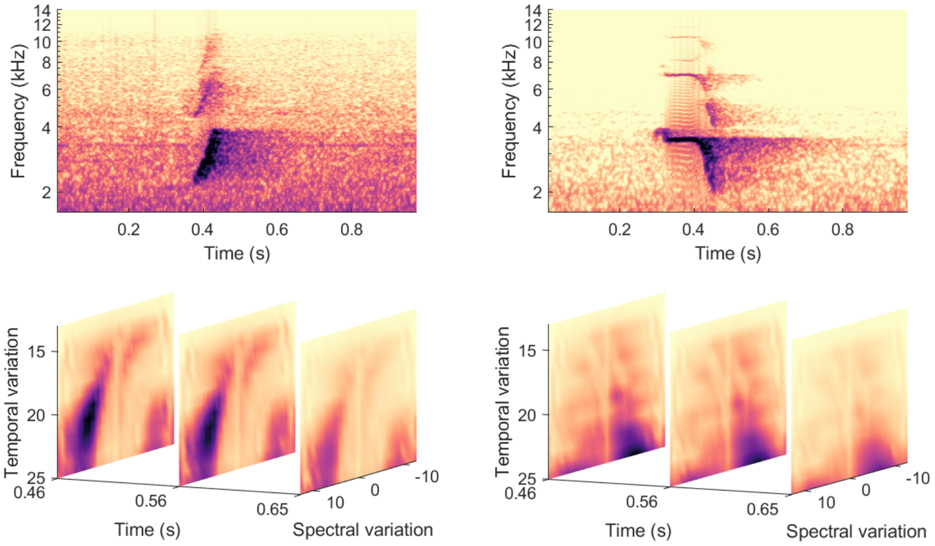
Segmentation

The intervals between two onsets include a large proportion of silence, as observed from Figure 6.7 top. We extract chick call segments from the inter-onset intervals by removing silence where the energy of the signal is below a certain threshold. A threshold of -30 dB is used based on experimenting with different values. Figure 6.7 bottom shows the comparison of the reference chick call segments and the output segments of our detection system.

We use both frame- and event-based precision $\mathcal{P}$, recall $\mathcal{R}$, and F-measure $\mathcal{F}$ introduced in Subsection 2.2.3 to evaluate the detected chick call segments. For event-based evaluation, we use the onset-offset method, i.e., a segment is correctly detected when its onset falls in a tolerance window of 200 ms around that of the reference call and its duration is at least 50% of the reference call duration. We first use the *mir_eval* Python library (Raffel et al., 2014) to calculate a maximum match of the onset times between reference and detected segments subject to the window constraint; and then compare the matched segments and reference in terms of duration. As shown in Table 6.4, average F-measures of 65.0% for frame-based evaluation, and 61.1% for event-based evaluation are obtained. Checking the detection errors from the audio, we find that the low F-measures for chicks 34F, 48F, and 72F result from the large proportion of pleasure calls. Some of these calls are too soft to be distinguished from other sounds such as steps of the chicks as they move in the experiment arena.

6.2.4 Chick call recognition

Motivated by the similar spectro-temporal patterns exhibited by chick calls and the portamento playing technique (see Section 5.1), we apply the joint time–frequency scattering (JTFS) features on the detected segments corresponding to chick calls for classification. Figure 6.8 shows the log-spectrograms and the frame-wise JTFS features of a pleasure call and a contact call. The directions of frequency changes are captured by the clear slopes, as shown by the bottom figures. For pleasure calls, the energy concentrates on the upward (left) side, while that of the contact call appears on the downward (right) side. Note that in Section 5.2 for portamento recognition, we use the JTFS of the direction with maximum spectro-temporal energy or the JTFS averaged over both directions, which we defined as the direction-invariant JTFS. Aiming at developing a general framework for playing technique recognition, we focus only on the playing technique type, e.g. regarding upward and downward portamenti as one class; while, in this section, chick calls with upward and downward frequency changes belong to different classes, i.e., pleasure calls and contact calls. Therefore, we use the JTFS of both directions for chick call classification in this section. Due to the limited call type annotations in our dataset, we include only 4 chicks for the evaluation of our recognition system. Table 6.5 lists the number of calls from each class, which are highly imbalanced.



(a) Pleasure call

(b) Contact call

Figure 6.8: Direction of frequency changes captured by joint time–frequency scattering (JTFS). Top: log-spectrograms of pleasure call and contact call; bottom: frame-wise JTFS features of pleasure call and contact call.

Chick	Pleasure	Contact	Uncertain	Total
85M	115	315	9	439
87M	613	492	146	1251
89F	47	681	60	788
91M	35	454	34	523
Total	810	1942	249	3001

Table 6.5: Number of pleasure, contact, and uncertain calls produced by each chick in the chick call dataset.

Recognition system

We propose a chick call recognition system with three different classification schemes:

- Scat-Only: in this scheme, a machine learning classifier takes the JTFS features of the whole recordings as input and outputs frame-wise call type labels; neighbouring frames with the same label are then fused into chick call events. Similarly to portamento technique recognition in Section 5.5, we postprocess the obtained chick call events by gap filling and minimum duration pruning. We fill the gaps between neighbouring events when the gaps are shorter than the shortest event in the training set; and prune the events that have smaller duration than the minimum duration event in the training set. The minimum duration is automatically calculated subject to the call type and the train-test split during recognition.

- Seg-Scat: this method builds upon the detection system introduced in Subsection 6.2.3. A machine learning classifier takes the JTFS features of the extracted chick call segments as input, outputs frame-wise labels, and assigns one label to each segment based on the majority vote of its frame labels.

- Seg-HLF: this scheme also uses the extracted chick call segments in Subsection 6.2.3 and assigns labels at segment level. Different from the Seg-Scat, it takes high level features (HLF) calculated from each segment as input and outputs one label per segment directly. This is a causal system proposed to facilitate real-time implementations.

All classification schemes use support vector machines (SVMs) (Hastie et al., 2009) with Gaussian kernels as classifiers due to their good generalisability based on a limited amount of training data (Albu and Martinez, 1999). In the recognition process, we conduct a subject-independent evaluation, i.e., splitting recording of one chick in the test set and recordings of the remaining 3 chicks in the training set. Four splits are conducted in a circular way to increase the validity of the results. Within each split, we run a 3-fold cross-validation, sampling on the training dataset in a way that ensures each fold includes approximately the same ratio of positive and negative class instances for a given chick call types. This is to avoid the cases that there is no example or there are too few examples of a give chick call class in the validation set if we further split the training set based on chick subject. The hyperparameters to be optimized are the error penalty parameter C and the width of the Gaussian kernel γ. We use consistent parameter grids of $10^{\{0:1:2\}}$ and $10^{\{-4:1:-2\}}$ for C and γ, respectively, during training, and select the best SVM hyperparameters for testing. Both frame- and event-based precision $\mathcal{P}$, recall $\mathcal{R}$, and F-measure $\mathcal{F}$ are used as the evaluation metrics. The frame size is 93 ms for frame-based evaluation in all classification schemes. For event-based evaluation, we use the onset-offset method with the same collar as that we use for the segmentation evaluation in Subsection 6.2.3. We introduce details of the feature extraction process for each classification scheme in the following paragraphs.

Scat-Only We calculate the JTFS features of the full recordings using the same hyperparameters as that we used for portamento technique recognition in Subsection 5.3.1, i.e., $T = 2^{14}, Q_1^{(\mathrm{t})} = 16, Q_2^{(\mathrm{t})} = 2, Q_1^{(\mathrm{f})} = 2, \alpha = 2, M = [0, 50]$ Hz, and a spectral averaging scale covering the entire log-frequency axis. This is because both pleasure and contact calls exhibit monotonic frequency changes, similar to that of the portamento technique. To account for temporal context, we calculate the mean and standard deviation of 5 frames centred at the current frame. The frame size and dimensionality of the input feature to the classifier are 92 ms and 850, respectively.

Seg-Scat Different from the Scat-Only method, this classification scheme first conducts a segmentation step to extract chick call candidates using the detection method in Subsection 6.2.3. The JTFS features are then calculated for each segment using the same hyperparameters above; therefore the frame size and dimensionality of the features are the same as that of the Scat-Only method.

Seg-HLF For the purpose of a real-time implementation, we develop a computationally efficient system based on the HLF to classify the chick call candidates extracted in Subsection 6.2.3. Both the Scat-Only and the Seg-Scat methods use the JTFS features only, which requires fine resolution filterbanks in the scattering transform to capture variation of the spectro-temporal patterns, for example, 16 filters per octave in the first-order time scattering here. When our focus is the call direction only rather than variation of the whole spectro-temporal pattern, fewer filters and a larger frame size can be used, which would potentially improve computational efficiency.

Motivated by the characteristics of chick calls (see Subsection 6.2.2), we propose 6 high level features for the Scat-HLF classification scheme: duration, direction, frequency range, average frequency slope, average frequency slope change, and number of high energy frequency bands. Therefore we obtain a 6-dimensional feature vector for each segment, which is calculated from the JTFS transform except the duration S of the extracted chick call segments. We define the other five high level features as follows:

1. **Direction** (D): Rather than taking one side of the JTFS features (see Figure 6.8) as the Scat-Only and Seg-Scat, the Seg-HLF sums up the energy on each side and returns only two values per time frame: $d_{\text{up}}(n)$ and $d_{\text{down}}(n)$. $d_{\text{up}}(n)$ and $d_{\text{down}}(n)$ roughly measure the confidence of upward and downward directions, respectively, where $n = 1, 2, ..., N$ is the n-th time frame and N is the total number of time frames in the call segment. The frame-wise direction information can be expressed as the difference $d(n) = d_{\text{up}}(n) - d_{\text{down}}(n)$. We take the average of the frame-wise direction information as the direction feature of the segment:

$$D = \frac{1}{N} \sum_{n=1}^{N} d(n). \tag{6.1}$$

   $D > 0$ suggests that the direction of the segment is upward; and $|D|$ is the confidence of upward as compared to downward directions. To have comparable results, we calculate the direction information using the same JTFS hyperparamters in all three classification schemes. Note that in real-world applications, fewer filters and larger frame size are recommended for the Scat-HLF classification scheme.

2. **Frequency range** (F_r): Besides duration and direction, we observe that the discriminative information between pleasure and

contact calls also exists in the frequency range, average frequency slope, average frequency slope change, and number of high energy frequency bands, as observed from Figure 6.8. This information can be extracted from the first-order scattering transform. To have a finer temporal resolution, we use a smaller $T = 2^{12}$ and calculate only the first-order time scattering coefficients, which results in a frame size of 23 ms. For each call candidate, we extract the dominant frequency bands in the first-order time scattering and denote the corresponding centre frequency as $f_{\text{dom}}(n)$. The frequency range of the call is calculated as

$$F_r = \max\Big(f_{\text{dom}}(n)\Big) - \min\Big(f_{\text{dom}}(n)\Big), \tag{6.2}$$

with the unit Hz, where n=1,...,N.

3. **Average frequency slope** (F_{as}): We define the frame-wise frequency slope $f_{\text{slope}}(n)$ as the first-order difference of the dominant frequency:

$$f_{\text{slope}}(n) = \frac{f_{\text{dom}}(n) - f_{\text{dom}}(n-1)}{h}, \tag{6.3}$$

with the unit Hz/s, where n=2,...,N and h is the frame size. The average frequency slope of the call segment is then

$$F_{\text{as}} = \frac{1}{N-1}\sum_{n=2}^{N} f_{\text{slope}}(n). \tag{6.4}$$

4. **Average frequency slope change** (F_{asc}): The frame-wise frequency slope change $f_{sc}(n)$ is defined as

$$f_{sc}(n) = \frac{f_{\text{slope}}(n) - f_{\text{slope}}(n-1)}{h} \tag{6.5}$$

with the unit Hz/s^2, where n=3,...,N. We use the average to estimate the slope change over time for the call segment

$$F_{\text{asc}} = \frac{1}{N-2}\sum_{n=3}^{N} f_{sc}(n). \tag{6.6}$$

5. **Number of high energy frequency bands** (N_h): We count the

number of high energy frequency bands (N_h) from the first-order time scattering where the energy of the band is more than 50% of dominant band energy.

Baseline

We compare the JTFS features with commonly used spectral features, the mel-frequency cepstral coefficients (MFCCs) (Abeßer et al., 2010), for chick call recognition. Similarly to the proposed recognition system, we conduct two classification schemes for the MFCCs, i.e., MFCC-Only and Seg-MFCC. The former calculates the MFCCs of the full recordings, outputs frame-wise labels of chick call type, and fuses neighbouring frames with the same labels into chick call events. The latter first conducts a segmentation step to extract chick call candidates using the detection method in Subsection 6.2.3 and then extracts the MFCCs for each segment. The frame size and dimensionality of the MFCC features are 25 ms and 24, respectively.

Results

Table 6.6 and Table 6.7 list the frame- and event-based F-measures for chick call recognition in different classification schemes using the scattering features and the MFCCs: Scat-Only, Seg-Scat, Seg-HLF, MFCC-Only, and Seg-MFCC. Note that only the classification schemes with ‘Seg-’ use features calculated from the detected chick call segments (see Subsection 6.2.3) while the Scat-Only and the MFCC-Only schemes recognise chick calls directly from full-length recordings. To show how the onset detection and segmentation performance affect the ‘Seg-’ classification schemes, we also compare the recognition results using the annotated segments to those using the detected segments. ‘A-D’ in tables refers to the averaged F-measure over all chicks using the detected segments while ‘A-A’ corresponds to that using the annotated segments.

We compare the recognition results in Table 6.6 and Table 6.7 from four fronts: the scattering features versus the MFCCs; the three classification schemes using the scattering features; the recognition results using detected versus annotated chick call segments; and the results for different types of chick calls. Comparing the Scat-Only with the MFCC-Only method, we observe that they achieve comparable results in the frame-based evaluation, both with macro F-measures of 32.1%; while

Chick	Scat-Only			Seg-Scat			Seg-HLF			MFCC-Only			Seg-MFCC		
	P	C	U	P	C	U	P	C	U	P	C	U	P	C	U
85M	14	77	14	18	76	8	25	68	0	4	84	0	7	63	0
87M	28	77	11	32	79	25	31	75	8	4	73	5	0	75	11
89F	0	85	0	14	88	19	11	87	0	8	90	0	9	89	0
91M	0	79	0	7	85	8	8	81	0	3	88	26	3	81	0
A-D	10.5	79.5	6.3	17.8	82.0	15.0	18.8	77.8	2.0	4.8	83.8	7.8	4.8	77.0	2.8
A-A	/	/	/	60.3	95.3	22.0	56.0	91.3	0.8	/	/	/	17.0	91.0	8.3
A-D (MF)		32.1			38.3			32.9			32.1			28.2	
A-A (MF)		/			59.2			49.4			/			38.8	

Table 6.6: Frame-based results of chick call recognition based on detected and annotated call segments using the proposed and the baseline methods: (1) Scat-Only: detect and classify chick calls using the JTFS features only; (2) Seg-Sat: segment audio into chick call candidates and classify each segment using the JTFS features; (3) Seg-HLF: segment audio into chick call candidates and classify each segment using high level features (HLF); (4) MFCC-Only: detect and classify chick calls using the MFCCs only; (5) Seg-MFCCs: segment audio into chick call candidates and classify each segment using the MFCCs. P, C, and U represent pleasure, contact, and uncertain calls, respectively. A-D refers to the averaged recognition result over all chicks using the detected chick call segments while A-A corresponds to that using the annotated chick call segments (MF = macro F-measure and all the other numbers are F-measure scores; '/' = not applicable).

Chick	Scat-Only			Seg-Scat			Seg-HLF			MFCC-Only			Seg-MFCC		
	P	C	U	P	C	U	P	C	U	P	C	U	P	C	U
85M	11.5	45.9	0	19.7	51.4	10.0	21.9	58.4	0	2.3	60.3	0	6.5	42.2	0
87M	16.7	36.4	5.6	30.0	42.0	18.0	26.0	47.8	4.5	3.4	47.8	5.5	0.6	46.6	9.7
89F	0	21.2	0	12.1	70.0	15.8	9.4	76.4	0	4.3	66.0	0	9.2	66.0	0
91M	0	15.6	0	11.0	63.1	10.3	9.5	65.0	0	0	69.6	26.7	2.5	60.3	0
A-D	7.0	29.8	1.4	18.2	56.6	13.5	16.7	61.9	1.1	2.5	60.9	8.0	4.7	53.8	2.4
A-A	/	/	/	60.5	88.9	17.3	58.6	82.8	0.7	/	/	/	15.1	80.4	8.2
A-D (MF)		12.7			29.4			26.6			23.8			20.3	
A-A (MF)		/			55.6			47.4			/			34.6	

Table 6.7: Event-based results of chick call recognition based on detected and annotated call segments using the proposed and the baseline methods: (1) Scat-Only: detect and classify chick calls using the JTFS features only; (2) Seg-Sat: segment audio into chick call candidates and classify each segment using the JTFS features; (3) Seg-HLF: segment audio into chick call candidates and classify each segment using high level features (HLF); (4) MFCC-Only: detect and classify chick calls using the MFCCs only; (5) Seg-MFCCs: segment audio into chick call candidates and classify each segment using the MFCCs. P, C, and U represent pleasure, contact, and uncertain calls, respectively. A-D refers to the averaged recognition result over all chicks using the detected chick call segments while A-A corresponds to that using the annotated chick call segments (MF = macro F-measure and all the other numbers are F-measure scores; '/' = not applicable).

in event-based evaluation, the former underperforms the latter, with macro F-measures of 12.7% against 23.8%. For the comparison of the scattering features and the MFCCs using the detected segments, both Seg-Scat and Seg-HLF outperform Seg-MFCC, with macro F-measures improved by 10.1% and 4.7%, respectively, in the frame-based evaluation; the corresponding improvements in the event-based evaluation are 9.1% and 6.3%, respectively.

Narrowing the scope with the recognition results using the scattering features—Scat-Only, Seg-Scat, and Seg-HLF—the Seg-Scat achieves the best performance in both frame- and event-based evaluation. The Seg-Scat increases the macro F-measure by 6.2% and 5.4% as compared to that of the Scat-Only and the Seg-Scat, respectively, in the frame-based evaluation; macro F-measure improvement of 16.7% and 2.8% appears correspondingly in the event-based evaluation. It is expected that the Scat-Only scheme achieves comparable frame-based but much lower event-based macro F-measures in contrast to the Seg-Scat method, where the latter makes use of chick call segments, either detected or annotated.

As can be seen from both tables, all 'Seg-' methods, i.e., Seg-Scat, Seg-HLF, and Seg-MFCC, exhibit much better performance using the annotated chick call segments as compared to using the detected chick call segments. In the frame-based evaluation, macro F-measures increase by 20.9%, 16.5%, and 10.6% for the three methods, respectively; the improvement is even higher in the event-based evaluation, with 26.2%, 20.8%, 14.3% correspondingly. This verifies the potential of the JTFS features (in the Seg-Scat scheme) or high level features (in the Seg-HLF scheme) for chick call classification. Inspecting the F-measures of each type of calls, we notice that all five methods exhibit better performance on contact call classification than that on recognising pleasure and uncertain calls. This may be attributed to the small amount of samples for the latter two types of calls. Yet, the scattering features are less sensitive to data imbalance as compared to the MFCCs. For example, the Seg-Scat and Seg-HLF achieve F-measures of 60.3% and 56.0% against 17.0% from the Seg-MFCC for recognising pleasures call in the frame-based evaluation, although all three methods have comparable results on contact call classification. The Seg-Scat method is the most robust to data imbalance.

6.2.5 Discussion and conclusions

As a pilot study aiming at developing a real-time interaction system for chick and robot, this section has proposed a chick call detection system and a chick call recognition system based on the joint time–frequency scattering (JTFS). This is motivated by the observation that chick calls have similar spectro-temporal patterns as the portamento playing technique, both exhibiting continuous frequency changes with temporal modulations, as shown in Figure 6.8 top and Figure 5.2 (a), respectively, We introduce three classification schemes for recognising chick calls: using the JTFS features of full recordings (Scat-Only); segmenting audio into chick call candidates and classifying each segment using the JTFS features (Seg-Scat); and, segmenting audio into chick call candidates and classifying each segment using high level features (Seg-HLF). The results show that the scattering features, either the JTFS or high-level features, outperform the MFCCs for recognising chick calls. The Seg-Scat method achieves the best result and is the most robust to data imbalance. The Seg-HLF underperforms the Seg-Scat method while the former may potentially reduce computation with less filters and a larger frame size in practical applications. The comparison between the classification results using detected and ground truth chick call segments verifies the potential of the JTFS feature for chick call recognition.

Four limitations exist in the current study. We conduct experiments in a laboratory where the collected data is much cleaner as compared to real living conditions of the chicks. For the latter case, prepossessing the recordings by a denoising method to remove noise prior to the application of the JTFS features would be useful. We place one chick at a time in the experiment arena while in a practical case there may be many chicks vocalising simultaneously. For example, a pleasure call produced by one chick may overlap with a contact call produced by another chick. In such cases, we could apply a source separation technique (Stöter et al., 2019) to separate the sound of multiple chicks before calculating the JTFS features. The other two limitations include the the small amount of data we have used for evaluating the recognition system and the highly imbalanced number of samples for each class.

One direction for future work will be using semi-supervised learning with limited annotations or unsupervised learning without annotations for recognising chick calls. The latter may also potentially discover new chick call patterns. Based on the detection and classification system, a close-loop interaction between a robot and the chicks (Lerch et al., 2011) could be designed. Using methods that handle imbalanced data may improve the recognition performance of chick calls. We could also compare the proposed systems to vocalisation detection and classification systems developed for other animals in the literature, for example, the broiler stress detection system proposed in Jakovljević et al. (2019).

Chapter 7

Conclusions and Future Work

In this thesis, we have proposed a general framework for playing technique recognition, with generalisability evaluated over different datasets with a variety of instrumental and vocal techniques, and applicability tested on additional audio classification problems. In this chapter, we summarise the contributions in Section 7.1, discuss the strengths and weaknesses of the proposed methodology in Section 7.2, and present possible directions for future research in Section 7.3.

7.1 Summary of Contributions

With limited data available for a classification problem, a possible way to guarantee performance without sacrificing generalisability is to find a compact and informative representation that removes variabilities irrelevant to the task. This thesis provides such a way based on the scattering transform for playing technique recognition, a problem suffering from the scarcity of data. Identifying the variabilities of playing techniques, we find that the irrelevant ones include time-shifts, time-warps, and frequency-transpositions, which are indicated by the independence of playing techniques to performer, piece, instrument, genre, and pitch (see Section 2.2).

The scattering transform is a flexible framework with various combinations of temporal and spectral wavelet decompositions, each combination providing certain invariance properties. Based on the spectro-temporal patterns of playing techniques, we propose the adaptive scattering and the direction-invariant joint time–frequency scattering (dJTFS), each for representing one family of playing techniques. The adaptive scattering is a variant of the scattering framework, providing representations such as the adaptive time scattering (AdaTS), the adaptive time–rate scattering (AdaTRS), and the combination of these two operators (AdaTS+AdaTRS). All these representations are invariant to large frequency-transpositions besides the invariance to time-shifts and time warps. These invariance properties are desirable for representing pitch modulation-based techniques (PMTs), a group of periodic modulations elaborated on stable pitches. The dJTFS differs from the adaptive scattering in that the former applies frequency scattering along the acoustic frequency axis, which captures spectro-temporal modulations. This fits the characteristics of pitch evolution-based techniques (PETs), a group of playing techniques exhibiting monotonic pitch changes. Besides the invariance properties, the adaptive scattering are more compact as compared to the standard time scattering; and the dJTFS captures the joint activation of PETs with the dimensionality reduced in half in contrast to the original joint time-frequency scattering.

The methodology is first tested on a newly created dataset of Chinese bamboo flute performances (CBFdataset), followed by evaluations over three additional datasets with a variety of instrumental and vocal techniques. For the verification on the former, we focused on seven commonly used playing techniques in music signals: vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando; and group them into PMTs (the first four) and PETs (the last three). We begin with designing two recognition systems with binary classification schemes, each focusing on one family of playing techniques (see Section 4.3 for PMT recognition and Subsection 5.5.1 for PET recognition), and set the scattering hyperparameters according to the characteristics of each type of playing technique. This is followed by another recognition system with a multiclass classification scheme (see Subsection 5.5.2) which classifies all playing techniques simultaneously and provides confusion matrices between playing techniques. The system with a multiclass classification scheme forms the prototype for evaluating the methodology on the additional datasets (see Section 5.6). All systems achieve comparable or better results compared to the start-of-the-art. We provide a formal interpretation of the role of each component in the scattering feature extractors, confirmed by explanatory visualisations.

Motivated by the invariance properties, we further test the applicability of the proposed representations to other audio classification problems. We first apply the adaptive time scattering and the standard time scattering for identifying violin performers in polyphonic orchestral music (see Section 6.1) using vibratos detected by our proposed playing technique recognition system. Another system is developed for detecting and classifying chick calls (see Section 6.2), such as pleasure and contact calls, using the joint time–frequency scattering. This is inspired by the observation that the chick calls exhibit similar spectro-temporal patterns as certain playing techniques.

As an endeavour to enrich the available data for analysing playing technique in context and for computational research on non-Western instruments, we create and publicly release the CBFdataset to the community, which is the first dataset on the Chinese bamboo flute (CBF). Containing full-length CBF performances and expert annotations of playing techniques, it can be used for computational music performance analysis for this particular instrument.

7.2 Discussion

Despite the potential of the proposed framework, we also present some other perspectives and limitations regarding the work conducted in this thesis. The introduction of PMTs and PETs presents a perspective of analysing playing techniques in groups. Techniques addressed are not limited to the seven playing techniques investigated. For example, the current work only considers four types of PMTs, i.e., vibrato, tremolo, trill, and flutter-tongue; the methodology is applicable to other periodic patterns, such as tonguing. The recognition performance on other types of playing techniques in the additional datasets validates the generalisability of the proposed methodology.

A consistent yet flexible taxonomy for playing techniques, either for instrumental or vocal techniques, is under-explored. Robustness to minor alternations of the taxonomy remains a challenge for playing technique recognition systems. The same technique may exist under a different name in the context of another instrument or genre. Take the playing techniques in the additional datasets (see Subsection 2.2.2) for instance, portamento in the VPset corresponds to glissando in the SOL dataset. The definition of a playing technique may also overlap depending on the player or singer performing it, for example, trill and vibrato in the VocalSet (Wilkins et al., 2018).

The experiments in this thesis all operate on the original datasets directly without any data augmentation. In real-world applications where the music may be processed in various ways, augmenting the data with sound effects such as reverberation and flanger (Ramires and Serra, 2019) may improve the robustness of the proposed recognition system. Another observation is that data imbalance may not significantly influence the detection results. The techniques with many more samples do not achieve better results, for example, the portamento technique in the CBFdataset and the straight technique in the VocalSet. This points to a robustness and compactness of the proposed representations to learn quickly the discriminative information based on a small number of examples.

We use in this thesis only the supporter vector machine classifier, other classifiers such as convolution neural network classifiers or recurrent classifiers, could be investigated. The recurrent classifiers like long short-term memory units (Warrick et al., 2019) account for temporal context, which might be useful for playing technique recognition. Since co-articulations form a small portion of the CBFdataset, we conduct single-label multiclass classification during the recognition. This is done by discarding the samples with more than one label. In practice, a user may expect a recognition system to detect all playing technique components in a co-articulation. In this case, a multilabel classifier should be the most appropriate choice with a modification of the evaluation metric simultaneously.

During the recognition of PMTs in Chapter 4, we notice that the dominant band trajectory exhibits instabilities resulting from octave jumps, i.e., frequency switches between harmonic partials. Preprocessing the trajectory by a median filter improves its stability but also smooths out short note changes bound with PMTs. To avoid the latter case, one may improve the system by decomposing a certain number of frequency bands with harmonic relationships to the dominant band. As we have investigated in Section 6.1, the adaptive scattering suffers from the inaccurate extraction of the decomposition trajectory when being applied to polyphonic music. The information loss at this error-prone stage cannot be recovered for subsequent processing. However, the entire pipeline is potentially applicable to polyphonic cases if we preprocess the music by a source separation technique (Stöter et al., 2019) or use a multi-pitch detection and instrument recognition method that would assign a pitch to a specific instrument (Giannoulis and Klapuri, 2013) to obtain the target track prior to the decomposition trajectory extraction. This may increase the possibility that the decomposition trajectory always corresponds to a harmonic partial of the target instrument.

7.3 Future Work

We summarise the potential directions for future research into two groups: improvement of the methodology itself and other possible applications of the methodology. Trainable scattering (Cotter and Kingsbury, 2019) is a potential future direction in the first group. In this thesis, we set the hyperparameters of the scattering representations motivated by the characteristics of the playing techniques in general. This is to develop an explainable and generalisable playing technique recognition framework that can be applied to other instruments and datasets directly without retraining. Indeed, this manual setting of scattering hyperparameters may not lead to an optimal performance for a specific dataset. As future work, we could automate this process by tuning the hyperparameters of the scattering transform and the classifier jointly for each type of playing technique, or for a specific instrument or genre. The explainability of the scattering transform may provide trainable scattering a promising way to develop explainable deep learning models for audio signals. Finding the meanings of the obtained hyperparameters from music perspective and identifying the change of regions in the representation caused by hyperparameter variation would boost our understanding of the task at hand. The proposed adaptive scattering and dJTFS representations are two examples of the possible variants of the scattering transform. The flexibility of the scattering transform suggests that another direction would be to expand the framework by developing new operators or adding other existing operators to make the system as general-purpose as possible. For example, the spiral scattering (see Subsection 2.5.2) which captures variations across harmonics, may provide useful information for the recognition of playing techniques characterised by harmonic variations, such as multiphonics.

The investigation of the scattering transform and the proposed playing technique recognition framework may benefit a wide range of applications. The invariance properties provided by the different scattering representations may be attractive to other music signal analysis tasks, such as music structure analysis, genre recognition, instrument recognition, and music transcription. There could be some future development work towards a system that provides an automatic transcription of playing techniques in a way that is easy to understand for a user who is not a computer science expert, for example, by developing a VAMP plugin. Motivated by the observation in Figure 4.2 that the second-order scattering transform carries information on the modulation rate, we can use the scattering transform as a tool for playing technique modelling. Figure 7.1 shows an example of modelling the modulation rate of a trill played on G6-A6. A clear harmonic partial appears between 5 and 8 Hz, which indicates the range of the modulation rate. The modulation extent can be implicitly estimated by the decomposition of the expanded frequency bands around the decomposition trajectory (see Section 4.2).

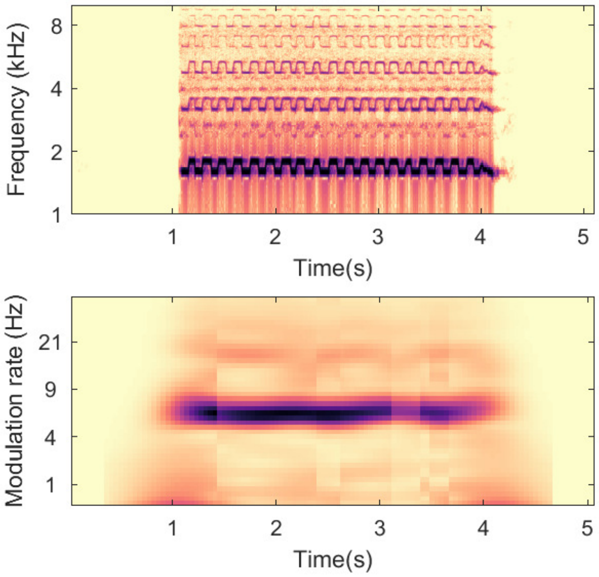


Figure 7.1: Example of a trill modelling played on G6-A6 using adaptive scattering. Top: log spectrogram; bottom: adaptive time scattering feature before log-normalisation.

Playing technique recognition and modelling can greatly help music synthesis systems generate realistic sounds that account for acoustic variations due to the exercise of a variety of instrumental or vocal techniques.

A music note ornamentor is also possible since playing techniques carry important information regarding musical styles. Remodelling a straight note based on a playing technique or articulation of a professional performer or synthesising playing techniques that go beyond real instrument limitations present other attractive directions for further exploration, for example, creating a flutter-tongue effect for piano.

We conclude that the scattering transform offers a versatile and explainable representation for analysing playing techniques in real-world music performances, and opens up new avenues for audio signal analysis.

Bibliography

Jakob Abeßer and Gerald Schuller. Instrument-centered music transcription of solo bass guitar recordings. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 25(9):1741–1750, 2017.

Jakob Abeßer, Hanna Lukashevich, and Gerald Schuller. Feature-based extraction of plucking and expression styles of the electric bass guitar. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2290–2293, 2010.

Felix Albu and Dominique Martinez. The application of support vector machines with Gaussian kernels for overcoming co-channel interference. In *Proceedings of the IEEE Signal Processing Society Workshop*, pages 49–57, 1999.

Islah Ali-MacLachlan. *Computational analysis of style in Irish traditional flute playing*. PhD thesis, Birmingham City University, 2019.

Joakim Andén and Stéphane Mallat. Scattering representation of modulated sounds. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2012.

Joakim Andén and Stéphane Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.

Joakim Andén, Vincent Lostanlen, and Stéphane Mallat. Joint time–frequency scattering. *IEEE Transactions on Signal Processing*, 67(14): 3704–3718, 2019.

Lydia Ayers. Synthesizing trills for the Chinese dizi. In *International Computer Music Conference (ICMC)*, pages 227–30. Singapore, 2003.

Lydia Ayers. Synthesizing timbre tremolos and flutter tonguing on wind instruments. In *International Computer Music Conference (ICMC)*. Miami, Florida, USA, 2004.

Lydia Ayers. Synthesising Chinese flutes using Csound. *Organised Sound*, 10(1):37–49, 2005.

Carlo Baugé, Mathieu Lagrange, Joakim Andén, and Stéphane Mallat. Representing environmental sounds using the separable scattering transform. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8667–8671, 2013.

Emmanouil Benetos. *Automatic Transcription of Polyphonic Music Exploiting Temporal Evolution*. PhD thesis, Queen Mary University of London, UK, 2012.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

Sebastian Böck and Gerhard Widmer. Maximum filter vibrato suppression for onset detection. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, volume 7, 2013.

Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, José R Zapata, and Xavier Serra. Essentia: an open-source library for sound and music analysis. In *Proceedings of the 21st ACM International Conference on Multimedia*, page 855–858, 2013.

Elodie F Briefer. Vocal expression of emotions in mammals: mechanisms of production and evidence. *Journal of Zoology*, 288(1):1–20, 2012.

Judith C Brown and Paris Smaragdis. Independent component analysis for automatic note extraction from musical trills. *The Journal of the Acoustical Society of America (JASA)*, 115(5):2295–2306, 2004.

Carlos E Cancino-Chacón, Maarten Grachten, Werner Goebl, and Gerhard Widmer. Computational models of expressive music performance: A comprehensive and critical review. *Frontiers in Digital Humanities*, 5(25):1–23, 2018.

Carlos Eduardo Cancino-Chacón, Thassilo Gadermaier, Gerhard Widmer, and Maarten Grachten. An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music. *Machine Learning*, 106(6):887–909, 2017.

Chris Cannam, Christian Landone, and Mark Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1467–1468, 2010.

Estefania Cano, Derry FitzGerald, Antoine Liutkus, Mark D Plumbley, and Fabian-Robert Stöter. Musical source separation: An introduction. *IEEE Signal Processing Magazine*, 36(1):31–40, 2018.

Jane Charles. *Playing Technique and Violin Timbre: Detecting Bad Playing*. PhD thesis, Dublin Institute of Technology, Ireland, 2010.

Yuan-Ping Chen, Li Su, and Yi-Hsuan Yang. Electric guitar playing technique detection in real-world recording based on F0 sequence pattern recognition. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 708–714, 2015.

Elaine Chew. *Towards a Mathematical Model of Tonality*. PhD thesis, Massachusetts Institute of Technology, USA, 2000.

Nicholas Collias and Martin Joos. The spectrographic analysis of sound signals of the domestic fowl. *Behaviour*, 5(1):175–188, 1953.

Nicholas Cook. Between art and science: Music as performance. *Journal of the British Academy*, 2:1–25, 2014.

Fergal Cotter and Nick Kingsbury. A learnable scatternet: Locally invariant convolutional layers. In *IEEE International Conference on Image Processing*, pages 350–354, 2019.

Kaixuan Cuan, Tiemin Zhang, Junduan Huang, Cheng Fang, and Yun Guan. Detection of avian influenza-infected chickens based on a chicken sound convolutional neural network. *Computers and Electronics in Agriculture*, 178:105688, 2020.

Shuqi Dai, Zheng Zhang, and Gus G. Xia. Music style transfer: A position paper. *arXiv preprint arXiv:1803.06841*, 2018.

Johanna Devaney, Michael I Mandel, and Ichiro Fujinaga. A study of intonation in three-part singing using the automatic music performance analysis and comparison toolkit (AMPACT). In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 511–516, 2012.

Xiaodong Du, Lenn Carpentier, Guanghui Teng, Mulin Liu, Chaoyuan Wang, and Tomas Norton. Assessment of laying hens' thermal comfort using sound technology. *Sensors*, 20(2):473, 2020.

Jean-Francois Ducher and Philippe Esling. Folded CQT RCNN for real-time recognition of instrument playing techniques. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

Alf Gabrielsson. The performance of music. In *The Psychology of Music*, pages 501–602. Academic Press, 1999.

Mikel Gainza and Eugene Coyle. Automating ornamentation transcription. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages I–69, 2007.

Mikel Gainza, Eugene Coyle, and Robert Lawlor. Single note ornaments transcription for the Irish tin whistle based on onset detection. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2004.

Dimitrios Giannoulis and Anssi Klapuri. Musical instrument recognition in polyphonic audio using missing feature approach. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 21 (9):1805–1817, 2013.

Olivier Gillet and Gaël Richard. ENST-Drums: an extensive audio-visual database for drum signals processing. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 156–159, 2006.

Sergio Giraldo and Rafael Ramírez. Performance to score sequence matching for automatic ornament detection in jazz music. In *International Conference of New Music Concepts (ICMNC)*, 2015.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT Press, 2016.

Glenn Eric Hall, Hassan Ezzaidi, Mohammed Bahoura, and Christophe Volat. Classification of pizzicato and sustained articulations. In *European Signal Processing Conference (EUSIPCO)*, pages 1–5, 2013.

Yoonchang Han and Kyogu Lee. Hierarchical approach to detect common mistakes of beginner flute players. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 77–82, 2014.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Science & Business Media, 2009.

Perfecto Herrera, Alexandre Yeterian, and Fabien Gouyon. Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques. In *International Conference on Music and Artificial Intelligence (ICMAI)*, pages 69–80, 2002.

Yu-Fen Huang, Jeng-I Liang, I-Chieh Wei, and Li Su. Joint analysis of mode and playing technique in guqin performance with machine learning. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

Thomas Hummel. Algorithmic orchestration with contimbre. *Journées d'Informatique Musicales-JIM2014*, pages 139–140, 2014.

Nikša Jakovljević, Nina Maljković, Dragiša Mišković, Petar Knežević, and Vlado Delić. A broiler stress detection system based on audio signal processing. In *IEEE Telecommunications Forum (TELFOR)*, pages 1–4, 2019.

Peter Jančovič, Münevver Köküer, and Wrena Baptiste. Automatic transcription of ornamented Irish traditional flute music using hidden Markov models. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 756–762, 2015.

Jae Won Noella Jung. *Jascha Heifetz, David Oistrakh, Joseph Szigeti: Their Contributions to the Violin Repertoire of the Twentieth Century.* PhD thesis, Florida State University, USA, 2007.

Patrik N Juslin and Petri Laukka. Communication of emotions in vocal expression and music performance: different channels, same code? *Psychological Bulletin*, 129(5):770, 2003.

Christian Kehling, Jakob Abeßer, Christian Dittmar, and Gerald Schuller. Automatic tablature transcription of electric guitar recordings by estimation of score- and instrument-related parameters. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 219–226, 2014.

Anssi P Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on speech and audio processing*, 11(6):804–816, 2003.

Nadine Kroher and Emilia Gómez. Automatic singer identification for improvisational styles based on vibrato, timbre and statistical performance descriptors. In *International Computer Music Conference (ICMC)*, 2014.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

Alexander Lerch, Claire Arthur, Ashis Pati, and Siddharth Gururani. An interdisciplinary review of music performance analysis. *Transactions of the International Society for Music Information Retrieval*, 3(1), 2020.

Alexandre Lerch, Pierre Roy, François Pachet, and Laurent Nagle. Closed-loop bird-computer interactions: A new method to study the role of bird callings. *Animal Cognition*, 14:203–211, 2011.

Pei Ching Li, Li Su, Yi Hsuan Yang, and Alvin W.Y. Su. Analysis of expressive musical terms in violin using score-informed and expression-based audio features. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 809–815, 2015.

Xiaocang Li. *History of Chinese Opera*. Social Sciences Academic Press, 2016.

Beici Liang, György Fazekas, Andrew P. McPherson, and Mark Sandler. Piano pedaller: a measurement system for classification and visualisation of piano pedalling techniques. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 325–329, 2017.

Beici Liang, György Fazekas, and Mark Sandler. Piano sustain-pedal detection using convolutional neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–245, 2019.

Cynthia Liem, Alan Hanjalic, and Craig Sapp. Expressivity in musical timing in relation to musical structure and interpretation: a cross-performance, audio-based approach. In *Proceedings of the AES 42nd International Conference*, 2011.

Longshen Liu, Bo Li, Ruqian Zhao, Wen Yao, Mingxia Shen, and Ji Yang. A novel method for broiler abnormal sound detection using WMFCC and HMM. *Journal of Sensors*, 2020, 2020.

Vincent Lostanlen and Stéphane Mallat. Wavelet scattering on the pitch spiral. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2015.

Vincent Lostanlen, Joakim Andén, and Mathieu Lagrange. Extended playing techniques: the next milestone in musical instrument recognition. In *5th International Conference on Digital Libraries for Musicology (DLfM)*, 2018.

Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, third edition, 2008.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203, 2016.

G. Marx, J. Leppelt, and F. Ellendorff. Vocalisation in chicks (*Gallus gallus* dom.) during stepwise social isolation. *Applied Animal Behaviour Science*, 75(1):61–74, 2001.

Paul Masri. *Computer Modelling of Sound for Transformation and Synthesis of Musical Signals*. PhD thesis, University of Bristol, UK, 1996.

Matthias Mauch and Simon Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, 2014.

Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the Python in Science Conference*, volume 8, pages 18–25, 2015.

Duncan W. H. Menzies and Andrew P. McPherson. Highland piping ornament recognition using dynamic time warping. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 50–53, 2015.

Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. Metrics for polyphonic sound event detection. *Applied Sciences*, 6(6):162, 2016.

Miguel Molina-Solana, Josep Lluís Arcos, and Emilia Gómez. Identifying violin performers by their expressive trends. *Intelligent Data Analysis*, 14(5):555–571, 2010.

Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer, 2015.

Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

Andreas Neocleous, George Azzopardi, Christos N. Schizas, and Nicolai Petkov. Filter-based approach for ornamentation detection and recognition in singing folk music. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 558–569, 2015.

Tin Lay Nwe and Haizhou Li. Exploring vibrato-motivated acoustic features for singer identification. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 15(2):519–530, 2007.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *arXiv preprint, arXiv:1609.03499*, 2016.

Tan Hakan Ozaslan and Josep Lluis Arcos. Legato and glissando identification in classical guitar. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 457–463, 2010.

Tan Hakan Özaslan, Xavier Serra, and Josep Lluis Arcos. Characterization of embellishments in ney performances of makam music in Turkey. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 13–18, 2012.

Caroline Palmer and Sean Hutchins. What is musical prosody? *Psychology of Learning and Motivation*, 46:245–278, 2006.

Hee-Suk Pang and Doe-Hyun Yoon. Automatic detection of vibrato in monophonic music. *Pattern Recognition*, 38(7):1135–1138, 2005.

Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM, Paris, France, 2004.

Eric Prame. Measurements of the vibrato rate of ten singers. *The journal of the Acoustical Society of America*, 96(4):1979–1984, 1994.

Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2): 257–286, 1989.

Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. mir_eval: A transparent implementation of common mir metrics. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

António Ramires and Xavier Serra. Data augmentation for instrument classification robust to audio effects. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2019.

Rafael Ramirez, Esteban Maestre, Alfonso Perez, and Xavier Serra. Automatic performer identification in Celtic violin audio recordings. *Journal of New Music Research*, 40(2):165–174, 2011.

Loïc Reboursière, Otso Lähdeoja, Thomas Drugman, Stéphane Dupont, Cécile Picard-Limpens, and Nicolas Riche. Left and right-hand guitar playing techniques detection. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2012.

Ricardo Malheiro Renato Panda and Rui Pedro Paiva. Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, (1):1–1, 2018.

Justin Salamon and Emilia Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 20(6): 1759–1770, 2012.

Alvaro Sarasúa, Baptiste Caramiaux, Atau Tanaka, and Miguel Ortiz. Datasets for the analysis of expressive musical gestures. In *Proceedings of the 4th International Conference on Movement Computing*, pages 1–4, 2017.

Christian Schörkhuber and Anssi Klapuri. Constant-Q transform toolbox for music processing. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 3–64, 2010.

Chi-Ching Shih, Pei-Ching Li, Yi-Ju Lin, Yu-Lin Wang, Alvin W. Y. Su, Li Su, and Yi-Hsuan Yang. Analysis and synthesis of the violin playing styles of Heifetz and Oistrakh. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2017.

Carl Southall, Chih-Wei Wu, Alexander Lerch, and Jason Hockman. MDB Drums: An annotated subset of MedleyDB for automatic drum transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

Ajay Srinivasamurthy, Andre Holzapfel, Kaustuv Kanti Ganguli, and Xavier Serra. Aspects of tempo and rhythmic elaboration in Hindustani music: A corpus study. *Frontiers in Digital Humanities*, 4:20, 2017.

Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America (JASA)*, 8(3):185–190, 1937.

Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. Open-Unmix - A reference implementation for music source separation. *Journal of Open Source Software*, 4(41):1667, 2019.

Li Su, Hsin-Ming Lin, and Yi-Hsuan Yang. Sparse modeling of magnitude and phase-derived spectra for playing technique classification. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(12):2122–2132, 2014a.

Li Su, Li-Fan Yu, and Yi-Hsuan Yang. Sparse cepstral and phase codes for guitar playing technique classification. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 9–14, 2014b.

Etienne Thoret, Philippe Depalle, and Stephen McAdams. Perceptually salient regions of the modulation power spectrum for musical instrument identification. *Frontiers in psychology*, 8(587), 2017.

Chen-Gia Tsai. *The Chinese Membrane Flute (Dizi): Physics and Perception of Its Tones.* PhD thesis, Humboldt University of Berlin, Germany, 2003.

Martin Vetterli and Jelena Kovacevic. *Wavelets and Subband Coding.* Prentice-hall, 1995.

Changhong Wang, Emmanouil Benetos, Vincent Lostanlen, and Elaine Chew. Adaptive time–frequency scattering for periodic modulation recognition in music signals. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 809–815, November 2019a.

Changhong Wang, Emmanouil Benetos, Xiaojie Meng, and Elaine Chew. HMM-based glissando detection for recordings of Chinese bamboo flute. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 545–550, May 2019b.

Changhong Wang, Vincent Lostanlen, Emmanouil Benetos, and Elaine Chew. Playing technique recognition by joint time–frequency scattering. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 881–885, 2020a.

Changhong Wang, Emmanouil Benetos, Vincent Lostanlen, and Elaine Chew. Adaptive scattering transforms for playing technique recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, submitted.

Changhong Wang, Emmanouil Benetos, Shuge Wang, and Elisabetta Versace. Joint scattering for automatic chick call recognition. *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, submitted.

Zehao Wang, Jingru Li, Xiaoou Chen, Zijin Li, Shicheng Zhang, Baoqiang Han, and Deshun Yang. Deep learning vs. traditional MIR: a case study on musical instrument playing technique detection. In *13th International Workshop on Machine Learning and Music at ECML/PKDD*, pages 5–9, 2020b.

Philip A Warrick, Vincent Lostanlen, and Masun Nabhan Homsi. Hybrid scattering-LSTM networks for automated detection of sleep arousals. *Physiological Measurement*, 40(7):074001, 2019.

Julia Wilkins, Prem Seetharaman, Alison Wahl, and Bryan Pardo. VocalSet: A singing voice dataset. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 468–474, 2018.

Chih-Wei Wu and Alexander Lerch. On drum playing technique detection in polyphonic mixtures. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 218–224, 2016.

Zhengzheng Xing, Jian Pei, and Eamonn Keogh. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, 12 (1):40–48, 2010.

Luwei Yang. *Computational Modelling and Analysis of Vibrato and Portamento in Expressive Music Performance*. PhD thesis, Queen Mary University of London, UK, 2017.

James E. Youngberg and Steven F. Boll. Constant-Q signal analysis and synthesis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 375–378, 1978.

Yongming Zhan. *Fourteen Basisc Lectures of Dizi.* People's Music Publishing House, 2009.

Weiliang Zhang. *Zhudi Art Research.* People's Music Publishing House, 2011.

Songting Zhao. *Ten Tutorials of Dizi Playing Techniques.* Culture and Art Publishing House, 2001.

Yudong Zhao, Changhong Wang, György Fazekas, Emmanouil Benetos, and Mark Sandler. Violinist identification based on vibrato features. In *29th European Signal Processing Conference (EUSIPCO)*, 2021.