

Automatic Transcription of Bass Guitar Tracks applied for Music Genre Classification and Sound Synthesis

Dissertation

**zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)**

**vorlelegt der Fakultät für Elektrotechnik und Informationstechnik
der Technischen Universität Ilmenau**

von Dipl.-Ing. Jakob Abeßer geboren am 3. Mai 1983 in Jena

Gutachter: Prof. Dr.-Ing. Gerald Schuller

Prof. Dr. Meinard Müller

Dr. Tech. Anssi Klapuri

Tag der Einreichung: 05.12.2013

Tag der wissenschaftlichen Aussprache: 18.09.2014

Acknowledgments

I am grateful to many people who supported me in the last five years during the preparation of this thesis. First of all, I would like to thank Prof. Dr.-Ing. Gerald Schuller for being my supervisor and for the inspiring discussions that improved my understanding of good scientific practice.

My gratitude also goes to Prof. Dr. Meinard Müller and Dr. Anssi Klapuri for being available as reviewers. Thank you for the valuable comments that helped me to improve my thesis.

I would like to thank my former and current colleagues and fellow PhD students at the Semantic Music Technologies Group at the Fraunhofer IDMT for the very pleasant and motivating working atmosphere. Thank you Christian, Holger, Sascha, Hanna, Patrick, Christof, Daniel, Anna, and especially Alex and Estefanía for all the tea-time conversations, discussions, last-minute proof readings, and assistance of any kind. Thank you Paul for providing your musicological expertise and perspective in the genre classification experiments.

I also thank Prof. Petri Toiviainen, Dr. Olivier Lartillot, and all the colleagues at the Finnish Centre of Excellence in Interdisciplinary Music Research at the University of Jyväskylä for a very inspiring research stay in 2010. Many experiments described in this thesis would not have been possible without the support of other researchers in the Music Information Retrieval community. I want to thank Justin Salamon, Matti Ryyränen, and Anssi Klapuri for providing me with their algorithms' transcription results and Cory McKay for publishing the jSymbolic software.

Finally, my biggest thanks go to my family, thank you Harald, Monika, Michel, Wolf for your support and thank you Franziska for your love and patience.

Where words fail, music speaks.
(Hans Christian Andersen)

Abstract

Music recordings most often consist of multiple instrument signals, which overlap in time and frequency. In the field of Music Information Retrieval (MIR), existing algorithms for the automatic transcription and analysis of music recordings aim to extract semantic information from mixed audio signals. In the last years, it was frequently observed that the algorithm performance is limited due to the signal interference and the resulting loss of information. One common approach to solve this problem is to first apply source separation algorithms to isolate the present musical instrument signals before analyzing them individually. The performance of source separation algorithms strongly depends on the number of instruments as well as on the amount of spectral overlap.

In this thesis, isolated instrumental tracks are analyzed in order to circumvent the challenges of source separation. Instead, the focus is on the development of instrument-centered signal processing algorithms for music transcription, musical analysis, as well as sound synthesis. The electric bass guitar is chosen as an example instrument. Its sound production principles are closely investigated and considered in the algorithmic design.

In the first part of this thesis, an automatic music transcription algorithm for electric bass guitar recordings will be presented. The audio signal is interpreted as a sequence of sound events, which are described by various parameters. In addition to the conventionally used score-level parameters note onset, duration, loudness, and pitch, instrument-specific parameters such as the applied instrument playing techniques and the geometric position on the instrument fretboard will be extracted. Different evaluation experiments confirmed that the proposed transcription algorithm outperformed three state-of-the-art bass transcription algorithms for the transcription of realistic bass guitar recordings. The estimation of the instrument-level parameters works with high accuracy, in particular for isolated note samples.

In the second part of the thesis, it will be investigated, whether the sole analysis of the bassline of a music piece allows to automatically classify its music genre. Different score-based audio features will be proposed that allow to quantify tonal, rhythmic, and structural properties of basslines. Based on a novel data set of 520 bassline transcriptions from 13 different music genres, three approaches for music genre classification were compared. A rule-based classification system could achieve a mean class accuracy of 64.8 % by only taking features into account that were extracted from the bassline of a music piece.

The re-synthesis of a bass guitar recordings using the previously extracted note parameters will be studied in the third part of this thesis. Based on the physical modeling of string instruments, a novel sound synthesis algorithm tailored to the electric bass guitar will be presented. The algorithm mimics different aspects of the instrument's sound production mechanism such as string excitement, string damping, string-fret collision, and the influence of the electro-magnetic pickup. Furthermore, a parametric audio coding approach will be discussed that allows to encode and transmit bass guitar tracks with a significantly smaller bit rate than conventional audio

coding algorithms do. The results of different listening tests confirmed that a higher perceptual quality can be achieved if the original bass guitar recordings are encoded and re-synthesized using the proposed parametric audio codec instead of being encoded using conventional audio codecs at very low bit rate settings.

Zusammenfassung

Musiksignale bestehen in der Regel aus einer Überlagerung mehrerer Einzelinstrumente. Die meisten existierenden Algorithmen zur automatischen Transkription und Analyse von Musikaufnahmen im Forschungsfeld des Music Information Retrieval (MIR) versuchen, semantische Information direkt aus diesen gemischten Signalen zu extrahieren. In den letzten Jahren wurde häufig beobachtet, dass die Leistungsfähigkeit dieser Algorithmen durch die Signalüberlagerungen und den daraus resultierenden Informationsverlust generell limitiert ist. Ein möglicher Lösungsansatz besteht darin, mittels Verfahren der Quellentrennung die beteiligten Instrumente vor der Analyse klanglich zu isolieren. Die Leistungsfähigkeit dieser Algorithmen ist zum aktuellen Stand der Technik jedoch nicht immer ausreichend, um eine sehr gute Trennung der Einzelquellen zu ermöglichen. In dieser Arbeit werden daher ausschließlich isolierte Instrumentalaufnahmen untersucht, die klanglich nicht von anderen Instrumenten überlagert sind. Exemplarisch werden anhand der elektrischen Bassgitarre auf die Kangerzeugung dieses Instrumentes hin spezialisierte Analyse- und Klangsynthesealgorithmen entwickelt und evaluiert.

Im ersten Teil der vorliegenden Arbeit wird ein Algorithmus vorgestellt, der eine automatische Transkription von Bassgitarrenaufnahmen durchführt. Dabei wird das Audiosignal durch verschiedene Klangereignisse beschrieben, welche den gespielten Noten auf dem Instrument entsprechen. Neben den üblichen Notenparametern Anfang, Dauer, Lautstärke und Tonhöhe werden dabei auch instrumentenspezifische Parameter wie die verwendeten Spieltechniken sowie die Saiten- und Bundlage auf dem Instrument automatisch extrahiert. Evaluationsexperimente anhand zweier neu erstellter Audiodatensätze belegen, dass der vorgestellte Transkriptionsalgorithmus auf einem Datensatz von realistischen Bassgitarrenaufnahmen eine höhere Erkennungsgenauigkeit erreichen kann als drei existierende Algorithmen aus dem Stand der Technik. Die Schätzung der instrumentenspezifischen Parameter kann insbesondere für isolierte Einzelnoten mit einer hohen Güte durchgeführt werden.

Im zweiten Teil der Arbeit wird untersucht, wie aus einer Notendarstellung typischer sich wiederholender Basslinien auf das Musikgenre geschlossen werden kann. Dabei werden Audiomerkmale extrahiert, welche verschiedene tonale, rhythmische, und strukturelle Eigenschaften von Basslinien quantitativ beschreiben. Mit Hilfe eines neu erstellten Datensatzes von 520 typischen Basslinien aus 13 verschiedenen Musikgenres wurden drei verschiedene Ansätze für die automatische Genreklassifikation verglichen. Dabei zeigte sich, dass mit Hilfe eines regelbasierten Klassifikationsverfahrens nur Anhand der Analyse der Basslinie eines Musikstückes bereits eine mittlere Erkennungsrate von 64,8 % erreicht werden konnte.

Die Re-synthese der originalen Bassspuren basierend auf den extrahierten Notenparametern wird im dritten Teil der Arbeit untersucht. Dabei wird ein neuer Audiosynthesealgorithmus vorgestellt, der basierend auf dem Prinzip des Physical Modeling verschiedene Aspekte der für die Bassgitarre charakteristische Kangerzeugung wie Saitenanregung, Dämpfung, Kollision zwischen Saite und Bund sowie dem Tonabnehmerverhalten nachbildet. Weiterhin wird ein parametrischer

Audiokodierungsansatz diskutiert, der es erlaubt, Bassgitarrenspuren nur anhand der ermittelten notenweisen Parameter zu übertragen um sie auf Dekoderseite wieder zu resynthetisieren. Die Ergebnisse mehrerer Hötest belegen, dass der vorgeschlagene Synthesealgorithmus eine Re-Synthese von Bassgitarrenaufnahmen mit einer besseren Klangqualität ermöglicht als die Übertragung der Audiodaten mit existierenden Audiokodierungsverfahren, die auf sehr geringe Bitraten ein gestellt sind.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research Objectives	2
1.3	Contributions	4
1.4	Thesis Outline & Summary	4
1.5	Own Publications	6
1.6	Notation	7
1.6.1	Auxiliary Functions	7
1.6.2	Note Parameters & Features	7
1.6.3	Indices & Formula Signs	8
2	Foundations	11
2.1	The Electric Bass Guitar	11
2.1.1	Instrument Construction	11
2.1.2	Playing Techniques	13
2.1.3	Modelling the Sound Production	19
2.2	Score & Tablature Representation of Music	25
2.3	Machine Learning Methods	26
2.3.1	Evaluation using Cross-validation	26
2.3.2	Feature Selection & Feature Space Transformation	27
2.3.3	Classification	28
I	Automatic Transcription of Bass Guitar Tracks	31
3	Related Work	35
3.1	Bass Transcription	35
3.1.1	Common Presumptions	36
3.1.2	Algorithmic Steps	36
3.2	Instrument-level Parametrization of String Instrument Recordings	41
3.2.1	Estimation of Playing Techniques	42
3.2.2	Estimation of the Fretboard Position	46
4	Contribution	51
4.1	Instrument-centered Bass Guitar Transcription Algorithm	52
4.1.1	Development Data Sets	52
4.1.2	Pre-processing & Spectral Estimation	52

4.1.3	Onset Detection	54
4.1.4	Fundamental Frequency Tracking	55
4.1.5	Offset Detection	57
4.1.6	Spectral Envelope Modeling	57
4.1.7	Envelope Segmentation & Loudness Estimation	60
4.1.8	Feature Extraction	61
4.1.9	Estimation of Plucking Style & Expression Style	65
4.1.10	Estimation of String Number & Fret Number	65
4.1.11	Context-based Error Correction	66
4.2	Data Sets	66
4.2.1	IDMT-SMT-BASS	66
4.2.2	IDMT-SMT-BASS-SINGLE-TRACKS	66
5	Evaluation	69
5.1	Estimation of Plucking Styles & Expression Styles from Bass Guitar Notes	69
5.2	Estimation of Expression Styles Subclasses from Bass Guitar Notes	70
5.3	Estimation of String Number from Bass Guitar Notes	72
5.4	Transcription of Bass Guitar Tracks	74
5.5	Estimation of Instrument-level Parameters from Bass Guitar Tracks	77
6	Summary	81
II	Application for Music Genre Classification	83
7	Related Work	87
7.1	Genre Classification using Score-based Audio Features	87
7.1.1	Instrument Track	87
7.1.2	Feature Extraction	88
7.1.3	Classification	89
7.1.4	Evaluation	90
7.1.5	Conclusion	91
8	Contribution	93
8.1	Score-based Audio Features	93
8.1.1	Tonality	93
8.1.2	Rhythm	98
8.1.3	Structure	104
8.2	Data Sets	105
8.2.1	IDMT-SMT-BASS-GENRE-MIDI	105
9	Evaluation	107
9.1	Automatic Music Genre Classification based on Repetitive Bass Patterns	107
10	Summary	117

III Application for Sound Synthesis & Parametric Audio Coding	119
11 Related Work	123
11.1 Digital Music Synthesis	123
11.2 Physical Modeling	124
11.3 Synthesis Model Extensions and Calibration	124
11.4 Parametric Audio Coding Schemes	126
12 Contribution	129
12.1 Physical Modeling Algorithm for Realistic Bass Guitar Synthesis	129
12.1.1 Waveguide Model	129
12.1.2 Tuning of the Model Parameters	132
12.1.3 Inharmonicity	134
12.2 Parametric Model-based Audio Coding of Bass Guitar Tracks	134
12.2.1 Overview	134
12.2.2 Parameter & Bitrate	134
12.2.3 Application for Polyphonic Music	136
13 Evaluation	137
13.1 Perceptual Audio Quality of Synthesized Basslines	137
13.2 Perceptual Improvements by Applying Model Tuning to Specific Instruments . .	139
13.3 Importance of Plucking Styles and Expression Styles on the Perceptual Quality of Synthesized Basslines	141
14 Summary	145
List of Figures	147
List of Tables	149
References	153

1 Introduction

1.1 Motivation

In the field of Music Information Retrieval (MIR), different tasks such as automatic music transcription, music genre classification, music similarity computation, or music recommendation require to automatically extract semantic properties from given music recordings. These properties range from a song's instrumentation to its underlying harmonic and rhythmic structure. Humans require years of musical training in order to successfully perform these tasks by ear. Therefore, the attempt to develop algorithms to automatically solve these tasks for arbitrary music pieces seems fairly ambitious.

In most music recordings, multiple instruments are played simultaneously. In order to extract semantic information, analyzing the individual instrument tracks appears to be more promising than analyzing the mixed audio signal. Since the signals of the sound sources overlap both in time and frequency, the task of source separation is very demanding and subject to various scientific publications. So far, sound separation algorithms can mimic the human auditory ability to isolate particular sound sources only to a limited extent. Ideally, the analysis of individual instruments can be performed using perfectly isolated audio tracks that were extracted from an audio mixture. In this thesis, the error-prone step of source separation will be omitted by solely analyzing bass guitar tracks taken from multi-track recordings. The multi-track recording technique is common practice nowadays in music studios.

The bass guitar was established in modern rock and pop music in the early 1950s. Since the instrument has electro-magnetic pickups, its output signal could be amplified on stage. This was a significant advantage compared to the acoustic double bass, which—at that time—still was the most popular bass instrument. The bass guitar allowed the bass player to cope with the increasing overall loudness of other instruments such as the drums or the electric guitar. Within the last decades, a wide range of playing techniques were adopted from the electric guitar and transferred to the bass guitar. The instrument itself is nowadays widely used in many music genres, both as accompaniment and solo instrument.

This thesis is structured into three parts that deal with the automatic transcription of bass guitar tracks as well its application for music analysis and sound synthesis. First, bass guitar tracks will be transcribed in order to describe each note by a suitable set of parameters. Then, the obtained parameters will be used in two application scenarios: First, a musical analysis algorithm will be presented that allows to automatically classify the musical genre of a piece of music solely based on the repeating bass pattern. Second, a sound synthesis algorithm will be introduced that allows to re-synthesize the original bass guitar track based on the extracted note parameters.

The topic of this thesis is of strong interdisciplinary nature between audio signal processing, music information retrieval, machine learning, and musicology. However, the focus of this thesis is mainly on the first two fields, the other two fields will be discussed only where necessary.

1.2 Research Objectives

Figure 1.1 illustrates the structure of this thesis. The research objectives followed in the three parts music transcription, music analysis, and audio coding & sound synthesis will be detailed in the next sections.

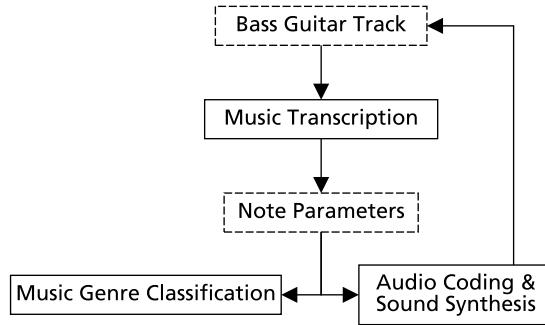


Figure 1.1: Flowchart illustrating the structure of this thesis. After a given bass guitar track is transcribed, the extracted notes parameters are used in two application scenarios. First, the music genre is automatically classified. Second, the note parameters are transmitted to a sound synthesis algorithm that allows to re-synthesize the original bass guitar track.

Music Transcription

Musical instruments such as the bass guitar provide a large vocabulary of expressive gestures or playing techniques, which can be used for creating different sounds. The main research question of the first part of this thesis is: How can the note events in bass guitar recordings be best described by a limited set of parameters? Playing the bass guitar with different playing techniques at different geometric positions on the instrument neck leads to very diverse sound characteristics. Therefore, methods for automatic transcription of bass guitar tracks must be well-adjusted to the instrument's sound production mechanisms.

In the last years, automatic music transcription algorithms seem to have reached an upper performance limit, a so called “glass ceiling” [156]. In this thesis, it will be investigated, whether *instrument-centered music transcription algorithms* allow to outperform general purpose transcription algorithms, which are not tailored to a specific instrument.

The bass guitar transcription algorithm that will be proposed in Chapter 4 first extracts a *piano-roll* representation of the recorded audio signal, where each note event is described by the parameters note onset, offset, and pitch. In contrast to a musical score, the onset and offset times are measured in absolute time and not mapped to the underlying metric structure (beat times) of a musical piece. These *score-level parameters* will be extended by a set of *instrument-level parameters* including the applied playing techniques and the fretboard position on the instrument.

Table 1.1 illustrates, which type of audio data is analyzed in the three parts of this thesis. In the first and third part, isolated bass guitar tracks (audio signals) will be analyzed. The musical analysis explained in the second part will be based on symbolic MIDI files.

Table 1.1: Type of audio data used in the three parts of this thesis. The transcription and sound synthesis discussed in Part I and Part III are based on isolated bass guitar recordings. The genre classification experiments presented in Part II are performed on symbolic MIDI files.

Data type	Part I (Bass Guitar Transcription)	Part II (Genre Classification)	Part III (Sound Synthesis & Audio Coding)
Mixed audio tracks	-	-	-
Isolated audio tracks	x	-	x
Symbolic (MIDI)	-	x	-

Musical Genre Classification

In the second part of the thesis, the score-level parameters will be analyzed to estimate the music genre of a song. The main hypothesis to be investigated is that the bassline itself contains important stylistic information that can be used for genre classification. Following the semantics of MIR, the automatic transcription discussed in Part I can be thought of as a *low-level* analysis, whereas the musical analysis performed in Part II constitutes a *high-level* analysis. Part I focuses on low-level representations, namely the time signal and the spectrogram of the recorded audio signal. In Part II, semantic properties closely related to music theory are extracted from a bassline.

In the first step of the musical analysis, a set of transcription-based audio features are computed. These numerical descriptors quantify different tonal and rhythmic properties of a given bassline. All experiments towards genre classification will be performed using basslines encoded as MIDI files as shown in Table 1.1. Hence, the rhythmic context—given by the time signature and the tempo—and the harmonic context—given by the key and the chords—are considered to be known in advance.¹ In the second step of the musical analysis, three different approaches to classify the music genre based on the extracted features will be compared. For instance, the extracted audio features will be used to train and test statistical classification models for the tasks of music genre classification.

Sound Synthesis & Parametric Audio Coding

In Part III, an instrument-based audio coding scheme will be described, which is tailored towards the bass guitar. The scheme consists of an analysis step and a synthesis step. The analysis step includes the bass guitar transcription algorithm described in Section 1.2. The synthesis step includes a physical modeling algorithm that mimics the sound production of the bass guitar. In comparison to conventional audio coding approaches, the presented system allows to substantially reduce the amount of data to be transmitted. The continuous audio signal is represented as a sequence of note events and each event is characterized by a set of parameters. While reducing the complexity of a piece of music to a fractional part, the extracted parameters also allow to manipulate recordings by changing individual parameters such as the applied playing techniques.

¹The automatic estimation of the tempo, time signature, and key are MIR research tasks of their own and will not be tackled in this thesis.

1.3 Contributions

The main contributions of this thesis can be summarized as follows.

- An instrument-centered transcription algorithm for isolated bass guitar recordings that includes the estimation of the applied playing techniques and fretboard positions.
Section 4.1
- Two novel datasets with bass guitar recordings and annotations that were published as evaluation benchmark for various MIR analysis tasks.
Section 4.2
- Various score-based (high-level) features for characterizing basslines w.r.t. tonality, rhythm, and structure.
Section 8.1
- A novel dataset that consists of 520 basslines from 13 music genres for the task of music genre classification.
Section 8.2
- Comparison of three classification paradigms for music genre classification based on repetitive bass patterns.
Section 9.1
- A novel physical modeling algorithm for electric bass guitar sound synthesis that includes 11 different playing techniques and its application for a parametric audio coding scheme.
Chapter 12
- Investigation of different influence factors on the perceived quality of sound synthesis based on listening tests.
Chapter 13

1.4 Thesis Outline & Summary

This thesis is structured into 3 main parts as described in Section 1.2. The following topics are covered in the individual chapters.

- Chapter 2 provides several foundations for this thesis. First, the construction of the electric bass guitar as well as typical playing techniques are described. Also, models for the sound production of string instruments are discussed in general. Then, a brief comparison between the score and the tablature as the most important written music representations for notating basslines is provided. Finally, the machine learning algorithms for feature selection, feature

space transformation, and classification, which are used throughout this thesis, are briefly introduced.

Part I - Automatic Transcription of Bass Guitar Tracks

- **Chapter 3** reviews state-of-the-art publications on bass transcription as well as on the estimation of playing techniques and fretboard positions from string instrument recordings.
- **Chapter 4** describes the proposed bass guitar transcription algorithm in detail. Furthermore, two novel audio datasets are presented, which were published as evaluation benchmark.
- **Chapter 5** presents five experiments that were performed to evaluate the performance of the proposed transcription algorithm in different scenarios.
- **Chapter 6** summarizes the first part.

Part II - Application for Music Genre Classification

- **Chapter 7** reviews existing publications on automatic music genre classification using score-based audio features.
- **Chapter 8** gives an overview over all score-based (high-level) features, which are used to describe basslines with respect to tonality, rhythm, and structure. Also, a novel dataset of 520 basslines from 13 music genres is introduced, which is used in the genre classification experiment.
- **Chapter 9** compares three different approaches for music genre classification based on repetitive bass patterns. An evaluation experiment is described and the results are discussed.
- **Chapter 10** summarizes the second part.

Part III - Application for Audio Synthesis & Parametric Audio Coding

- **Chapter 11** provides an overview over publications on physical modeling sound synthesis of string instruments. Special focus is put on the synthesis of string instruments as well as their characteristic playing techniques. Furthermore, existing parametric audio coding schemes are briefly reviewed.
 - **Chapter 12** first details the proposed physical modeling algorithm for bass guitar synthesis. Second, a parametric audio coding scheme is discussed, where the proposed bass guitar transcription algorithm, which was presented in the first part, is combined with the synthesis algorithm.
 - **Chapter 13** summarizes the results of three listening tests that were performed in order to evaluate the perceptual quality of the proposed synthesis algorithm. In addition, different influence factors on the perceived audio quality were investigated in detail.
 - **Chapter 14** summarizes the third part.
-

1.5 Own Publications

In this section, the main own publications related to each of the three parts of this thesis are listed. Additional publications are given in the main bibliography.

Part I - Automatic Transcription of Bass Guitar Tracks

- 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), pp. 2290-2293, Dallas, USA, 2010.
- Jakob Abeßer, Christian Dittmar, and Gerald Schuller, “Automatic Recognition and Parametrization of Frequency Modulation Techniques in Bass Guitar Recordings”, in Proceedings of the 42nd Audio Engineering Society (AES) International Conference on Semantic Audio, pp. 1-8, Ilmenau, Germany, 2011.
- Christian Dittmar, Estefanía Cano, Jakob Abeßer, and Sascha Grollmisch, “Music Information Retrieval Meets Music Education”, in Multimodal Music Processing. Dagstuhl Follow-Ups, Meinard Müller, Masataka Goto, and Markus Schedl, Eds., vol. 3, pp. 95-120. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2012.
- Jakob Abeßer, “Automatic String Detection for Bass Guitar and Electric Guitar”, in From Sounds to Music and Emotions - 9th International Symposium, CMMR 2012, London, UK, June 19-22, 2012, Revised Selected Papers, Mitsuko Aramaki, Mathieu Barthet, Richard Kronland-Martinet, and Sólví Ystad, Eds., vol. 7900, pp. 333-352, London, UK, 2013. Springer.
- Jakob Abeßer and Gerald Schuller, “Instrument-centered Music Transcription of Bass Guitar Tracks”, Proceedings of the AES 53rd Conference on Semantic Audio, pp. 166-175, London, UK, 2014.

Part II - Application for Music Genre Classification

- Jakob Abeßer, Hanna Lukashevich, Christian Dittmar, and Gerald Schuller, “Genre Classification using Bass-Related High-Level Features and Playing Styles”, in Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR), pp. 453-458, Kobe, Japan, 2009.
- Jakob Abeßer, Paul Bräuer, Hanna Lukashevich, and Gerald Schuller, “Bass Playing Style Detection Based on High-level Features and Pattern Similarity”, in Proceedings of the 11th International Society of Music Information Retrieval (ISMIR), pp. 93-98, Utrecht, Netherlands, 2010.
- Jakob Abeßer, Hanna Lukashevich, and Paul Bräuer, “Classification of Music Genres based on Repetitive Basslines”, Journal of New Music Research, vol. 41, no. 3, pp. 239-257, 2012.

Part III - Application for Audio Synthesis & Parametric Audio Coding

- Patrick Kramer, Jakob Abeßer, Christian Dittmar, and Gerald Schuller, “A Digital Waveguide Model of the Electric Bass Guitar Including Different Playing Techniques”, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 353-356, Kyoto, Japan, 2012.
- Jakob Abeßer, Patrick Kramer, Christian Dittmar, and Gerald Schuller, “Parametric Audio Coding of Bass Guitar Recordings using a Tuned Physical Modeling Algorithm”, in Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13), Maynooth, Ireland, 2013.

1.6 Notation

1.6.1 Auxiliary Functions

Let x be a vector of length $N_x = \dim(x)$ with discrete-valued elements. The set of all unique values of the elements of x is stored in the vector $u^{[x]}$ of length $N_u = \dim(u^{[x]})$. The number of appearances of each unique value in x is stored in the *histogram count* vector $n^{[x]} \in \mathbb{N}^{N_u}$:

$$n^{[x]}(i) = \sum_{k=1}^{N_x} \delta_k \text{ with } \delta_k = \begin{cases} 1 & , \text{ if } x(k) \equiv u^{[x]}(i), \\ 0 & , \text{ otherwise.} \end{cases} \quad (1.1)$$

By normalizing the histogram count vector, a *probability density* $p^{[x]} \in \mathbb{R}^{N_u}$ is derived as

$$p^{[x]}(i) = \frac{n^{[x]}(i)}{N_u}. \quad (1.2)$$

In this thesis, N denotes the number of notes in a bassline.

1.6.2 Note Parameters & Features

The bass guitar track is represented by a set of note parameters. On a *score-level*, each note event is represented by its pitch \mathcal{P} , which corresponds to its fundamental frequency as

$$\mathcal{P} = \lfloor 12 \log_2 \left(\frac{f_0}{440} \right) + 69 + \frac{1}{2} \rfloor \quad (1.3)$$

The beginning time of a note event (*onset*) is denoted as \mathcal{O} , its *duration* as \mathcal{D} . Both parameters are conventionally measured in seconds based on a physical time representation. The *loudness* \mathcal{L} is measured in dB.

Additionally, the *instrument-level* parameters *plucking style* \mathcal{S}_P , *expression style* \mathcal{S}_E (compare Section 2.1.2), *string number* \mathcal{N}_S , and *fret number* \mathcal{N}_F (compare Section 4.1.10) will be extracted for each note event.

All features, which will be used for classification tasks in Part I and Part II of this thesis are denoted by χ with a corresponding subscript.

1.6.3 Indices & Formula Signs

Table 1.2 gives an overview over the most important indices used in this thesis. A complete list of the formula signs used in this thesis is given in Table 1.3.

Table 1.2: Indexing symbols used in this thesis.

Variable	Description	Unit
a	Magnitude	- / dB
f	Frequency	Hz
h	Harmonic number ($h = 0$ refers to the fundamental frequency)	-
i	Note / pattern number	-
k	Frequency index	-
m	Mode number ($m = 1$ refers to the fundamental frequency)	-
n	Time frame number	-
t	Time	s

Table 1.3: Symbols used in this thesis.

Formula Sign	Description	Unit
a_h	Harmonic magnitude	- / dB
$a^l(n)$	Aggregated magnitude envelope of l -th note	- / dB
A	Accuracy	(%)
$\alpha_{\text{On}}(n)$	Onset detection function	-
β	Inharmonicity coefficient	-
c	Wave propagation speed	m/s
C	Cost factor (SVM)	m/s
d_{NB}	Spatial distance between the nut and the bridge of the bass guitar	m
d_{FB}	Spatial distance between a fret and the bridge of the bass guitar	m
d_s	String diameter	m
E	Young's Modulus	Pa
$\hat{f}(k, n)$	Instantaneous frequency	Hz
f_{cut}	Low-pass filter cut-off frequency	Hz
f_h	Harmonic frequency	Hz
$f_{\text{Mode}, m}$	Mode frequency	Hz
f_s	Sampling frequency	Hz
f_+	Wave function	m
f_-	Wave function traveling in opposite direction	m
F	F-Measure	(%)
γ	Parameter of RBF kernel (SVM)	-
g	Damping factor	-
$H_F(z)$	Fractional delay filter	-
$H_L(z)$	Loop filter	-
k_h	Frequency index of h -th harmonic	-

Table 1.3: Symbols used in this thesis.

Formula	Sign	Description	Unit
n_{\max}		Number of frequency bins	-
l		(Vibrating) string length	m
L		Likelihood function	-
$M(k, n)$		Magnitude of Short-time Fourier Transform	- / dB
$M_{\text{IF}}(k, n)$		Reassigned magnitude spectrogram	- / dB
$M_{\text{IF, acc}}(k)$		Accumulated magnitude spectrogram	- / dB
$M_{\text{O}}(k, n)$		Kernel matrix used for onset detection	- / dB
n_{\max}		Number of frames	-
n_{On}^l		Onset (time) frame	-
n_{Off}^l		Offset (time) frame	-
n_{Peak}^l		Magnitude peak (time) frame	-
N		Number of notes	-
N_D		Number of feature space dimensions	-
N_{FFT}		FFT length	-
N_H		Number of harmonics	-
N_I		Number of items	-
N_S		Number of samples	-
p		Probability	-
P		Precision	(%)
$\mathcal{P}_{\text{String}}$		String tunings	-
\mathcal{N}_S		String number	-
\mathcal{N}_F		Fret number	-
S_c		Scattering matrix	-
ρ		Mass density	kg/m
ρ_i		Resistors damping factors	-
r_G		Radius of Gyration	m
R		Recall	(%)
\mathcal{S}_E		Expression style class	-
\mathcal{S}_P		Plucking style class	-
S_S		Cross-sectional area of a string	m^2
t		Time	s
T		Tension	N
x		Spatial location	m
$X(k, n)$		Short-time Fourier Transform	-
χ		Feature	-
y		String displacement	m
\dot{y}		String velocity	m/s

2 Foundations

2.1 The Electric Bass Guitar

The bass guitar is a plucked string instrument, which was first developed in the 1930s. Up until the middle of the 20th century, the acoustic double bass was the most commonly used bass instrument in music ensembles in Western Europe and North America. In the 1950s, new musical instruments such as the electric guitar became more popular and led to an increase in loudness on stage. At that time, the double bass could not be amplified accordingly. Hence, the bass guitar became popular since it allowed the bass players to be louder [90]. Electro-magnetic pickups were used to capture the string vibrations and to convert them into electric signals, which could be transmitted to an external bass amplifier. The bass guitar was strongly influenced by the electric guitar in terms of instrument construction, sound production, and playing techniques. Nowadays, the electric bass guitar is the most widely used bass instrument in various music genres.

In Section 2.1.1, the construction of the bass guitar will be detailed. Section 2.1.2 will explain all bass guitar playing techniques that are investigated in this thesis. Finally, Section 2.1.3 will discuss how the instrument sound production can be modeled.

2.1.1 Instrument Construction

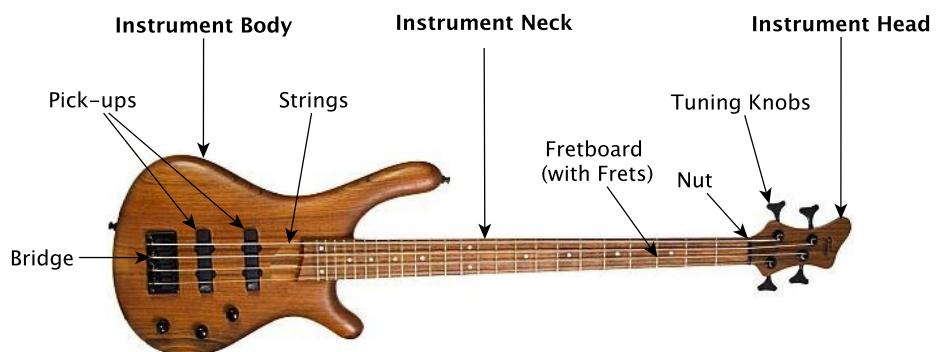


Figure 2.1: Electric bass guitar (model: Fame Baphomet IV) with instrument parts.

French stated in [56] that “[...] from a strictly mechanical point of view, a guitar is a device that connects strings under tension to a resonator with flexible walls and offers a convenient way to shorten the strings to raise their frequencies.” This statement also holds true for the electric bass guitar due to its similar construction. The instrument consists of two parts, a solid wooden

instrument body and an instrument neck, which is connected to the body by screws or glue. Figure 2.1 illustrates the instrument’s construction. Bass guitars commonly have four to six strings, which are attached to the body on the bridge and wired to the tuning knobs on the instrument head.

Fretboard & Frets

Two different types of bass guitars exist—fretted and fretless bass guitars. Similar to guitars, *fretted bass guitars* have metal frets mounted on the instrument fretboard. The geometric distances between the frets follow the logarithmic relationship between the fundamental frequencies of musical notes in the equal temperament tuning as shown in Figure 2.2.

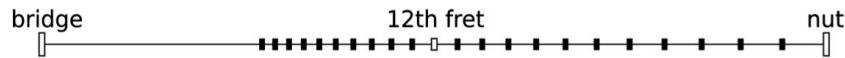


Figure 2.2: Logarithmic spacing of frets on the instrument neck between the bridge and the nut [36]. The 12-th fret is located in the middle of the string.

With d_{NB} being the distance between the nut and the bridge, the distance between the i -th fret and the bridge can be computed as $d_{FB}(i) = 2^{-i/12}d_{NB}$. If the playing hand is moved by one fret closer towards the bridge, the emerging note pitch is raised by one semitone. Consequently, if a note is played at the 12-th fret, the note pitch is one octave higher than the note pitch of the open string. The fixed fret positions allow the musician to play musical notes with the correct fundamental frequency. However, smaller pitch deviations (lower than one semitone) can be achieved by bending the vibrating string vertically using the *vibrato* or *bending* techniques, which will be explained in Section 2.1.2. Bass guitars have between 21 and 24 frets. Hence, the notes that are playable on each string cover a pitch range of up to two octaves.

In contrast to fretted bass guitars, *fretless bass guitars* have no frets on the instrument neck. The absence of frets on the fretboard allows for continuous pitch modulation by horizontal hand movement, i.e., the *slide* technique (see Section 2.1.2 for details).

Tuning & Strings

The most common tuning for four-string bass guitars is from the lowest to the highest string: E₁ ($f_0 = 41.2$ Hz), A₁ (55.0 Hz), D₂ (73.4 Hz), and G₂ (98.0 Hz). Hence, if the bass guitar has 24 frets, its fundamental frequency range is between 41.2 Hz and 382.0 Hz. In this thesis, the focus is on four-string bass guitars since they are most commonly used. Most bass guitar strings have a flexible core that is wrapped with steel wire. The geometry of bass guitar strings clearly deviates from the theoretical model of an infinitely thin string as will be discussed in Section 2.1.3.

Pickups, Audio Effects, and Amplification

In contrast to acoustic instruments such as the acoustic guitar or the double bass, the solid instrument body of the electric bass guitar barely vibrates or radiates acoustic waves. Instead, the string vibration is captured by *electro-magnetic pickups*, which are attached to the instrument

body close to the strings. These pickups transform the kinetic energy of the vibrating string into an electric signal, which is then transmitted by cable from the instrument output jack to an electric *amplification system*. External amplification allows the bass guitar player to adjust the instrument loudness to the overall loudness of the performing band. Most bass guitars include an *on-board equalization system* that allow the musician to adjust the timbre of instrument sound by changing the spectral envelope. Since most bass guitars have two pickups at different relative positions on the instrument, each individual pickup signal has a different acoustic characteristic. The interested reader is referred to [113] for more details on pickup simulations used in sound synthesis models.

Often, audio effect pedals are inserted in the signal chain between the bass guitar and the amplification system. Audio effects such as distortion, chorus, or delay can alter the instrument sound in various ways. The corresponding sound characteristics can be used to automatically detect the applied audio effects in a recording as for instance shown in [161] and [160]. However, the analyzed bass guitar tracks in this thesis are not further processed with audio effects.

Instrument Timbre

The timbre of recorded bass guitar notes and their acoustic properties are affected amongst others by the following factors:

- The *plucking point* denotes the relative position of the plucking position on the string related to the overall string length. If the string is plucked close to the bridge, higher harmonics are stronger. If the string is plucked closer to the fretboard, the instrument sound is often described to be “warmer” since the lower harmonics are more present in the signal. In this thesis, the plucking point will be considered as constant.
- The *fretboard position* denotes the string and the fret number where a note is played on the instrument neck. The bass guitar allows to play certain note pitches at different fretboard positions, which—due to the different string diameters—result in different note timbres. Algorithms for the automatic estimation of the fretboard position from audio recordings will be reviewed in Section 3.2.2.
- The *playing technique* describes how the string vibration is excited by the musician. This affects the initial string displacement function $y(x, 0)$, the initial string velocity function $\dot{y}(x, 0)$ as well as the amount of string damping as will be detailed in Section 2.1.2. In Section 3.2.1, existing methods to automatically estimate string instrument playing techniques from audio recordings will be reviewed.

2.1.2 Playing Techniques

Excerpts from this section were previously presented in [13] and [9]. As pointed out by Cuzzucoli and Lombardo in [37], it is crucial to understand the “relationship between performance technique and sound quality”. The authors emphasize that the finger-string interaction is the most important part in the process of playing a plucked string instrument. In this thesis, the process of playing the bass guitar is modeled by *two consecutive performance gestures* of the musician:

Table 2.1: Proposed taxonomy of bass guitar playing techniques. 5 plucking style classes and 6 expression styles classes are listed as well as their abbreviations. The expression styles *vibrato*, *bending*, and *slide* are furthermore split into two sub-classes.

Plucking Styles		Expression Styles			
Classes		Classes		Sub-classes	
Finger-Style	(FS)	Normal	(NO)		
Picked	(PK)	Harmonics	(HA)		
Muted	(MU)	Dead-notes	(DN)		
Slap-Thumb	(ST)	Vibrato	(VI)	Slow Vibrato	(VIS)
Slap-Pluck	(SP)	Bending	(BE)	Fast Vibrato	(VIF)
				Quarter-tone	(BEQ)
				Bending	
				Half-tone	(BEH)
				Bending	
		Slide	(SL)	Slide-up	(SLU)
				Slide-down	(SLD)

- The first performance gesture describes the initial plucking of a string using the plucking hand, which will be referred to as the *plucking style*. Examples of different plucking styles are shown in Figure 2.3, Figure 2.4, Figure 2.5, and Figure 2.6.
- The second gesture describes the way in which the playing hand is used to manipulate the string vibration, which will be referred to as the *expression style*. Figure 2.7 and Figure 2.8 illustrate different expression styles.

Table 2.1 illustrates a taxonomy of all bass guitar plucking styles and expression styles investigated in this thesis. Five different plucking styles and six expression styles are considered. The three expression styles *bending*, *vibrato*, and *slide*, which involve a modulation of the fundamental frequency of the note, are further subdivided into two sub-classes each. In Section 4.2.1, a novel dataset of recorded bass guitar notes will be described in detail that was used to evaluate the automatic recognition of playing techniques. All shown photographs in this section were made by Patrick Kramer.

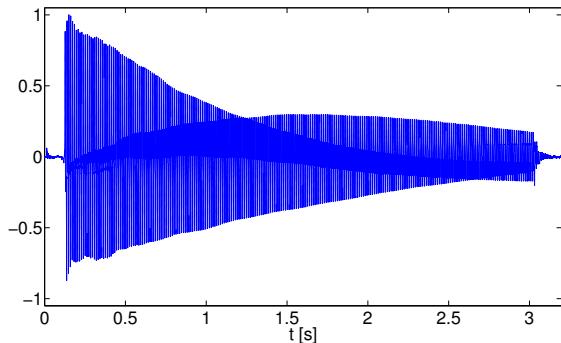
Plucking Styles

The plucking style describes the initial hand gesture of the musician used to excite the string vibration. This gesture mainly affects the acoustic properties of the bass guitar note during the attack part. Jansson [84] has shown that the plucking direction—being either orthogonal or parallel to the instrument body surface—also affects the decay rate of the string during the decay part as shown in [55]. In the following sub-sections, all bass guitar plucking styles discussed in this thesis will be explained in detail.

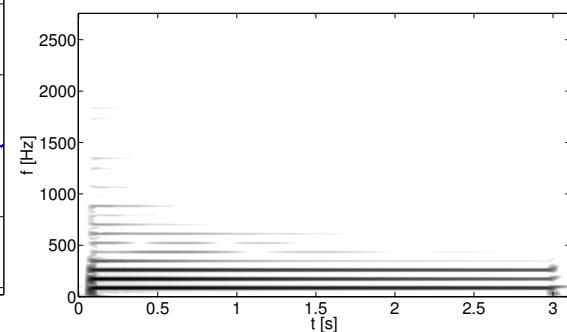
Finger-style The *finger-style* (FS) technique was originally adopted from playing the double bass. The strings are plucked by the index and the middle finger in an altering way as shown in Figure 2.3.



(a) Plucking hand positioning



(b) Time signal



(c) Magnitude spectrogram

Figure 2.3: Plucking hand positioning for the *finger-style* (FS) plucking style. An example note is given as time signal (left figure) and magnitude spectrogram (right figure).

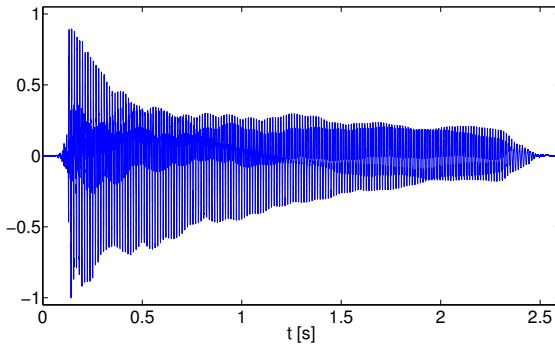
Picked Alternatively, the string can be plucked using a plastic pick instead of using the fingers as illustrated in Figure 2.4. This plucking style is denoted as *picked* (PK) and is frequently used in genres related to rock and heavy metal. The emerging bass guitar sound is brighter compared to the finger-style technique.

Slap The two techniques *slap-thumb* (ST) and *slap-pluck* (SP) are characterized by striking the string using the thumb and picking the string using either the index or the middle finger (see Figure 2.5). Both plucking techniques cause the string to collide with the higher frets on the instrument neck due to the high deflection of the string. The two slap techniques result in a typical metal-like sound.

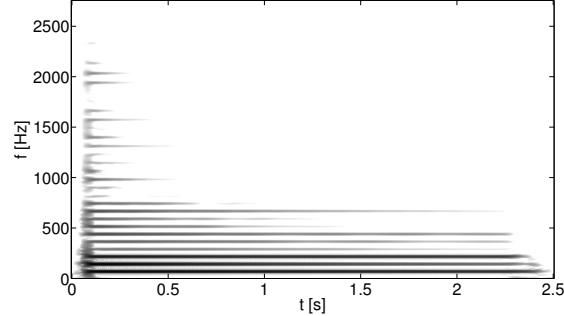
Muted While playing the bass guitar, both hands can be used to *damp* the vibrating string. If the plucking style *muted* (MU) is used, the string is plucked using the thumb of the playing hand and simultaneously damped by using the inner hand side as shown in Figure 2.6. The amount of damping effectively shortens the harmonic decay part. Originally, this technique was used to imitate the typical muffled double bass sound in jazz-related music genres.



(a) Plucking hand positioning



(b) Time signal



(c) Magnitude spectrogram

Figure 2.4: Plucking hand positioning for the *picked* (PK) plucking style. An example note is given as time signal (left figure) and magnitude spectrogram (right figure).

Expression Styles

The expression style is executed by the playing hand and mainly affects the signal properties in the note decay part.

Normal The most common way to play the bass guitar is without any expression style, which will be hereafter referred to as *normal* (NO). Here, one of the fingers of the playing hand is first located at a defined fret-string position that corresponds to the desired note pitch. Then, the string is pushed down on the instrument neck at this fretboard position as shown in Figure 2.3 before the string is plucked.

Dead-notes As mentioned before, the string damping affects the decay time of the harmonic note components and therefore affects the timbre of a bass guitar note. Strong damping can lead to a percussively sounding note with rarely any harmonic components, which will be referred to as the *dead-note* (DN) technique.

Harmonics Similar to the dead-note technique, the playing hand can be used to softly dampen the string vibration at specific geometric positions across the string. If the damping point is at an integer fraction of the string length, a standing wave with a node at the damping point is excited on the string. At the same time, all vibration modes with an anti-node at the damping

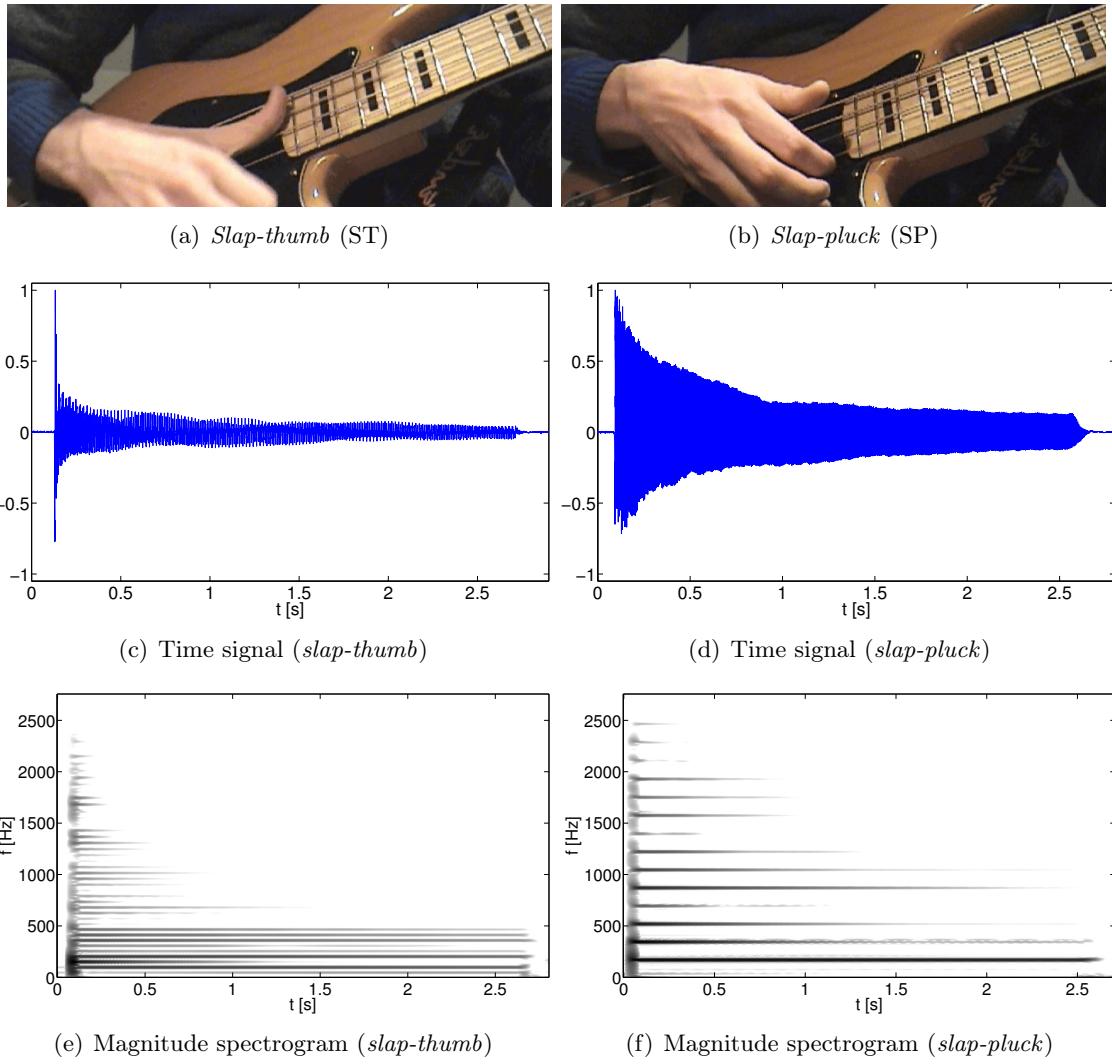


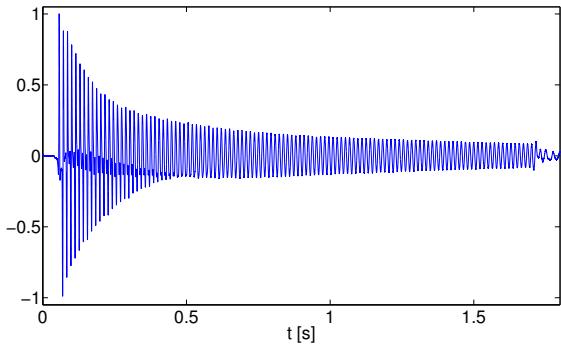
Figure 2.5: Plucking hand positioning for the plucking styles *slap-pluck* (SP) and *slap-thumb* (ST). Two example notes are given as time signals (middle row) and magnitude spectrograms (bottom row).

point are not excited. As a result, the harmonic spectrum differs from the spectrum of the freely vibrating string and perceived fundamental frequency of the played note is often much higher than the regular pitch range of the bass guitar. This technique is referred to as *harmonics* (HA) in this thesis.

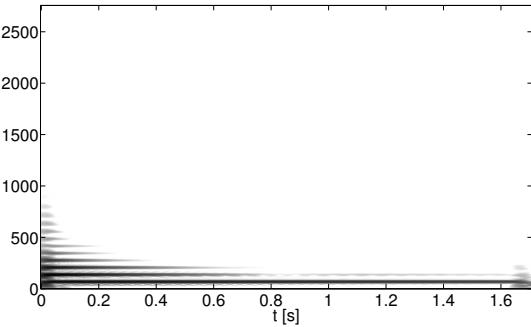
Vibrato & Bending Frequency modulation techniques such as *vibrato* (VI) and *bending* (BE) are very commonly used on string instruments. If the fret-string position is changed, the smallest pitch difference that can be achieved is one semitone. In contrast, modulation techniques allow



(a) Plucking hand positioning



(b) Time signal



(c) Magnitude spectrogram

Figure 2.6: Plucking hand positioning for the *muted* (MU) plucking style. An example note is given as time signal (left figure) and magnitude spectrogram (right figure).

for arbitrary (continuous) pitch alterations. After plucking the string with the plucking hand, the playing hand can be used to bend the string up- and downwards once (bending) or periodically (vibrato). Typical modulation frequencies are up to 2.5 Hz for bending and up to 4 Hz for vibrato [7].

In this thesis, two sub-classes *fast vibrato* (VIF) and *slow vibrato* (VIS) according to the modulation frequency of the vibrato technique as well as *quarter-tone bending* (BEW) and *semi-tone bending* (BES) according to the modulation range of the bending technique are closer investigated. These sub-classes allow for a better parametrization of a given musical performance on the bass guitar. Frequency modulation techniques go along with amplitude modulations of the partial envelopes. This effect was shown to be important for the perception of frequency modulations [85].

Slide Instead of playing two consecutive notes with two plucking gestures, the musician often plays the first note and *slides* (SL) upwards or downwards to the next note without a second note pluck. Again, as for the vibrato and bending technique, two sub-classes were defined depending on the direction of the slide, *slide up* (SLU) and *slide-down* (SLD).

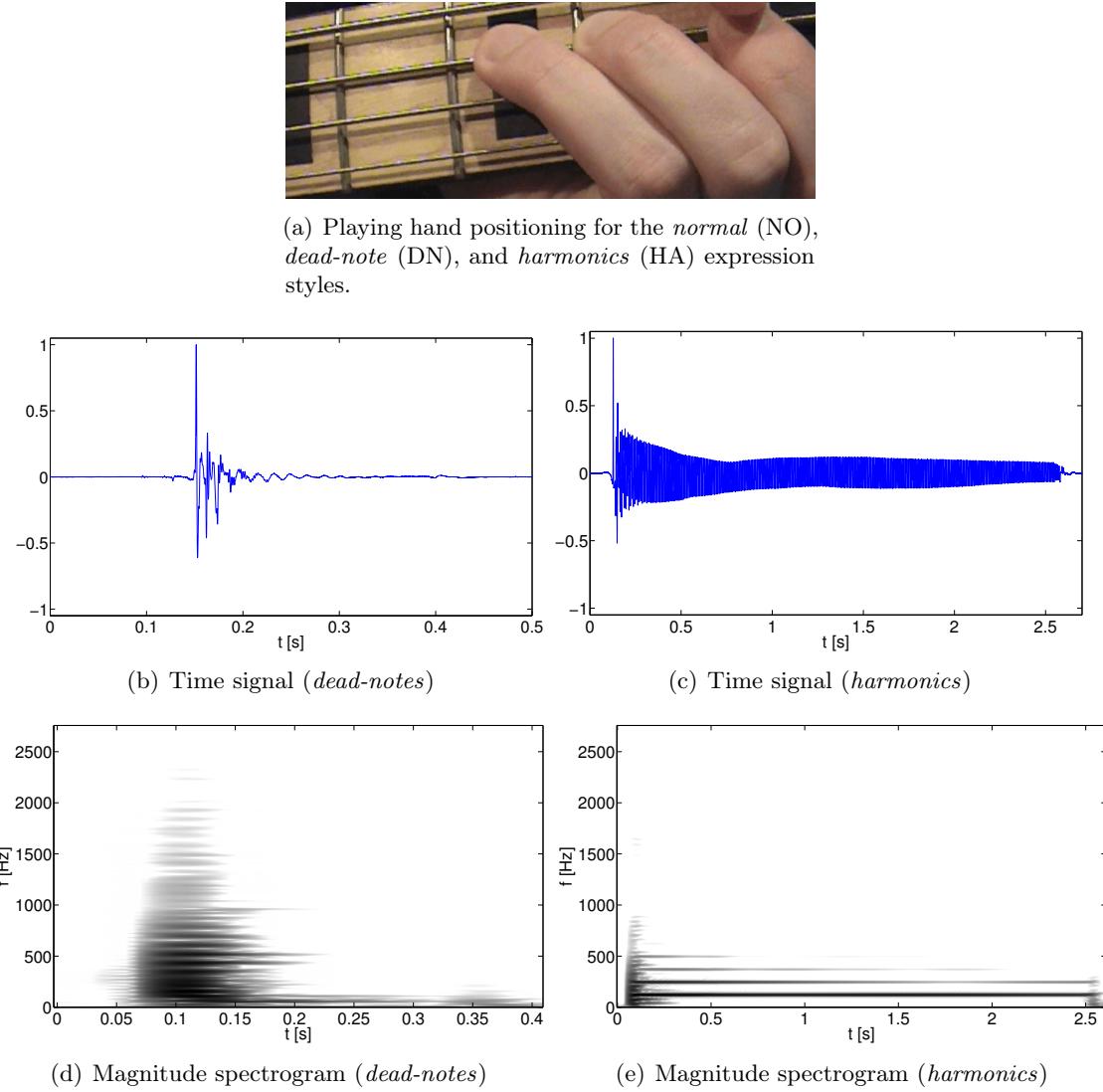


Figure 2.7: Playing hand positioning for the *dead-note* (DN) and *harmonics* (HA) expression styles. Two example notes are given as time signals (middle row) and magnitude spectrograms (bottom row).

2.1.3 Modelling the Sound Production

The bass guitar shares a similar sound production with other plucked string instruments such as the guitar or struck string instruments like the piano. As mentioned by Traube and Smith in [167], the sound production on a string instrument is characterized by a non-linear time-varying interaction between the finger and the string. In the following subsections, the basic string motion model will be explained and different aspects of the instrument timbre such as inharmonicity will be discussed.

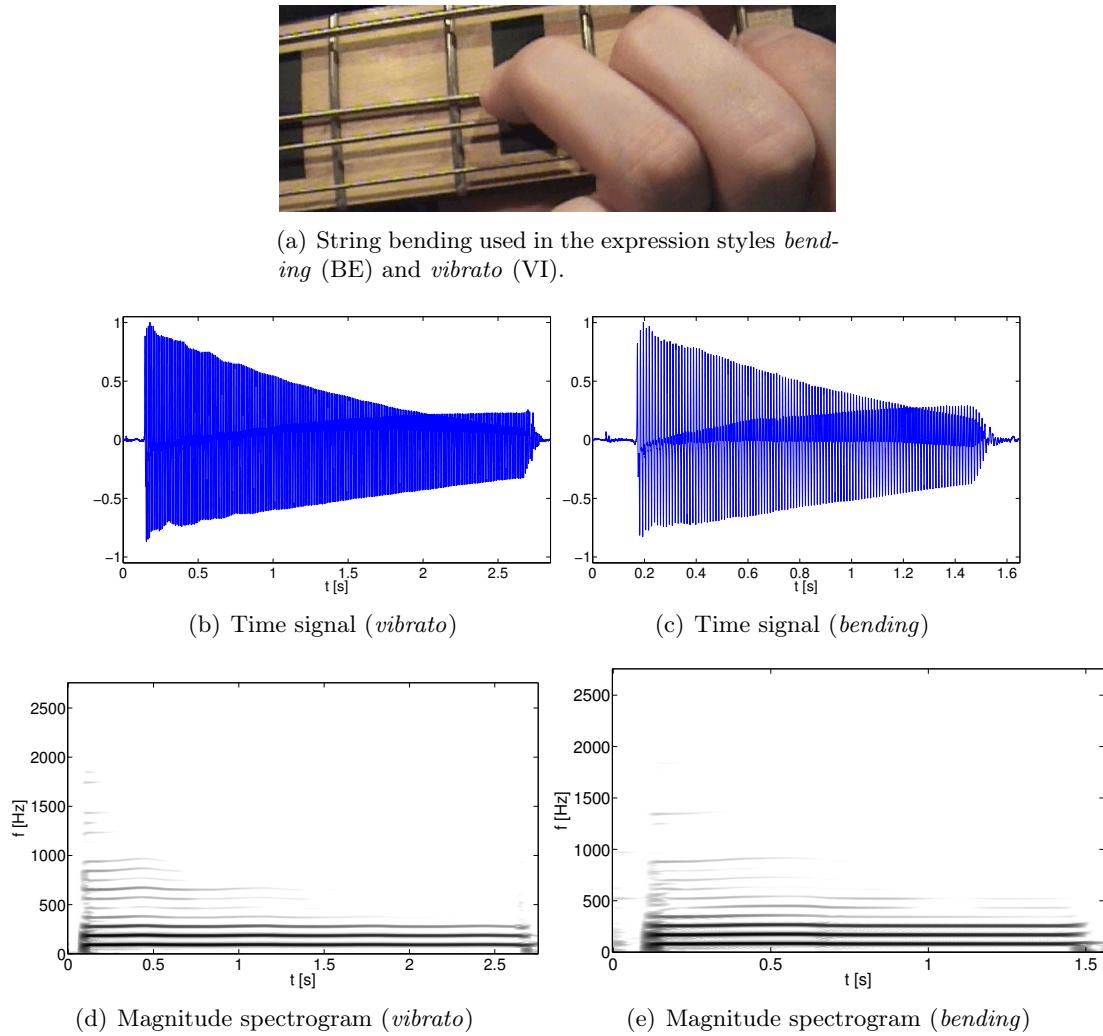


Figure 2.8: String bending using the playing hand applied for the *vibrato* (VI) and *bending* (BE) expression styles. Two example notes are given as time signals and magnitude spectrograms.

String Motion

The motion of an infinitely thin vibrating string is expressed by the differential equation

$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2}, \quad (2.1)$$

which describes the transversal string displacement y as a function of time t and geometric position x along the string. The propagation speed of the wave is denoted as c and depends on the mass density ρ (in kg/m) and the linear tension T (in N) of the string:

$$c = \sqrt{\frac{\rho}{T}} \quad (2.2)$$

D'Alembert (1717-1783) proposed the following general solution to (2.1):

$$y(t, x) = f_+(ct - x) + f_-(ct + x) \quad (2.3)$$

The string displacement can be modeled as a superposition of two waves that propagate in opposite directions along the string. The traveling waves are reflected on the string ends. Since the ends are fixed, the wave displacement is inverted after the reflection as shown in Figure 2.9. For the bass guitar, the reflection points are at the nut and the bridge as shown in Figure 2.1.

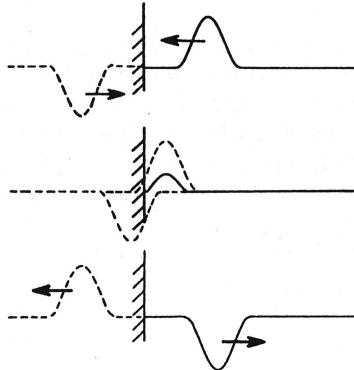


Figure 2.9: Reflection of a traveling wave (solid line) at a fixed end. The dotted line depicts an imaginary wave with opposite phase and traveling direction [55].

Equation (2.3) is the foundation of the physical modeling synthesis based on digital waveguide models as will be discussed in Section 11.2.

The functions f_- and f_+ generally can be of arbitrary nature. Based on (2.3), their initial sum at $t = 0$ has to be equal the initial string displacement when the string is plucked. The most common approach is to model both functions as a superposition of sinusoidal functions (*modes*) with the vibration mode frequencies

$$f_{\text{Mode},m} = \frac{m}{2l} \sqrt{\frac{T}{\rho}} = \frac{m}{2lc} \quad (2.4)$$

with m being the mode number and l being the string length. The first mode ($m = 1$) is denoted *fundamental frequency* and the higher modes ($m \geq 2$) are denoted *harmonics*. Figure 2.10 illustrates the vibrating modes of a string that is plucked in its center. All even-numbered modes are not excited here.

In this thesis, the *harmonic index* h is used to index the harmonic components instead. The fundamental frequency ($m = 1$) is indexed as f_0 with $h = 0$, hence

$$f_h \equiv f_{\text{Mode},(h+1)}. \quad (2.5)$$

The string vibration model discussed above depends on the following (idealized) conditions [55]:

1. uniform string with linear density ρ (this assumption is not fulfilled by the bass guitar string since it consists of a flexible core wrapped with a metal wire as discussed in Section 2.1.1),

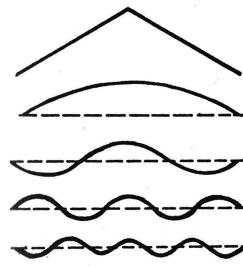


Figure 2.10: The upper graphic illustrates the initial string displacement if the string is plucked in the middle. Below, the resulting odd-numbered vibration modes ($m = 1, 3, 5\dots$) are shown [55].

2. small string displacement y (this assumption is violated for the slap techniques since this playing technique cause the string to collide with the upper frets as explained in Section 2.1.2), and
3. no string damping.

String Damping

In contrast to the simple string vibration model discussed before, different factors cause string damping when the electric bass guitar is played:

- finger damping of the plucking hand (for instance using the *muted* plucking style as shown in Section 2.1.2),
- finger damping by the playing hand (for instance using the *dead-note* plucking style as shown in Section 2.1.2),
- sympathetic coupling of the string vibration through the soundboard through the bridge (which can be omitted for solid-body instruments [91]),
- the internal friction due to the stiffness of the string,
- air friction (viscous dissipation), and
- direct sound radiation of the string.

String damping allows the performing musician to control the note duration. The string displacement decays approximately exponentially over time.

As shown for instance by Dayan and Behar in [39], the differential equation (2.1) can be extended by a “frictional resistance to motion proportional to the speed of the string elements” $R \frac{\partial y}{\partial t}$ to cope for the internal friction of the string. The authors showed that first, the resulting string vibration function $y(x, t)$ is multiplied by an decreasing exponential function and second, higher harmonics decay faster than lower harmonics since the damping is proportional to the harmonic index h .

Inharmonicity

As detailed in [55], (2.1) must be extended for real strings (i.e., for piano, guitar, or bass guitar) by a restoring force caused by the string stiffness as

$$\mu \frac{\partial^2 y}{\partial t^2} = T \frac{\partial^2 y}{\partial x^2} - E S_S r_G^2 \frac{\partial^4 y}{\partial x^4} \quad (2.6)$$

with

- E denoting Young's Modulus—a material property used to measure the stiffness of elastic materials,
- S_S denoting the cross-sectional area, which is proportional to the square of the string diameter d_S as $S_S = \pi d_S^2 / 4$, and
- r_G denoting the radius of gyration.

On an ideal vibrating string, waves travel without dispersion, i.e., the wave velocity is independent of the frequency. For the case of a stiff string however, the wave propagation is *dispersive* and the purely harmonic frequency relationship of an ideal string ($f_h = (h + 1)f_0$) is distorted. The harmonic frequencies are stretched towards higher harmonic indices as

$$f_h = (h + 1)f_0 \sqrt{1 + \beta(h + 1)^2}; \quad h \geq 0. \quad (2.7)$$

The *inharmonicity coefficient* β depends on different properties of the vibrating string:

$$\beta = \pi^2 E S K^2 / T l^2. \quad (2.8)$$

The string length l is approximately constant for all strings, the string diameter usually varies from 0.45 mm (G string) to 1.05 mm (E string) for an electric bass guitar.

In order to estimate the inharmonicity coefficient from instrument recordings, different methods such as cepstrum analysis, the harmonic product spectrum [60], the filter output of inharmonic comb-filters [61], and harmonic frequency deviations [147] were proposed in the literature. Hodgekinson et al. analyzed steel-string guitar tones in [79] and found that the inharmonicity factor β is time-dependent if the guitar strings are plucked hard. In this thesis, the time-dependency of the inharmonicity coefficient will not be considered.

Järveläinen et al. performed different listening tests about the audibility of inharmonicity towards humans [86]. The authors found that human listeners detect inharmonicity easier for lower notes compared to higher notes since auditory cues such as beating are better audible. In [87], the authors showed that the perception of inharmonicity is affected by different harmonics of the signal. In the field of MIR, algorithms for source separation, music transcription, and instrument recognition also analyze the inharmonicity to achieve better performance (see for instance [100], [181], or [23]).

String Beating & Two-stage decay

The plucked string vibrates in three modes, a transversal mode, a longitudinal mode, and a torsional mode. Generally, the latter two modes only have a minor affect on the string vibration and can be neglected [55]. The transversal mode can be separated into a horizontal and a vertical component that have slightly different vibration frequencies and different decay rates [85]. The superposition of both components can lead to an amplitude modulation of the harmonic envelopes with a very low modulation frequency. This non-linear effect called *string beating* is often incorporated into sound synthesis models of the guitar, e.g., by using the dual-polarization model proposed by Karjalainen et al. in [91]. As a result of the beating, a *two-stage decay* of the harmonics can sometimes be observed. However, in this thesis, the simplified assumption of a one-stage exponential decay is used as detailed in the next section. The interested reader is referred to [55] and [82] for more details about the instrument construction and sound production model of string instruments.

Two-stage Envelope Model

Throughout this thesis, a two-stage envelope model is used. The magnitude envelopes of all harmonics are segmented into an attack part and a decay part. During the attack part, the envelope is approximated using a linear increasing function. During the decay part, a decaying exponential function is used as approximation. Figure 2.11 illustrates the applied model for the fundamental frequency and the lowest three harmonics.

Non-linear effects that were discussed in the previous section are not considered in this model. As will be shown in Section 5.1, the experiments towards automatic classification of plucking and expression styles revealed that this simple model is sufficient for the goals of this thesis.

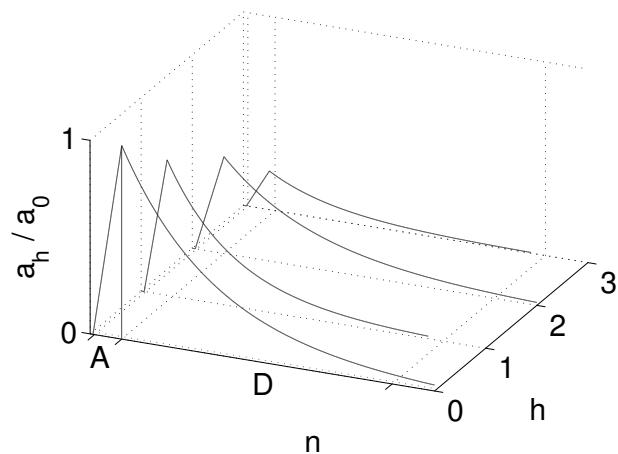


Figure 2.11: Two-stage envelope model used to approximate the magnitude envelopes of all harmonic components. The normalized linear magnitudes a_h/a_0 are shown over time frames n and harmonic indices h . The attack part (A) is approximated by a increasing linear function and the decay part (D) is approximated by a decaying exponential function.

2.2 Score & Tablature Representation of Music

Music pieces can be described as a mixture of multiple instrument tracks. Musicians need to be able to describe and notate these tracks in a written form in order to learn music, teach music, and communicate musical ideas. As will be discussed in Section 3.1, the process of music transcription extracts a symbolic representation that allows to describe musical recordings in an efficient and compact way. Each musical note is understood as a temporal event that can be characterized by a set of distinct parameters such as its pitch or its duration.

Conventional written notation can only represent musical expressiveness to a certain extend [91]. Symbolic music representations often fails to capture “non-symbolic properties such as expressiveness of a musical performance” [66]. Expressiveness can for instance be found in playing gestures of the musicians such vibrato or glissando that allow to modulate the fundamental frequency of a note (compare also Section 2.1.2 and Section 3.2.1).

In this section, the *score* and the *tablature* will be reviewed briefly as the two most popular written representation to notate basslines. Both will be compared based on their accessibility and popularity as well as their common fields of application. Figure 2.12 illustrates an example of a bassline shown as score and as tablature.

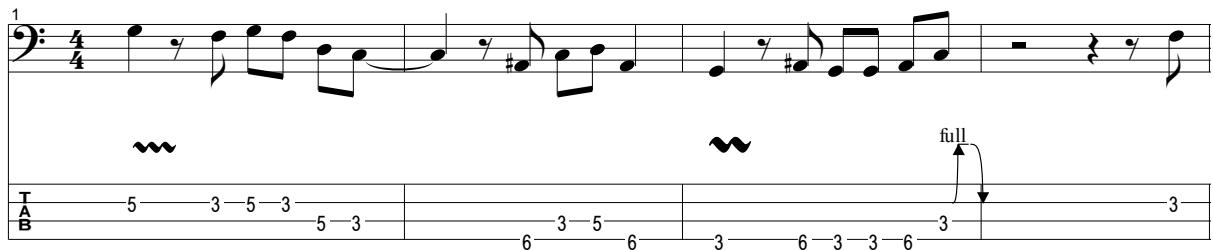


Figure 2.12: Score and tablature representation of a bassline [45]. While the score mainly encodes the note’s pitch, onset, and duration, the tablature encodes the corresponding string number and fret number on the instrument neck.

Score

The score-notation is the oldest and most popular form of music notation [69]. Scores offer a unified and well-established vocabulary for musicians to notate music pieces of different complexity for different musical instruments. Scores are widely available for many music genres although they usually need to be purchased.

As it can be seen in the upper subplot of Figure 2.12, the score encodes the note pitch as well as the note onset and offset time. The score can only provide a tempo-independent rhythmic representation. Hence, without the additional information of a global tempo, the note onset and offset times can not be represented in absolute time.

Tablature

The tablature representation is specialized on the geometry of fretted string instruments such as the guitar or the bass guitar. Notes are not visualized based on their pitch, onset, and duration

but based on their geometric position (string number and fret number) on the instrument neck. Therefore, additional information about the instrument’s string tunings is crucial to determine the note pitch values. Tablatures often include additional performance instructions such as playing techniques. For instance, in the lower subplot of Figure 2.12, two notes with vibrato are indicated with wavy lines in the first and third bar; one note with a whole-tone bending is indicated in the end of the third bar.

The main advantage of the tablature representation is that it resolves the ambiguity between note pitch and fretboard position as discussed in Section 3.2.2. Also, tablatures are very popular and freely accessible over the internet. As pointed out by Macrae and Dixon in [115], they “require no formal training to understand nor specific software to read or write”.

This benefit comes along with several problems. Tablatures are hard to read for musicians who play other instruments such as piano, trumpet, or saxophone since the geometry of those instruments is not comparable to the geometry of fretted string instruments. Furthermore, tablatures most often lack information about the duration of notes. The biggest problem is that tablatures are not standardized. Therefore, they are often incomplete or erroneous because they are mostly created by semi-professional musicians.

2.3 Machine Learning Methods

In this section, machine learning concepts as used in this thesis will be briefly introduced. All classification experiments described in Part I and Part II are based on cross-validation, which will be detailed in the next section. Section 2.3.2 will discuss different techniques for feature selection and feature space transformation and Section 2.3.3 reviews various classification algorithms.

2.3.1 Evaluation using Cross-validation

In classification experiments, a set of N_I items is represented vectors of N_F feature values in a feature matrix $\chi \in \mathbb{R}^{N_I \times N_F}$ and the corresponding class labels $c \in \mathbb{Z}^{N_I}$. For example, in Part I, classified items are individual note events in a bassline and the class labels are the corresponding plucking style, expression style, and string number. In Part II, items are complete basslines and the class labels are the corresponding music genres. Statistical models are commonly trained using a subset of the data denoted as *training set*. The trained model is used to predict the class labels of items from another (unseen) subset, which is denoted as *test set*.

In order to evaluate the performance of a given set of features in combination with a given classification model, *cross-validation* is performed. The complete data set is split into N_{Fold} sub-sets. The classification experiment that includes a training step and a prediction step is performed N_{Fold} times. In each cross-validation fold, another one of the sub-sets is used as test set and the remaining $N_{\text{Fold}} - 1$ sub-sets are used to train the classification model. Before applying feature selection, feature space transformation, and classification algorithms, the training set feature matrix is normalized to zero mean and unit variance along the feature dimensions. Using the mean and variance values obtained from the training set, the test set feature matrix is also normalized in each fold. If the percentage of class items varies over different classes, *stratified* cross-validation is performed, i.e., in each fold, it is ensured, that the relative number of class items in the training and test set is the same as in the original data set. The overall performance

of the experimental configuration is obtained by averaging for instance the classification accuracy, i.e., the portion of correctly classified items, over all folds.

2.3.2 Feature Selection & Feature Space Transformation

As discussed for instance in [154], the number of feature dimensions N_F should be significantly smaller than the number of items N_I in order to avoid an *overfitting* of the model to the given data. Overfitting occurs, when the predictive model is too complex, i.e., the number of model parameters is too high for a given number of observations. In case of overfitting, the model fails to learn the underlying relationship of a given training data set and cannot properly generalize in order to make correct predictions on unseen data. In order to reduce the number of features of a given data set, *feature space transformation* (FST) can be applied by mapping the original feature dimensions as linear combinations to a reduced number of feature dimension. Alternatively, *feature selection* (FS) can be used by keeping only those feature dimensions that best separate the items of different classes in the feature space. In the following sections, the feature space transformation methods Linear Discriminant Analysis (LDA) and Generalized Discriminant Analysis (GDA) as well as the feature selection method Inertia Ratio Maximization using Feature Space Projection (IRMFSP) will be briefly reviewed.

Linear Discriminant Analysis (LDA)

LDA is one of the most often used supervised FST methods [59]. The original feature dimensions are linearly mapped to a reduced feature space while guaranteeing a maximal linear separability between the classes. Here, the ratio of the between-class variance, i.e., the variance between the class centroids in the feature space, and the within-class variance, i.e., the variance among the items within each class, is maximized.

Generalized Discriminant Analysis (GDA)

If the classification problem is non-linear, a linear discrimination between different classes in the feature space is not possible. In order to overcome this problem, GDA first maps the original feature space into a higher dimensional feature space where a linear discrimination is possible [22]. A similar approach is used for the Support Vector Machines (SVM) classifier as will be shown in Section 2.3.3. The dot product in a high-dimensional space is replaced by a kernel function in the original space. This procedure is called *kernel trick* and allows to reduce the higher computational effort.

Inertia Ratio Maximization using Feature Space Projection (IRMFSP)

The IRMFSP algorithm was proposed by Peeters and Rodet in [137]. Similarly to LDA, the ratio of between-class variance and the total-class variance is used as optimization criterion. The IRMFSP algorithm iteratively selects a chosen number of features that maximize this criterion. After each iteration, the remaining feature dimensions are orthogonalized to the selected one. This ensures that no features are selected that provide similar information as the previously ones. In this thesis, the IRMFSP algorithms with the modifications proposed in [51] is used.

2.3.3 Classification

A classifier predicts a discrete valued target variable such as a class label from a set of observed feature values. The classifiers described in this section can be categorized as *generative* and *discriminative* classifiers. Generative classifiers learn the density distributions that underly each class in the feature space based on the corresponding observations (items) in the training set. Discriminative classifiers retrieve optimal boundaries between the classes in the feature space.

Support Vector Machines (SVM)

The SVM algorithm is a binary discriminative classifier that aims to find an optimal decision place between the feature vectors of two different classes [180]. The kernel trick explained in Section 2.3.2 is applied to make the classification problem linearly solvable. Different kernel functions exist for the SVM classifier. In this thesis, the Radial Basis Function (RBF) kernel is used.

This kernel function only requires two parameters—a regularization parameter C and a kernel parameter γ . During the classifier training, the two parameters are optimized based on a three-fold grid search. The search grid that is used for finding the optimal parameter values is refined in each fold. Finally, the kernel parameter configuration leading to the best classification performance on the training set is used.

Since multi-class problems are investigated in this thesis and the SVM only allows for binary class decisions, the “one-against-one” multi-class SVM approach is used as detailed in [34]. For each pair of classes in the training set, a binary SVM classifier is trained using the corresponding samples from the training set. In order to predict the class of an unseen observation, the results of the binary classifiers are collected as votes and the class having the highest number of votes is assigned.

Gaussian Mixture Models (GMM)

Gaussian Mixture Models (GMM) are generative classifiers. The probability density function (PDF) underlying the distribution of each class in the feature space is modeled as a weighted sum of single Gaussian densities. The classifier parameters, i.e., the mean and covariance matrix, are commonly estimated using the Expectation-Maximization (EM) algorithm [44] as will be explained in Section 4.1.6.

Naive Bayes (NB) Classifier

Naive Bayes classifiers (NB) are simple probabilistic classifiers. The classification decision is only based on the means and variances of the classes for different feature dimensions. NB rely on the assumption that all feature dimensions are statistically independent. Despite this strong simplifying assumption, NB classifiers have been shown to perform well for different real-world tasks [185].

k -Nearest Neighbors (kNN) Classifier

The k -Nearest Neighbors (kNN) classifier assigns unseen samples to a class based on the sample’s proximity to the closest training examples in the feature space [47]. Here, the Euclidean distance

measure is used. The number of nearest neighbors k has influence on the level of generalization.

Classification and Regression Tree (CART)

The Classification and Regression Tree (CART) algorithm [26] is a non-parametric data mining algorithm to generate decision trees. No assumption towards the distribution of the feature vectors is made. A decision tree consists of consecutive binary decisions based on thresholding a particular feature dimension. Each tree node is split into two child nodes. The features having the best discriminative power with respect to the classes are selected first.

The number of decision levels is determined based on a stopping criterion, such as for instance a minimum number of items per node to be still considered for splitting. The generalization properties of the decision tree are controlled in a cross validation scenario, where the tree is pruned to a certain level in order to prevent overfitting the training data.

Part I

Automatic Transcription of Bass Guitar Tracks

Preface

In the first part of the thesis, the automatic transcription of bass guitar tracks will be discussed. The electric bass guitar was chosen as the most commonly used bass instrument in various music genres.

Music transcription denotes the process of estimating parameters that are suitable to characterize a music recording [102]. Commonly, these parameters characterize each played note on the abstract level of a musical score, which does not take the acoustic properties of the applied musical instrument into account. In this thesis, the conventionally used set of note parameters, i.e., the note pitch, onset, duration, and loudness, will be extended by a set of *instrument-specific* note parameters, i.e., the applied *playing techniques* (plucking style and expression style) and the applied *geometric position on the instrument neck* (fret number and string number), which describe *how* and *where* the musician plays certain notes on the instrument neck.

These additional parameters are especially relevant for string instruments, which allow to play notes of the same pitch at different positions on the instrument fretboard (see Section 3.2.2 and Section 4.1.10). Also, the use of different playing techniques is an important part of the “expressive repertoire” of each performing musician and has a major impact on the sound of a bass guitar recording [182].

Nowadays, music pieces are most often recorded as *multi-track recordings* with each instrument being recorded as an individual audio track. For the automatic transcription, the availability of isolated instrument tracks facilitates the analysis step since no or only minor overlap exists between different sound sources. In order to transcribe mixed audio recordings, *source separation* algorithms first need to be applied in order to isolate individual instruments from the mixed audio signal. In this thesis, only isolated bass guitar tracks will be analyzed in order to fully focus on the improvement of their parametric description.

The first part of the thesis is structured as follows. The related work on automatic bass transcription as well as on the estimation of playing techniques and fretboard position from string instrument recordings will be reviewed in Section 3.1 and Section 3.2. A novel algorithm for bass guitar transcription will be presented in Section 4.1. Furthermore, Section 4.2 introduces two novel audio datasets, which were published as evaluation benchmarks for the discussed signal processing tasks.

3 Related Work

3.1 Bass Transcription

Definition *Automatic music transcription* aims at describing a time-continuous audio signal as a sequence of acoustic events that correspond to musical notes. Note events are commonly described using the parameters pitch (corresponding to the fundamental frequency), onset (note begin), duration (note length), and loudness. In this thesis, these parameters will be referred to as *score parameters* since they represent the main information given in a musical score. The automatic transcription of a complex music piece is a very demanding task that includes multiple analysis steps such as instrument recognition, source separation, rhythmic analysis, and multi-pitch estimation [102].

Categorization Most automatic music transcription algorithms are designed “from an instrument-free perspective” [114] and can be categorized into the following transcription tasks:

- Drum transcription—transcription of percussion instruments
- Melody transcription—transcription of the singing voice or the most salient instrument voice (this task is usually considered as single-pitch estimation problem)
- Bass transcription—transcription of the bass instrument (also considered as single-pitch estimation problem)
- Polyphonic transcription—considered as multi-pitch estimation problem and therefore as the most challenging task

Other authors propose algorithms that are tailored to the transcription of specific instruments such as the violin [183], the cello [35], the guitar [54], or the piano [49]. All publications discussed in this section focus on a purely symbolic signal representation of the bass instrument track. None of the publications take specific instrument properties of the electric bass guitar into account. The interested reader is referred to [143] and [102] for a detailed review of other existing music transcription algorithms.

Challenges Several challenges need to be faced along with the automatic transcription of a music recording. First, the transcription of complex music recordings is error-prone and often ambiguous, regardless of whether it is performed by human experts or by automatic algorithms. Most pitch estimation errors result from overlapping harmonic signal components in the spectrum. The overlap is caused by the fundamental frequency relationships of musical notes for various intervals. Therefore, particularly in polyphonic audio signals with multiple notes sounding simultaneously, the estimation of the fundamental frequency often leads to ambiguous results.

Second, depending on the type of instrument, the harmonic frequencies can deviate from a pure harmonic relationship [55]. As discussed in Section 2.1.3, notes played on string instruments such as the bass guitar exhibit an *inharmonic* relationship between the harmonic frequencies, which has to be taken into account in the music transcription process.

In [66], Goto lists three general challenges in transcribing music recordings. First, the number of instruments is generally unknown. Second, the instrument characteristics of the present instruments such as being harmonic or percussive are generally unknown. Finally, the assignment between detected harmonic frequency components and the present instruments is generally unknown. Goto also discusses the problem of the “missing fundamental”, i.e., some musical instruments show a harmonic structure with barely no spectral energy at the fundamental frequency, which complicates the precise detection of the fundamental frequency course over time. However, in the experiments performed in this thesis, this problem was never observed for bass guitar recordings.

Applications Music transcription algorithms have many potential applications. Commonly, music students can either purchase commercially available transcriptions or transcribe songs by ear, which is both time consuming and error-prone. In music education applications such as Songs2See [45], the automatic transcription of single instrument tracks allow to generate scores or tablatures from arbitrary music pieces. At the same time, the transcription results can be exploited in source separation algorithms in order to generate playback tracks [30] by removing the transcribed instrument from the mixed signal.

The automatic extraction of musical notation can facilitate the musicological analysis of large music collections. Even if the transcription results are partially erroneous, they still can provide a valuable basis for a semi-automatic transcription procedure that includes a manual proof-reading and correction stage in the end [8].

3.1.1 Common Presumptions

Bass transcription algorithms are commonly premised on two presumptions [77]. First, the bass instrument is the lowest pitched harmonic instrument in a music ensemble and plays the dominant melodic line in the lower pitch register. The common fundamental frequency range of the bass instrument is between around 40 Hz and 400 Hz. Second, the bass instrument plays monophonic melodies, i.e., note sequences with no temporal overlap between consecutive notes. In music practice, this assumption holds true for most of the music genres [182]. In some music styles such as funk or jazz however, basslines can contain polyphonic elements such as double-stops or even three-voiced and four-voiced chords. These presumptions are exploited in different steps in the transcription process as will be discussed in the following sections.

3.1.2 Algorithmic Steps

The most important processing steps of bass transcription algorithms are shown in Figure 3.1 and will be detailed in the following sections. State-of-the-art bass transcription algorithms will be discussed and classified accordingly.

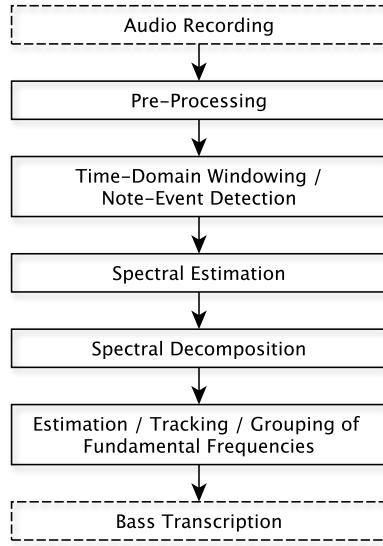


Figure 3.1: Algorithmic steps of bass transcription algorithms.

Pre-processing

Filtering Due to the low fundamental frequency range of bass notes, *down-sampling* can be applied to the analyzed audio-signal. Considering for instance a down-sampling to a sampling frequency of $f_s = 11.025 \text{ kHz}$, the first 10 harmonics of a high bass note with a fundamental frequency of $f_0 = 382 \text{ Hz}$ (highest note on a four-string bass guitar with 24 frets) are still below the Nyquist frequency and can be detected as peaks in the magnitude spectrum. Using down-sampling has two advantages. First, it increases the computational efficiency of the transcription algorithms and second, harmonic signal components from other instruments at higher frequencies are filtered out. However, down-sampling decreases the temporal resolution that can be achieved.

Dittmar et al. apply a strong down-sampling by factor 32 in [46]. They only consider harmonic components up to around 690 Hz as fundamental frequency candidates. Other authors similarly perform an initial low-pass or band-pass filtering to limit the analyzed frequency range [66, 155]. Instead of assuming a fixed upper frequency limit for potential fundamental frequency values of the bass notes, Rynänen and Klapuri estimate a variable upper f_0 -limit in each time frame, which depends on the given musical context, i.e., the notes played by other instruments [152].

Source Separation *Source separation* algorithms can be applied to mixed audio signals to emphasize the instrument track signal that is targeted for automatic transcription. The low-pass filtering discussed in the previous section can be interpreted as a first approach for source separation. Often, other instruments are suppressed in the spectral representation of the mixture signal before the bassline is transcribed. Uchida and Wada initially transcribe the melody instrument track and remove its harmonic components from the mixture signal via spectral subtraction [175]. In [170], Tsunoo et al. initially remove signal components from percussive instruments using the harmonic/percussion sound separation (HPSS) algorithm proposed in [130]. The basic idea is to

remove transient spectral components with a wide-band energy distribution. These components can be usually associated to percussive instruments or typical signal transients during the attack phase of harmonic instruments.

Spectral Whitening The harmonic magnitudes of bass notes are time-dependent and their relationship strongly depends on the bass instrument itself. To cope with that problem, Ryyränen and Klapuri propose to initially apply *spectral whitening* to suppress timbral characteristics of the applied bass instrument. Spectral whitening flattens the spectral energy distribution over different frequency bands and makes the transcription algorithm more robust to different instrumentations of music pieces [101].

Time-domain Windowing & Note Event Detection

The choice of the window size has a strong influence on the overall performance of the transcription system. A large window size results in a higher achievable frequency resolution and noise-robustness, whereas a small window size allows to better capture short-time fluctuations such as note vibrato in the signal [155]. Note detection is often performed in the time domain by envelope detection methods. Dittmar et al. propose to apply half-wave rectification, low-pass smoothing, envelope differentiation, and detection of rising slopes with a dynamic threshold criterion to detect note events [46]. Hainsworth and Macleod apply low-pass filtering at 200 Hz, signal smoothing using a Gaussian kernel, and a subsequent peak picking using a dynamic threshold criterion [77]. Other authors perform note detection in the spectral domain, usually after several frame-wise f_0 -estimates are grouped to note events [66, 152, 153]. This approach can be considered to be more robust for multi-timbral music, where note events from percussion instruments can be mistaken as note onsets of the bass instrument.

Spectral Estimation

Due to its computational efficiency, the short-time Fourier transformation (STFT) is most often used for *spectral estimation* [46, 77, 152, 153, 175]. However, the spectral leakage effect limits the achievable frequency resolution especially in lower frequency bands. Other spectral estimation techniques such as the instantaneous frequency (IF) spectrogram are applied to improve the achievable frequency resolution in the lower frequency bands [46, 66]. Tsunoo et al. use a wavelet-based constant-Q transform in [171] to obtain a time-frequency representation with a logarithmically spaced frequency axis. Using a linearly spaced frequency axis, the harmonic peak frequencies in the magnitude spectrum are stretched towards higher frequencies. The main advantage of a logarithmically spaced frequency axis is that the harmonic peak structure remains almost constant and is just shifted to higher frequencies.

Spectral Decomposition

Spectral decomposition algorithms try to group multiple frequency components to a common fundamental frequency f_0 based on a shared harmonic frequency relationship. Frame-wise f_0 estimates can be grouped to note events based on different criteria. The f_0 progression of a note is assumed to have a continuous shape over time. In most transcription algorithms, the detected

notes are not represented by the estimated f_0 progression but instead by a distinct f_0 value, which represents the note pitch.

Instead of performing spectral decomposition in each spectral frame, Hainsworth and Macleod propose to align the start times of the analysed frames for f_0 estimation to the previously estimated note onset times [77]. Similarly, Dittmar et al. propose to initially classify the spectral frames into harmonic and percussive frames and only consider the harmonic frames for the f_0 estimation process [46]. This way, potential misclassification between bass notes and percussive bass-drum notes are avoided.

In order to estimate the fundamental frequency in a given time frame, different authors propose to extract a *harmonic saliency function*, which provides a likelihood-measure for a given fundamental frequency value at a given time. Klapuri and Rynnänen compute the harmonic salience of a f_0 candidate by summing up the spectral energy at the frequency bins of the corresponding harmonic frequencies [101, 152]. Salamon and Goméz extract a saliency function from the mid-level chromagram-based Harmonic Pitch Class Profile (HPCP) in [155]. The authors emphasize that the chosen representation is robust against variation in tuning, timbre, and dynamics. The HPCP is computed in the bass frequency band between 32.7 Hz and 261.6 Hz (as initially proposed by Goto in [66]), using a rather high resolution of 120 bins per octave¹ in order to capture modulations techniques of the fundamental frequency such as vibrato and glissando. Since the HPCP is interpreted as a harmonic saliency function, the f_0 estimate is obtained by locating the highest peak in the HPCP in each frame. Salamon and Goméz only aim at extracting the fundamental frequency course and no note events. Hence, no further processing steps such as note grouping are performed.

Especially for multi-pitch estimation, i.e. the transcription of polyphonic instrument tracks, the matrix factorization techniques Non-Negative Matrix Factorization (NMF) and Probabilistic Latent Component Analysis (PLCA) are successfully applied to decompose spectral representations of an audio signal into various components, which are commonly the notes and their harmonic structure [58, 144, 181]. However, these methods are not used in bass transcription algorithms so far.

Estimation, Tracking, and Grouping of the Note Fundamental Frequencies

Rynnänen and Klapuri presented a hybrid transcription framework for bass and melody transcription in polyphonic music in [152, 153]. The authors propose to combine two modeling approaches for music transcription. First, note events are modeled by the temporal progression of two acoustic features, the *pitch saliency* functions as explained in the previous section and an *accent signal* that captures the amount of “spectral novelty” in a given time frame. Second, the authors train a musicological model to use musical context knowledge. In particular, transition probabilities between different note pitch values are derived from the estimated key of a song. Viterbi decoding is applied to retrieve the bassline transcription, which here is the optimal sequence of bass notes and intermediate rests. To model the temporal progression of single notes, three-state left-to-right Hidden Markov Models (HMM) are trained. The three states relate to the attack, sustain, and release part of the note envelope.

¹Commonly used chromagram resolutions in MIR tasks such as audio-to-MIDI alignment, music synchronization, or chord estimation are 12 or 36 bins per octave [126].

In the bass transcription algorithm proposed by Hainsworth and Macleod in [77], an *amplitude confidence value* is computed for each frame-wise fundamental frequency candidate. Therefore, a comb filter is tuned to the fundamental frequency value and then used as filter on the magnitude spectrum at a given frame. A weighted sum of the harmonic magnitude values a_h is computed whereas the magnitude of the fundamental frequency a_0 is emphasized and odd harmonics are slightly higher rated than even harmonics. Then, different note hypotheses are generated by tracking f_0 candidates over time. Two algorithmic steps for tracking and trimming of f_0 hypotheses are applied to derive the final note estimates. A likelihood measure is computed for each hypothesis from the amplitude confidence measure and the duration of a hypothesis related to the distance between the note onset candidate to its successor. Since the bassline is assumed to be monophonic, the hypothesis with the highest likelihood is selected for each onset candidate.

Goto proposed the “PreFEst” (predominant-F0 estimation) algorithm in [66], which is used for a combined transcription of the main melody and the bassline. First, frequency components are extracted by using an STFT-based multi-rate filter bank and computing the instantaneous-frequency (IF). The overall spectrogram is modeled as a weighted sum of different tone models, which are combined probability density functions (PDFs) of fundamental frequency components and the corresponding harmonics. The spectral decomposition is based on the Expectation-Maximization (EM) algorithm. The tone models are adapted to the actual harmonic structure in the spectrogram. Based on the extracted harmonic saliency function, the most salient peaks are tracked over time and grouped to note events. Goto refers to this step as “multiple-agent structure”.

Post-processing

After the transcription steps are performed as detailed in the previous sections, most algorithms contain a final note-selection step in order to reduce the number of erroneous note events. For instance, Hainsworth and Macleod filter out notes that likely result from spurious onsets or from reverberation [77] in the end of their proposed bass transcription algorithm.

Evaluation

The evaluation of automatic music transcription algorithms requires a *dataset* of music recordings and corresponding reference transcriptions, i.e., the *ground truth annotation*. Moreover, a suitable *evaluation metric* must be defined that measures the performance of the transcription algorithms by comparing the automatic transcription results with the ground-truth annotations.

The annual MIREX competition poses several tasks related to the field of MIR, among others a melody transcription task (“Audio Melody Extraction”) [141] and multi-pitch estimation task (“Multiple Fundamental Frequency Estimation & Tracking”). A particular task for bass transcription was not posed so far.

Publications on bass transcription (as discussed in the previous sections) mostly use individual audio datasets that were not made publicly available [46, 66, 77]. These datasets vary in size and music genres but cannot be used to compare the performance of different bass transcription algorithms.

Uchida and Wada artificially generated their audio data for evaluation by re-synthesizing bass-lines from MIDI files [175]. This approach allows to use the given MIDI parameters as ground truth data for the score parameters to be transcribed. However, the (synthetic) audio data is not as realistic as real music recordings.

The only dataset used by multiple authors is the RWC Music Database [67, 68], in particular the “Popular Music Database” and the “Music Genre Database”. The dataset was introduced by Goto et al. in 2004 and has become the first widely used large-scale database that was used as benchmark in MIR tasks such as genre classification, instrument recognition, and music transcription. The database contains around 350 songs with both the audio signal and a corresponding MIDI file with ground truth transcription annotations made available. This dataset was used for evaluation purpose in [152, 153, 155]. Chapter 5 will give details on the applied evaluation measures in the performed experiments in this thesis.

3.2 Instrument-level Parametrization of String Instrument Recordings

The *score-related parameters* discussed in Section 3.1 only provide an abstract representation of an audio recording. Additional *instrument-related* parameters are necessary for a better description of the musical performance on the instrument. Figure 3.2 summarizes different criteria that are used to categorize publications in this section towards estimation of playing techniques and fretboard position from string instruments.

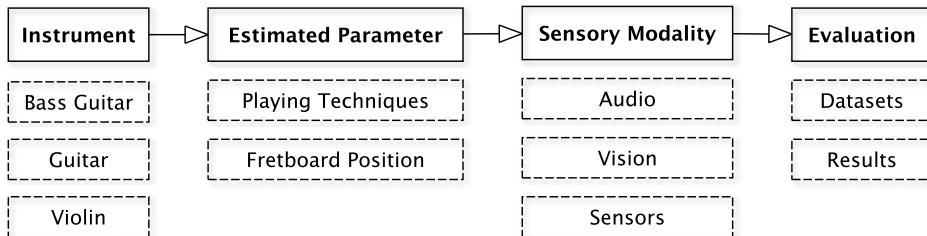


Figure 3.2: Categorization criteria of related work towards the estimation of the playing technique and fretboard position from string instrument recordings.

Instrument

Only a few publications focus on the extraction of instrument-related parameters from bass guitar recordings so far. In contrast, related string instruments such as the guitar and the violin are often analyzed in the literature. Both instruments share a similar sound production process and many playing techniques with the bass guitar [55]. Therefore, publications with focus on violin and guitar analysis are included in the literature review.

Estimated Parameter

The related work towards the estimation of playing techniques and the fretboard position will be discussed separately in Section 3.2.1 and Section 3.2.2.

Sensory Modality

Different *sensory modalities* are used in the literature for the purpose of data acquisition. As discussed in [32], the acquisition of physical and perceptual parameters from musical instruments can be categorized into *direct acquisition methods*, which are based on *sensors* that are attached to the instrument, and *indirect acquisition methods*, which are based on *audio* and *visual* analysis of recorded musical performances on the instrument. In addition, some authors propose *multi-modal approaches* that are based on a data fusion of different sensory modalities in a complementary way. The focus in this thesis will be on audio-based methods. Vision-based and sensor-based analysis methods will be discussed only briefly for comparison.

The three sensory modalities audio, vision, and sensors have different *advantages* and *disadvantages*. Audio analysis methods only require an instrument pickup or a microphone for data acquisition. Most musicians are familiar to this recording setup. However, audio-based analysis can only be used to analyze *perceptual parameters* that are related to the sound production mechanism of the musical instruments. In order to analyze *gestural parameters*, which characterize the movement of the musician during the musical performance, visual or sensory analysis is required.

Visual analysis often relies on cameras that are attached to the instrument or positioned close to the performing musician. If the fretboard is recorded, the detection performance can be impaired by bad lighting, masking of the fretboard by the playing hand, or the musician's movement itself.

Sensory analysis provides very accurate time-continuous measurements. Also, the movement data measured by motion capturing is closely related to the musicians playing gestures. Nevertheless, motion capture analysis often requires complex measurement setups that cannot be installed in every environment. Signals from capacitive sensors that are used to measure the hand pressure on the instrument fretboard are often noisy and exhibit crosstalk between spatially adjacent sensors. Both cameras and sensors, which are mounted to the instrument, can be intrusive to the musicians and hinder their performance.

Evaluation

For the estimation of playing techniques as well as for the estimation of the fretboard position, the applied data sets used for the evaluation experiments and the obtained results are discussed.

3.2.1 Estimation of Playing Techniques

A rich repertoire of playing techniques exists for the bass guitar, the guitar, and the violin. The publications discussed in this section analyze different string instruments. The sound production of these instruments can be separated into two *physical gestures*—a *plucking gesture* and an *expressive gesture* as discussed in Section 2.1.2. These gestures affect the sonic properties of the recorded instrument notes in a unique way, which allows human listeners to recognize and distinguish different playing techniques.

In the following sections, methods for estimation of playing techniques from bass guitar, guitar, and violin recordings will be reviewed. Special focus will be put on the investigated playing techniques and the applied parameter estimation methods. Finally, the datasets used for evaluation and the obtained results will be compared briefly.

Bass Guitar

To the best knowledge of the author, no other publication than [13] and [7] focused on the automatic recognition of bass guitar playing techniques so far. The contribution of these publications will be discussed in Chapter 4.

Guitar

The guitar is the most similar musical instrument to the bass guitar and shares most of its playing techniques [128].

Data Acquisition For *audio analysis*, acoustic guitar signals are usually captured with a microphone or a piezo pickup system.² Electric guitar signals are recorded with an electro-magnetic pickup attached to the instrument body under the strings. Reboursière et al. use a hexaphonic piezo pickup³ to capture the string vibrations of the six guitar strings individually [149]. This procedure allows to analyze strictly monophonic audio signals without temporal and spectral overlap between notes.

Visual analysis is usually based on one or multiple cameras recording the fretboard and the playing hand [36]. These cameras are mounted on the instrument neck or positioned in front of the performing musician [24].

For the *sensor-based analysis*, Guaus et al. mount capacitive sensors to the first 10 frets of a classical guitar [73, 74] to detect hand movements on the fretboard. The same setup is used by Torres in [165]. A completely different approach is proposed by Karjalainen et al. in [91]. Instead of playing a real guitar, the user plays an “air guitar”, i.e., the user only imitates to play a guitar by hand movement. The user wears gloves with attached magnetic sensors that allow to track the gloves’ location and orientation. The hand movement is optically tracked using cameras and playing gestures are recognized from the musician’s movement.⁴ Norton used a motion capture setup as well as “data gloves”, i.e., gloves that are equipped with sensors that the musician wears during the performance, to analyze typical guitar playing gestures in [128].

Playing Techniques & Parameter Estimation Methods The amount of *damping* that is used to play notes on a guitar is analyzed by Erkut et al. in [50]. Based on an STFT of the audio recording, the magnitude, frequency, and phase trajectories of the note harmonics are tracked over time. The decay rates of the individual harmonics are estimated to quantify the amount of

²Piezo pickups convert acoustic vibrations on the body surface of the instrument into an electric signal. In contrast to electro-magnetic pickups, piezo pickups can be used for acoustic instruments such as the classical guitar that have non-metallic strings.

³A hexaphonic pickup allows to capture the vibration of each individual string as a separate signal.

⁴The user is provided with sonic feedback from a guitar synthesis algorithm that is controlled by the detected gestures.

damping. The authors also analyze the *loudness* of different notes, which directly relates to the plucking force of the musician.

Frequency modulation techniques allow the musician to change the note pitch continuously as previously discussed in Section 2.1.2. These techniques are very widely used in guitar performances. Various publications analyze the estimated fundamental frequency course of single notes to detect the playing techniques *vibrato* [50, 74], *bending* [149], or *slides* (also denoted as glissando, appoggiatura) [133, 134, 149].

The timbre of guitar notes not only depends on the plucking style but also on the *plucking position* on the string. Traube and Depalle estimate the plucking point on the guitar and classify the two classical guitar playing techniques *sul ponticello* (plucking near the bridge) and *sul tasto* (plucking near the fretboard) [166]. Similarly, Orio analyzes how finger-style playing at different plucking positions affects the guitar note sound in [131].

In between consecutive note events, different *note transition techniques* can be applied. Common techniques are for instance the *slide* or the *hammer-on* and *pull-off* techniques (also referred to as *legato*) [74, 134, 149]. Note onsets are usually detected as the note attack transients, i.e., signal frames in the spectrum with a non-harmonic, wide-band characteristic. Various detection functions such as the High Frequency Content (HFC) measure or the spectral flux are used to identify non-harmonic frames. The f_0 curve is commonly extracted with automatic transcription algorithms, Özaslan et al. for instance apply the YIN algorithm [38]. Finally, the applied note transition technique is classified based on the shape of the f_0 curve. A smooth transition between two notes indicates the slide technique and a sudden change indicates the hammer-on or pull-off techniques.

Guitar playing techniques are usually related to single musical note events. However, some techniques describe the *playing of note sequences* in a particular way. Erkut et al. analyze repeated plucks in [50], Guaus et al. analyze the hand movement if *grace notes*⁵ are played [74]. In [108], Laurson et al. investigate the *rasgueado* technique, a rhythmically complex plucking technique, which is commonly used on classical guitars in flamenco music. The rasgueado technique is characterized by a sequence of fast consecutive note plucks with the finger nails and an up-wards and down-wards movement of the plucking hand. The authors emphasize that the “attack transients caused by nail-string contacts are wideband signals, which appear as short events even at high frequencies”. In comparison, bass guitar notes played with the *slap* techniques explained in Section 2.1.2 show similar spectral characteristics.

Evaluation Datasets & Results Most publications analyze multiple playing techniques. Taxonomies with two up to six different techniques are used. All publications analyze isolated instrument recordings with no overlap of other instruments. Most datasets used for evaluation are rather small and contain around 40-120 recordings [131, 133, 134] of single notes or simple melodies. Exclusively, Reboursière et al. created a larger dataset of 2832 note recordings in [149], which has a comparable size as the datasets published in this thesis (see Section 4.2).

Reboursière et al. report accuracy values between 93.0 % and 100.0 % for the classification of 4 expression styles (hammer-on, pull-off, bending, slide) and accuracy values between around 87.0 % and 100.0 % for the classification between the 2 plucking styles normal and muted in [149].

⁵Grace notes are short ornamentations in front of longer notes.

Fot the classification of note transition techniques such as slides and hammer-on and pull-off, Özaslan et al. achieved precision values between around 72.0 % and 84.0 % [133, 134].

Violin

Data Acquisition Similarly to the guitar, violins are either recorded using external microphones [20, 106] or piezo-electric pickups [116, 118]. In contrast, the violin is most often played using a bow. The plucking hand is only used for playing in case the pizzicato technique is used. This technique is comparable to the finger-style plucking style for the guitar and the bass guitar.

Multi-modal approaches that combine audio analysis and sensor-based analysis are proposed in the literature. The bow movement is crucial for the understanding of the violinist's performance gestures. Therefore, movement sensors are used, which are attached to the bow to measure its velocity, force, tilt, and distance between the bow-string contact and the bridge [31, 32, 116]. Maestre et al. use a two-sensor 3D tracking system based on electro-magnetic field (EMF) sensing to capture violin playing gestures [117]. Leroy et al. propose to use an optical pickup instead of commonly used electro-magnetic or piezo pickups for a laser-based pitch tracking in [110].

Playing Techniques & Parameter Estimation Methods *Bowing* is the most typical violin playing technique. It is analyzed using either movement sensors [31, 116] or audio analysis [32, 106].

The largest taxonomy of violin playing techniques so far is used by Barbancho et al. in [20]. The authors present an algorithm to automatically classify 7 different playing techniques applied in violin recordings—*pizzicato* (corresponds to the *finger-style* plucking style), *tremolo*, *spiccato*, *flageolett* (corresponds to *harmonics* expression style), *détaché* with and without accent, and *vibrato*. Both time-domain audio features (attack, sustain, and release time) and spectral features (pitch, spectral width) from a FFT warped to a logarithmic frequency axis are computed. In order to automatically classify the applied playing technique from the feature values, the authors use an “expressive decision flowchart”, i.e., a multi-stage decision tree algorithm, to derive a class decision based on feature values and corresponding thresholds. In contrast, Krishnaswamy and Smith only distinguished two techniques, plucking and bowing of the string [106].

Most publications use an initial transcription stage to detect the note events and their parameters [20, 114]. Afterwards, different low-level audio features such as the note envelope and the attack, sustain and release time [20] or features based on modulation and inharmonicity [114] are computed to model different playing techniques.

In terms of spectral representations, STFT [32, 106] along with extensions such as linear interpolation [183] or frequency warping [20] is used for fundamental frequency detection and the tracking of harmonics.

As for the guitar, the *vibrato* technique is usually detected based on the f_0 curve [183]. Interestingly, Barbancho et al. instead try to capture the spectral width around the harmonic peaks as a feature, since it is larger than for notes played without frequency modulation (due to frequency smearing).

Yin et al. used three assumptions to tune their algorithm for violin analysis [183]. First, only a limited pitch range (G₃-G₆) is considered for possible pitch candidates. Second, a fixed harmonic structure is assumed with most of the spectral energy being located in the fundamental frequency.

Finally, the authors search for monophonic melodies, i.e., note sequences with note temporal and spectral overlap between adjacent note events.

Hähnel and Berndt analyze different note articulation techniques such as *tenuto*, *neutral*, *staccato* & *staccatissimo*, *bow vibrato*, and *portato* in [76]. The authors focus on the note envelope, duration, and loudness. However, no automatic classification of note articulations is performed.

Evaluation Datasets & Results Loscos et al. state that the low-level audio descriptors used in [114] were tested with a rather large dataset of 1500 violin notes and double-stops. However, no quantitative evaluation results are reported.

Barbancho et al. performed an evaluation both on isolated violin notes from the RWC instrument sample database [67] as well as from home-recordings. However, the authors do not mention the size of the applied data set. The percentage of notes with correctly estimated pitch and playing technique is reported to be 81.0 % and 100.0 % for five different violins [20].

Carrillo and Wanderley designed their evaluation dataset in such way, that “most part of the violin controls space” is sampled, i.e., that the dataset contains as many different parameter configurations as possible [32] (the dataset size was only given as total number of time frames).⁶ The authors of [32] could achieve very good results: rates of correctly predicted frames of 95.0 % (bow velocity), 93.5 % (bow force), 95.0 % (bow tilt), 98.0 % (string & finger position), and 97.8 % (relative bow-bridge distance) are reported. The use of low-level audio features consistently outperformed perceptual audio features for the given prediction tasks.

Krishnaswamy and Smith use a dataset of 208 “spectral patterns” for their experiments. These samples come from 52 different fretboard positions (4 strings, 13 fret positions considered), two plucking techniques (bowed and plucked), as well as two plucking points [106]. Additional note and melody recordings were made for testing the classifier. The authors found that if all 208 spectral patterns are used for classifier training, the pitch detection worked without errors. However, in this case, the detection of the remaining parameters playing techniques and plucking point was not reliable. Krishnaswamy and Smith could achieve a better classification performance if only notes from the training database with the detected note pitch are used for training. Nevertheless, no quantitative evaluation results were provided. None of the discussed evaluation sets discussed in this section were published by the authors.

3.2.2 Estimation of the Fretboard Position

Spatial Parameters

On string instruments such as the bass guitar and the guitar, notes can not only be played with different playing techniques as discussed in the previous section. Three types of *spatial parameters* need to be considered:

Fingering Different fingers of the playing hand can be used to play a certain note. Similar to the fretboard position, the choice of the *fingering* is ambiguous since any finger can be used to play a note depending on the position of the playing hand. The automatic estimation of the

⁶As shown in Section 4.2.2, the creation of the IDMT-SMT-BASS-SINGLE-TRACKS database followed the same goal with special focus on bass guitar recordings.

playing hand fingering from bass guitar recordings will not be covered in the thesis. Previously presented approaches indicate that this tasks can not be successfully tackled solely based on audio analysis. In contrast, vision-based analysis methods [65] or machine learning methods based on probabilistic models [78, 145] are applied for this task.

Plucking Point The second spatial parameter is the *plucking point*, i.e., the point on the string, where the finger or the plectrum plucks the string. As discussed in Section 2.1.2, the plucking point influences the brightness of the bass guitar sound. Algorithms for plucking point detection that were published so far are based on audio analysis either in the time domain [138, 139] or in the frequency domain [166]. In this thesis, the plucking point is considered to be constant, which usually is a fair assumption in bass guitar performances [150, 182]. Hence, the automatic detection of the plucking point will not be covered here.

Fretboard position Due to the construction of the instrument and the tuning of the strings, most notes within the instrument's pitch range can be played at multiple positions on the instrument fretboard. The *fretboard position* defines a location on the instrument neck by a string number N_S and a fret number N_F . The strings are enumerated starting with $N_S = 1$ for the string with the lowest tuning. The frets are enumerated starting with $N_F = 0$ for the open string, $N_F = 1$ for the first fret, and so on. The tablature representation discussed in Section 2.2 provides information about the string number and the fret number for each note event. In this section, solely publications about the estimation of the fretboard position from string instrument recordings are reviewed.

Selection Criteria for Fretboard Positions

Musicians choose both the fretboard position(s) and the fingering(s) to play certain note sequence based on two different types of criteria:

- **Physical criteria:** The musician's main motivation is to minimize the overall physical strain that results from finger stretching and hand movement across the neck. In the music practice, vertical play on the instrument neck is often preferred over horizontal play. Musicians prefer to stay in a fixed fretboard position as long as possible and try to make use of the whole possible pitch range, which is given there [78].⁷
- **Stylistic / tonal criteria:** The choice of fretboard position and fingering is often influenced by common practice that is associated to playing in a specific music genre or using a specific playing technique on the instrument. Since the instrument strings have a different diameter and—for the case of the acoustic guitar—different material properties, playing in different fretboard positions result in different tonal properties of the instrument sound [118].

In the context of conventional music transcription, the string number and the fret number extend the set of note parameters discussed in Section 3.1. As a consequence, these parameters can be used to improve the performance in other estimation tasks. For instance, the estimation of

⁷However, in some genres such as rock and metal, guitar players occasionally prefer horizontal movement on the higher strings.

the fretboard position can improve the transcription accuracy. If a musician plays within a fixed fretboard position, the range of possible note pitch values can be constrained due to the known string tuning of the instrument. This constraint can be used to detect incorrect fundamental frequency estimates in the transcription process. Similarly to the estimation of playing techniques discussed in Section 3.2.1, three different sensory modalities can be distinguished in the reviewed publications, ranging from pure audio-based methods to methods that exploit the visual modality or that require sensors on the instrument.

Instruments

Bass Guitar

To the best knowledge of the author, no publication besides [4] and [14] dealt with estimation of spatial parameters from bass guitar recordings. The contribution of this publication will be detailed in Chapter 4.

Guitar

Data Acquisition The methods for data acquisition from guitar recordings are similar to those used for the estimation of playing techniques (compare Section 3.2.1).

In addition to regular electro-magnetic pickups, *hexaphonic pickups* are used. In contrast to commonly used guitar pickups, these pickups allow to capture the vibration of each individual string. The analysis of the individual string signals allows to avoid the ambiguity between various fretboard positions that can be used to play the same notes on the instrument and thus reduces the problem of polyphonic transcription to a set of monophonic transcription tasks [129].

Mechanically enhanced instruments are extended by sensors that allow a very precise measuring of the spatial hand position. The main disadvantage is that, “most of these methods, while accurate, are obtrusive” to the musicians [80] since they constrain the natural hand movement on the instrument and therefore affect the musical performance.

Kerdvibulvech and Saito propose to use a *stereo-camera setup* to record guitar player performances [97]. The authors apply *colored fingertips* on the musician’s hand. This method improves the visual hand tracking by avoiding potential confusion between skin color and the fretboard color. However, in practice, the method is obtrusive to the musicians.

Spatial Parameter & Parameter Estimation Methods The first group of publications analyze monophonic guitar recordings, i.e., melodies and single notes [4, 19, 167]. Traube and Smith try to match hypothetical comb-filter like spectra to the measured spectrum in order to estimate the *fingering point* [167]. In order to classify the *string number*, Barbancho et al. [19] compute various timbre-related spectral audio features such as the inharmonicity, relative magnitude of the harmonics, or the temporal decay factor of harmonics. The classification of the *string number* is performed using machine learning algorithms based on the extracted features.

The second group of publications focus on polyphonic guitar recordings. The estimation of the guitar voicing, i.e., the fretboard position of each finger, is done using audio analysis [18, 54, 129], visual analysis [28, 29, 96, 97, 136], or by combining both modalities to a multi-modal approach [80].

For the audio analysis, O’Grady & Rickard perform music transcription on the individual output signals of the hexaphonic guitar pickup [129]. These signals are inherently associated with a particular string number, hence they directly allow to estimate the string number. In [18], Barbancho et al. use a multi-pitch estimation algorithm to compute spectral saliency values for all possible pitch values within the pitch range of the guitar. These saliency values are interpreted as observations to a Hidden Markov Model (HMM). The played chord sequence is obtained by determining the most likely state sequence in the model that explains the observed saliency values. The authors distinguish 330 different fingering configuration for the most common three-voiced and four-voiced guitar chords. In [54], Fiss and Kwasinski presented a multi-pitch estimation algorithm tailored towards the guitar. Based on a STFT, quadratic interpolation is used to detect spectral peaks and two metrics based on relationships between harmonic frequencies are used to assign the most likely peaks to potential f_0 candidates. A multi-pitch estimation algorithm is also applied in the multi-modal approach presented by Hrybyk and Kim in [80]. This way, the pitch values of all notes in guitar chords are estimated and candidates for possible chord voicings can be derived (based on the knowledge about the guitar string tuning). The voicing is then estimated by spatially tracking the musicians’ hand using computer vision techniques.

Different additional *constraints* are applied in the literature to improve the estimation of spatial parameters. For instance, the metrics used in [54] are based on specific knowledge on the instrument such as the highest possible degree of polyphony (6 simultaneous notes) as well as the maximum stretch span of the playing hand within a fixed fretboard position. Barbancho et al. use two additional models to constrain the transitions between different HMM states—a musical model, which captures the likelihood of different chord changes, and an acoustic model, which measures the physical difficulty of changing the chord fingerings [18].

Evaluation Datasets & Results Based on the estimated spatial parameter, the applied datasets contain either single note recordings [4, 19, 167], polyphonic chord recordings [96, 97, 129, 136], or mixed datasets containing both types of recordings [18, 54, 80]. None of these datasets were published.

Various evaluation measures were used so a fair performance comparison among the proposed algorithms is difficult. The spatial accuracy of the *fingering point* estimation is reported to be below 1 cm in [167] using audio-based analysis and to be between 2.49 mm and 4.93 mm in [96, 97] using visual analysis. The percentage of correct *chord voicing* detection using audio analysis is given as 87.3 % by [54] (for 18 chord voicings) and 87.0 % by [18] (for 330 chord voicings). Vision-based analysis leads to an accuracy of 94.4 % [80] (for 24 chord voicings). No overall quantitative evaluation results were reported in [19, 28, 29, 129].

Violin

Data Acquisition Audio analysis is based on the microphone signal to record acoustic violins [106] or the pickup output signal from electric violins [118]. Zhang et al. analyze video recordings of violin performances [184].

Spatial Parameter & Parameter Estimation Methods The presented audio analysis algorithms to estimate the fretboard position from violin recordings are based on rather simple features—

spectral magnitude frames [106] and output energy values of filter bank using 8 filters [118]. Krishnaswamy & Smith apply Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to reduce the high-dimensional feature space. For the string number classification, they use a simple k-Nearest Neighbors (kNN) classifier where in contrast, Zhang et al. use a Gaussian Mixture Model (GMM) classifier.

Evaluation Datasets & Results Krishnaswamy and Smith use only a small dataset and report no quantitative evaluation results [106]. The authors found that the presented method works well for pitch detection. However, it does not allow for a reliable estimation of the fretboard detection.

Maezawa et al. use a bigger dataset containing recordings of two different violins (one electric, one acoustic) with different dynamic levels. They also use an additional audio-to-score alignment in order to apply context-based error correction [118]. The highest F-measure that was achieved for string number estimation was 86.0 %.

The visual analysis algorithm presented by Zhang et al. in [184] was evaluated on a dataset of 225 s of captured video covering 504 notes played in total. The string number was detected correctly in 94.2 % of all frames.

4 Contribution

In this chapter, a novel algorithm for the automatic transcription of bass guitar tracks that was published in [4, 7, 13, 14] will be detailed. The presented work is partially based on the collaboration with Hanna Lukashevich, Christian Dittmar (Semantic Music Technologies group, Fraunhofer IDMT), and Gerald Schuller (Technische Universität Ilmenau).

A bass guitar track is understood as a sequence of consecutive acoustic note events and each event is represented by a set of parameters. In the experiments described in Chapter 5, *perfectly isolated* bass guitar tracks are analyzed, which are not superimposed by other musical instruments. The problem of source separation, i.e., the isolation of an instrument track from a polyphonic audio mixture, is not investigated in this thesis. In multi-track recording sessions, the electric bass guitar track is commonly recorded using the direct instrument output signal or using a microphone close to the loudspeaker of the bass amplification system.

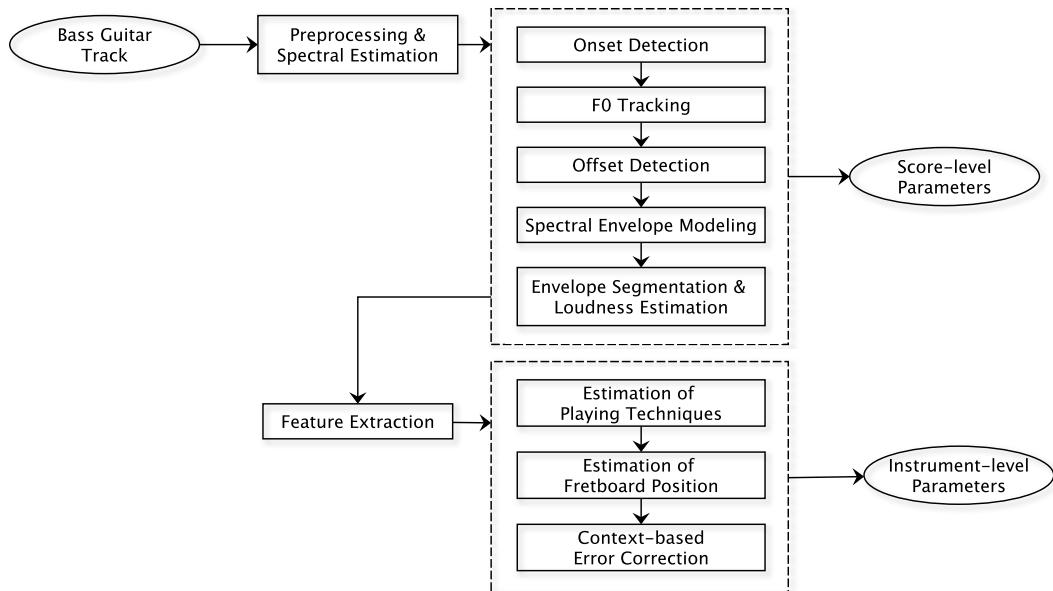


Figure 4.1: Processing flowchart and all algorithmic steps of the proposed bass guitar transcription algorithm.

Figure 4.1 illustrates the flowchart of the proposed algorithm for bass guitar transcription. It consists of two main parts, the estimation of *score-level parameters* and *instrument-level parameters*. The score-level parameters are the note pitch \mathcal{P} , onset \mathcal{O} , duration \mathcal{D} (both measured in seconds), and loudness \mathcal{L} . Based on these parameters, a bass guitar recording can be notated

as a piano roll as shown in Figure 4.2. In Part II of this thesis, a musical analysis of basslines based on score-level parameters will be discussed.

However, score-level parameters do not capture details about the expressive performance of the musician playing the instrument. To overcome this limitation, additional *instrument-level* parameters are extracted in the second part of the algorithm. First, the *plucking style* S_P and *expression style* S_E are estimated, which describe how the instrument strings were played by the musicians. Second, the *string number* N_S ¹ and *fret number* N_F are estimated, which locate where each note was played on the instrument fretboard.

As will be shown in Part III, the combined set of score-level and instrument-level note parameters can be used as input to a physical modeling algorithm in order to re-synthesize the original bass guitar track. In the following sections, all the processing steps of the transcription algorithm are detailed.

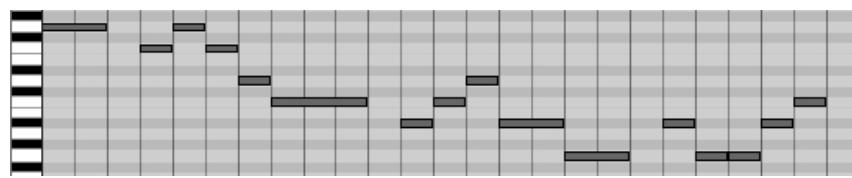


Figure 4.2: Piano roll notation of a bassline [45]. Each rectangle represents one note. The vertical position encodes the note’s pitch and the horizontal position and length of each rectangle encodes the onset and duration of the corresponding note.

4.1 Instrument-centered Bass Guitar Transcription Algorithm

4.1.1 Development Data Sets

In the following sections, two development sets *DS-1* and *DS-2* were used for optimizing different algorithm parameters. Development set *DS-1* contains 550 randomly selected isolated bass guitar notes taken from the *IDMT-SMT-BASS* dataset as introduced in Section 4.2.1 with 50 note examples for each of the 11 playing techniques discussed in Section 2.1.2. Development set *DS-2* contains all 1711 notes taken from the *IDMT-SMT-BASS* dataset that were recorded with the same electric bass guitar (Fame Baphomet 4 NTB) as the basslines in the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset as introduced in Section 4.2.2, which is used for the final evaluation of the transcription algorithm.

4.1.2 Pre-processing & Spectral Estimation

The (monaural) bass guitar track signal is initially down-sampled to a sampling frequency of $f_s \approx 5.51$ kHz using an adjusted anti-aliasing filter. Then, two different time-frequency representations are extracted. First, the *Short-time Fourier Transform* (STFT) $X(k, n)$ of the input signal

¹The string number enumerates the bass guitar strings from the lowest to the highest string. For instance, $N_S = 1$ corresponds to the lowest string.

$x \in \mathbb{R}^{N_S}$ (N_S denotes the number of samples) is computed using a blocksize of 512 samples, a hopsize of 32 samples, and a Hanning window. At the given sample rate, the temporal resolution of the spectrogram is 5.8 ms. The linear frequency index is denoted as k and the frame number (time index) is denoted as n both using zero-based indexing. Based on a zero-padding of the signal by factor 8, the resulting FFT length is $N_{\text{FFT}} = 4096$. The STFT magnitude spectrogram $M(k, n) = |X(k, n)|$ is used for spectral envelope modeling as will be described in Section 4.1.6. The frequency value that corresponds to each linear frequency index k is

$$f(k) = \frac{k}{N_{\text{FFT}}} f_s \quad (4.1)$$

with $0 \leq k \leq 2048$ and $0 \leq f(k) \leq f_s/2$.

Second, a reassigned magnitude spectrogram M_{IF} based on the *instantaneous frequency* (IF) is computed. Here, a logarithmically-spaced frequency axis with the frequency index k_{IF} is used. The corresponding frequency values are

$$f(k_{\text{IF}}) = 440 \times 2^{\frac{22+k_{\text{IF}}/10-69}{12}} \quad (4.2)$$

with $0 \leq k_{\text{IF}} \leq 780$ and $29.1 \leq f(k_{\text{IF}}) \leq f_s/2$.² The frequency axis has a resolution of 120 bins per octave, which was chosen to better capture micro-tonal variations of the fundamental frequency over time.

The instantaneous frequency $\hat{f}(k, n)$ for each time-frequency bin in the spectrogram $X(k, n)$ is estimated from the time-derivative of the local phase in the STFT spectrogram as proposed by Abe in [1].

The reassigned magnitude spectrogram $M_{\text{IF}}(k_{\text{IF}}, n)$ is computed as follows. For each time-frequency bin (k, n) in the STFT spectrogram, the magnitude value $M(k, n)$ is mapped to the corresponding time-frequency bin (k_{IF}, n) in the reassigned spectrogram M_{IF} . The frequency index k_{IF} is computed in such way that the corresponding frequency value $f(k_{\text{IF}})$ is closest to the original frequency value $f(k)$ on the linear frequency scale of the STFT spectrogram. Magnitude values mapping to the same time-frequency bins in M_{IF} are accumulated. Sinusoidal peaks tend to produce stable instantaneous frequency values in the surrounding frequency bins. Since the magnitude values of these bins are mapped towards a similar frequency position in the IF spectrogram, sinusoidal components result in sharp magnitude peaks in M_{IF} . The IF spectrogram is used for onset detection and fundamental frequency tracking as will be shown in Section 4.1.3 and Section 4.1.4.

Figure 4.3 shows excerpts from both the STFT magnitude spectrogram $M(k, n)$ (using a linearly-spaced frequency axis) and the IF magnitude spectrogram $M_{\text{IF}}(k_{\text{IF}}, n)$ (using a logarithmically-spaced frequency axis) for a bass guitar note played with *vibrato* (VI). It can be observed that the sharp peaks in the IF spectrogram are better suited for tracking of harmonic frequency values over time.

²The lower frequency limit of 29.1 Hz (MIDI pitch 22) corresponds to the note B \flat 0, which is one semitone below the lowest string of a *five-string* bass guitar. The algorithm is designed in such way that also bass guitar tracks recorded with a five-string bass guitar can be transcribed. However, in this thesis, only a four-string bass guitar with a lowest fundamental frequency of $f_0 = 41.2$ Hz (MIDI pitch 28) was used to record the evaluation datasets that will be presented in Section 4.2.

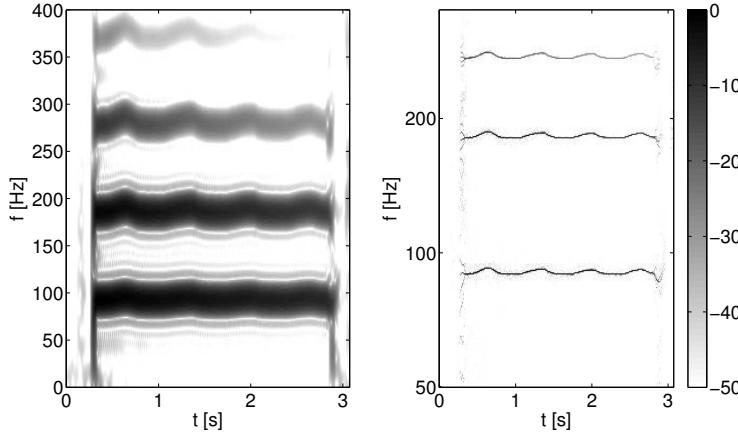


Figure 4.3: STFT magnitude spectrogram (linearly-spaced frequency axis) and IF spectrogram (logarithmically-spaced frequency axis) with dB magnitude scale for a bass guitar note played with *vibrato*. Both spectrograms are normalized for visualization purpose. It can be observed that the sharp peaks in the IF spectrogram are better suited for tracking of harmonic frequency components over time.

4.1.3 Onset Detection

In order to detect note onsets, a novel onset detection function is computed, which measures the *harmonic novelty*. The IF spectrogram M_{IF} is convolved with a kernel matrix

$$M_{\text{O}}(k, n) = [0.3, 1, 0.3]^T \times [1, 1, 1, 0, -1, -1, -1] \quad (4.3)$$

that is the time-reversed (matched) filter which has two important properties: filtering of sparse components along the frequency axis (presumably harmonic frequency components) and detection of rising magnitude slopes along the time axis (presumably note onsets). Only the central part of the convolution result

$$M_{\text{IF,O}}(k, n) = M_{\text{IF}}(k, n) * M_{\text{O}}(k, n) \quad (4.4)$$

is stored in $M_{\text{IF,O}}$ such that both $M_{\text{IF,O}}$ and M_{IF} have the same size. The onset detection function $\alpha_{\text{On}}(n)$ is computed as follows:

$$\alpha_{\text{On}}(n) = \max_k M_{\text{IF,O}}(k, n). \quad (4.5)$$

Note onset frames n_{On} are detected at all time frames n , where local maxima of $\alpha_{\text{On}}(n)$ are larger than

$$\alpha_{\text{On,min}} = 0.2 \max_n \alpha_{\text{On}}(n). \quad (4.6)$$

This empirical threshold was found using manual onset annotations for all notes in the development set *DS-1*. The threshold $\alpha_{\text{On,min}}$ leading to the maximum F-measure in onset detection was selected.

The onset time in seconds of the i -th note is computed as

$$\mathcal{O}(i) = n_{\text{On}}(i)/f_s. \quad (4.7)$$

The number of detected note onsets is denoted as N . Figure 4.4 shows the IF spectrogram $M_{\text{IF}}(f, t)$ of an excerpt of a bassline in the upper plot and the corresponding onset detection function $\alpha_{\text{On}}(t)$ in the lower plot. The detected onset positions are indicated as dashed lines. Even though the wide-band attack transients of the bass notes are not captured by the IF spectrogram, the proposed onset detection function shows clear peaks at the note onset times.

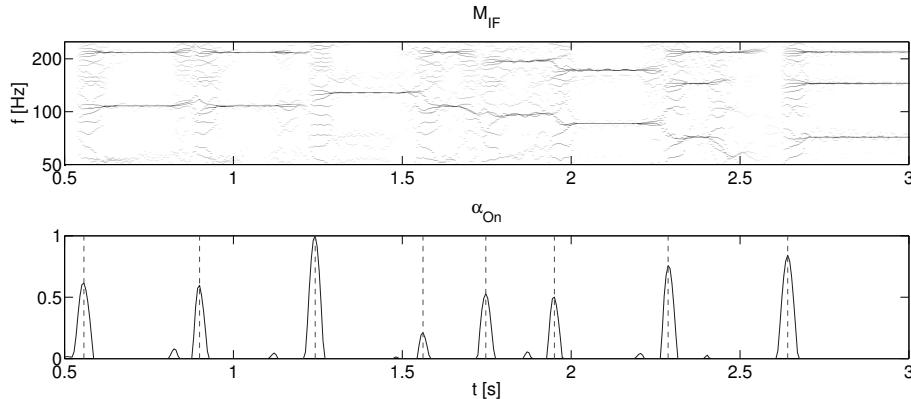


Figure 4.4: The upper figure illustrates the IF spectrogram $M_{\text{IF}}(f, t)$ of a bassline excerpt. Stable harmonic components indicate note events. The lower plot illustrates the onset detection function $\alpha_{\text{On}}(t)$ (solid line) as well as the detected onset positions t_{On} (dashed line).

4.1.4 Fundamental Frequency Tracking

After the onset detection, the note's fundamental frequency $f_0(n)$ is tracked for each note. Therefore, the spectral frames in $M_{\text{IF}}(k_{\text{IF}}, n)$ are first averaged over the first 20 % of the note's duration to obtain an *accumulated IF spectrum* vector $M_{\text{IF,acc}}(k_{\text{IF}})$. By averaging the magnitude frames only over the note's beginning period, smearing of harmonic peaks in the accumulated spectrum is prevented in case the note was played with the expression styles *bending* (BE), *vibrato* (VI), and *slides* (SL). If these styles are used, the fundamental frequency continuously changes over the duration of a note.

In order to get a pre-estimate of the note's fundamental frequency f_0 , the cross-correlation $C_{\text{M,c}}(\tau)$ is computed between $M_{\text{IF,acc}}(k_{\text{IF}})$ and a *harmonic spectral template* $c(k_{\text{IF}})$, which is shown in Figure 4.6. This template represents an idealized harmonic magnitude spectrum of a tone. The frequency positions of the spectral template peaks follow the harmonic relationship given in (2.7). The inharmonicity coefficient is set to $\beta = 0.0004$. This value was obtained by averaging over the inharmonicity coefficients of all notes from the development set *DS-1* except of those played with the *dead-note* (DN) technique.

The frequency index of the fundamental frequency $k_{\text{IF},0}$ is computed as

$$k_{\text{IF},0} = \arg \max_{\tau} C_{\text{M,c}}(\tau). \quad (4.8)$$

and the fundamental frequency f_0 is derived using (4.2). Finally, the note pitch is computed via

$$\mathcal{P}(i) = \lfloor 12 \log_2 \left(\frac{f_0}{440} \right) + 69 \rfloor. \quad (4.9)$$

by rounding to the equal temperament tuning.

Using 500 notes from the development set *DS-1* (the 50 notes played with the *dead-note* expression style were excluded since they have a percussive sound without a stable pitch), spectral templates with different numbers of harmonic peaks were compared for the task of pitch detection. Furthermore, spectral templates with peaks having unit magnitudes and spectral templates with doubled magnitude on the first two peaks were compared. As shown in Figure 4.5, a spectral template with 10 peaks and doubled magnitude on the first two peaks achieved the highest pitch detection accuracy of $A = 0.98$ on the development set. This spectral template is illustrated in Figure 4.6 and used in the transcription algorithm. Since the spectral envelopes of bass guitar notes played at different fretboard positions vary among different instruments, no further adaptation of the template peak magnitudes $c(k_{IF})$ was performed.

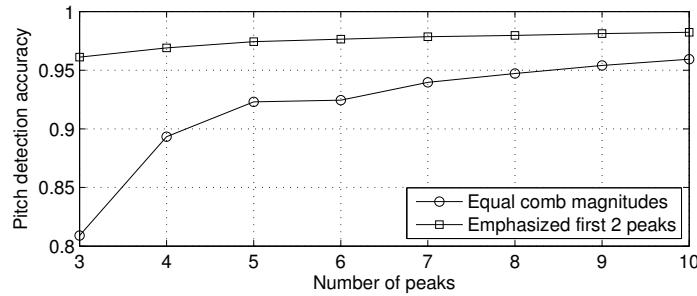


Figure 4.5: Pitch detection accuracy over 500 isolated notes (including all playing techniques but *dead-notes*). Circles indicate accuracy values obtained with spectral templates with unit magnitude, squares indicate spectral templates with doubled magnitude for the first two peaks.

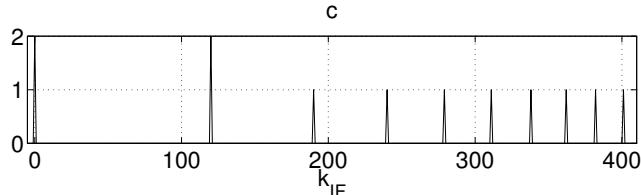


Figure 4.6: Harmonic spectral template $c(k_{IF})$ based on a logarithmic frequency axis and with doubled magnitude on the first two peaks.

The temporal f_0 -tracking is performed as follows. Starting at a start frame n_0 at approximately 10 % of the note duration, the tracking is performed forwards and backwards in time. A *continuity constraint* is applied such that in each time frame n , only those frequency indices k_{IF} around the estimated fundamental frequency index of the preceding frame are considered as f_0 candidates.

Here, a maximum deviation of plus minus 1 frame is allowed. In each frame, the cross-correlation between the spectral frame and the spectral templates is maximized to estimate $k_{IF,0}(n)$ as described before. The maximum cross-correlation value is stored for each frame as $C_{\max}(n)$. High cross-correlation values indicate a harmonic, sparse magnitude characteristic of the spectrum. Lower values indicate a percussive, wide-band characteristic.

4.1.5 Offset Detection

The *offset frame index* $n_{\text{Off}}(i)$ of the i -th note is retrieved as the first frame after $n_{\text{On}}(i)$, where $C_{\max}(n)$ remains below a relative threshold of 0.05 for at least four adjacent frames or a new note begins. Again, this threshold was determined using the *DS-1* set by maximizing the offset detection accuracy. Finally the note duration in seconds is obtained as

$$\mathcal{D}(i) = n_{\text{Off}}(i)/f_s - \mathcal{O}(i). \quad (4.10)$$

Figure 4.7 shows the results of the f_0 -tracking and detection of onset and offset for a bass guitar note played with *vibrato* expression style.

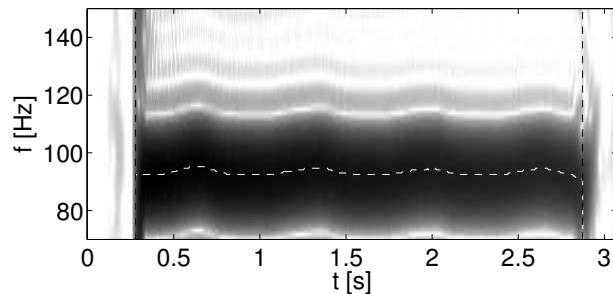


Figure 4.7: Bass note played with *vibrato* expression style (same as in Figure 4.3). The tracked fundamental frequency $f_0(t)$ is indicated by a white dashed line. The onset and offset times are shown as vertical black dotted lines. It can be observed that the corresponding excerpt of the STFT magnitude spectrogram $M(k, n)$ shown in the background exhibits strong spectral leakage, which prevents a precise tracking of the fundamental frequency over time.

4.1.6 Spectral Envelope Modeling

As a next step, the harmonic magnitudes $a_h(n)$ must be estimated to capture the timbral properties of bass guitar notes. For this purpose, partial tracking algorithms based on detecting spectral peaks and tracking them over time are most-often used in the literature (see for instance [71]). Due to the spectral leakage in lower frequency ranges, this approach is not promising for bass guitar notes. Therefore, in this thesis, the spectral envelope of each note is modeled using a *parametric approach*. This step aims at describing the fundamental frequency and the harmonics in the STFT magnitude spectrogram $M(k, n)$ using a set of time-varying frequency and magnitude values, which will be used for feature extraction afterwards.

Two simplifications are used. First, wide-band noise-like signal components such as the attack transients are not considered in the model. As will be shown in Section 4.1.8, the wide-band characteristic is nevertheless measured by some of the applied audio features. Second, the inharmonicity coefficient β is assumed to be constant for each individual note event.

Estimation of the Inharmonicity Coefficient

For each note event, the inharmonicity coefficient β is estimated once in the frame n_0 in the beginning of the note decay part (compare Section 4.1.4). A grid search for β with 100 equidistant grid points within the range $[0, 0.001]$ is performed. For each candidate $\hat{\beta}$, the corresponding harmonic frequencies $\hat{f}_{h,\hat{\beta}}$ of the first $N_H = 10$ harmonics (including the fundamental frequency) are computed as

$$\hat{f}_{h,\hat{\beta}} = (h + 1)f_0\sqrt{1 + \hat{\beta}(h + 1)^2} \quad (4.11)$$

with $0 \leq h \leq N_H$. The estimation of the fundamental frequency f_0 was explained in Section 4.1.4.

In order to estimate the inharmonicity coefficient, a likelihood measure $L(\hat{\beta})$ is computed for each candidate $\hat{\beta}$ by summing up the magnitude values $\hat{M}_{h,\hat{\beta}}$ at the corresponding harmonic frequencies as

$$L(\hat{\beta}) = \sum_{h=0}^{N_H} \hat{M}_{h,\hat{\beta}}. \quad (4.12)$$

The magnitude values $M_{h,\hat{\beta}}$ are computed using linear interpolation from the magnitude values $M(k, n_0)$ of the STFT magnitude spectrogram at the frequencies $f(k)$ (compare (4.1)).

Finally, the inharmonicity coefficient β is estimated as

$$\beta = \arg \max_{\hat{\beta}} L(\hat{\beta}). \quad (4.13)$$

Estimation of the Harmonic Magnitudes

In the next step, the harmonic magnitudes $a_h(n)$ are estimated using the Expectation-Maximization (EM) algorithm [124]. This algorithm implements an iterative maximum-likelihood (ML) estimation of parameters of a probability distribution based on a limited set of observations.³

In a given frame n , the (observed) magnitude spectrum $M(k, n)$ is first normalized to unit sum in order to be interpretable as probability density:

$$p_n(k) = \frac{M(k, n)}{\sum_k M(k, n)}. \quad (4.14)$$

This probability density is modeled as a sum of magnitude-scaled atom functions, which represent the spectral peaks at the harmonic frequencies. The atom function $W(k)$ is the Discrete Fourier transform of the Hanning window $w(n)$, which is applied in the time domain to compute the STFT spectrogram $X(k, n)$. As shown in Figure 4.8, the atom function $W(k)$ is truncated outside its first two side-lobes and normalized to unit magnitude.

³As an alternative, the harmonic magnitudes could be solved in close form via least-squares optimization. However, this approach was not investigated in this thesis.

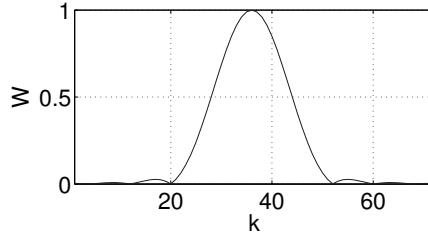


Figure 4.8: Atom function $W(k)$ used for spectral envelope modeling. The STFT blocksize is 512. The function was truncated outside its first two side-lobes and normalized to unit magnitude.

In each time frame n and in each iteration of the EM-algorithm, the following steps are performed⁴ based on the current parameters f_0 , β , and a_h . First, the harmonic frequencies f_h are computed as

$$f_h = (h+1)f_0(n_0)\sqrt{1+\beta(h+1)^2}. \quad (4.15)$$

Then, for each harmonic, the corresponding marginal density $p_h(k)$ is computed (E-step) as

$$p_h(k) = a_h W(k - k_h). \quad (4.16)$$

The frequency bin k_h is the closest bin to the harmonic frequency f_h :

$$k_h = \arg \min_k |f(k) - f_h| \quad (4.17)$$

Then, the marginal densities are normalized to unit sum as

$$p_h(k) \leftarrow \frac{p_h(k)}{\sum_h p_h(k)}. \quad (4.18)$$

In the last step, the harmonic magnitude weights are updated (M-step):

$$a_h \leftarrow \sum_k p_h(k) \cdot p(k) \quad (4.19)$$

After the last iteration, the harmonic magnitudes a_h are rescaled via

$$a_h \leftarrow a_h \cdot \sum_k M(k, n) \quad (4.20)$$

to match the observed magnitude spectrogram (compare (4.14)).

Starting in the frame n_0 , a frame-wise estimation of $a_h(n)$ is performed by stepping forward and backward in time, similarly to the f_0 tracking as described in Section 4.1.4. Since the spectral envelope in the decay part of a harmonic note has a continuous shape, the estimates of $a_h(n)$ can serve as a good initialization in the adjacent frames. In each frame, the EM algorithm is initialized with the optimal parameter set obtained in the previous frame. Five iterations are

⁴The index n is omitted in the following section for better readability.

used in the starting frame n_0 and two iterations are used in each of the remaining frames. After the envelope modeling, each note is described by the set of envelope parameters $[a_h(n), \beta, f_0(n)]$, which are used for feature extraction as will be described in the next section.

In Figure 4.9, the magnitude spectrogram $M(f, t)$ of a bass guitar note played with the *vibrato* expression style is shown in the left figure. The approximated magnitude spectrogram using the estimated parameters $a_h(n)$, β , and $f_0(n)$ is shown in the right figure. The magnitude envelopes of the harmonics (in particular the fourth harmonic) show the typical phenomena of *string beating*, which is well captured by the modeling procedure. Due to the discussed limitations of the modeling approach, the attack transients, occurring at the beginning of the note, are not properly modeled.

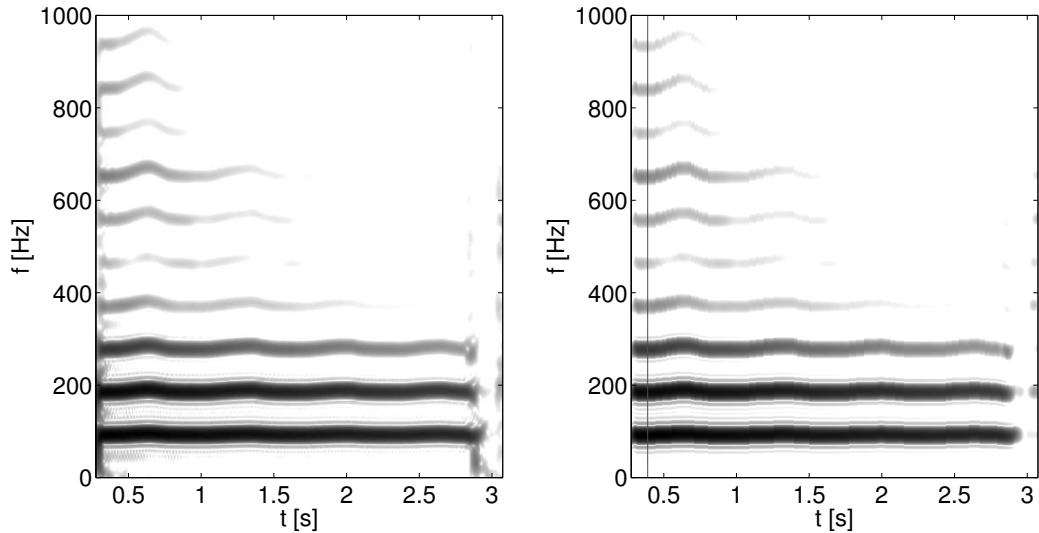


Figure 4.9: STFT spectrogram $M(f, t)$ of a bass guitar note played with the *vibrato* expression style: original (left) and modeled (right). The start frame used for the optimization is shown as vertical black line in the right figure. It can be observed that except for the onset segment, the time-varying frequency and magnitude trajectories of the different harmonics are well captured by the proposed spectral envelope modeling algorithm.

4.1.7 Envelope Segmentation & Loudness Estimation

Based on the harmonic envelopes $a_h(n)$, the *aggregated magnitude envelope* $a(n)$ is computed as

$$a(n) = \sum_h a_h(n). \quad (4.21)$$

As shown before in Figure 2.11, a simplified two-stage model is used to segment each note into an *attack part*, which is characterized by a rapidly increasing magnitude envelope and a *decay part*, which is characterized by approximately exponentially decaying magnitude values. Hence, the frame

$$n_{\text{Peak}} = \arg \max_n a(n) \quad (4.22)$$

is estimated as the boundary between the note's attack and decay part. The *loudness* of the i -th note is computed from the envelope peak magnitude in dB as

$$\mathcal{L}(i) = 20 \log_{10} a(n_{\text{Peak}}). \quad (4.23)$$

4.1.8 Feature Extraction

In the following sections, various audio features (denoted by χ) from different feature categories will be described. In the proposed transcription algorithm, all features are extracted for each note event. It will be shown in Section 4.1.9 and Section 4.1.10 how the plucking style, the expression style, and the string number are automatically classified using these features.

Harmonic Magnitudes & Frequencies

The first group of features describes the shape of the *aggregated magnitude envelope* $a(n)$ as computed in (4.21). Following the two-stage envelope model, $a(n)$ is modeled as an increasing linear function in the attack part as

$$a(n) \approx \alpha_1 n + \alpha_0 \quad (4.24)$$

for $n_{\text{On}} \leq n \leq n_{\text{Peak}}$

and as a decreasing exponential function in the decay part as

$$a(n) \approx a(n_{\text{Peak}}) e^{-\beta_1 n} \quad (4.25)$$

for $n_{\text{Peak}} \leq n \leq n_{\text{off}}$.

Linear regression is used to estimate the function parameters. The two coefficients α_1 and β_1 are used as features $\chi_{\text{slope,attack}}$ and $\chi_{\text{slope,decay}}$.

The second group of features are extracted over the harmonic magnitudes $a_h(n_{\text{Peak}})$ and frequencies $f_h(n_{\text{Peak}})$ in the note's peak frame. As previously detailed in [4], the *relative harmonic magnitudes*

$$\chi_{\text{a,rel}}(h) = a_h(n_{\text{Peak}})/a_0(n_{\text{Peak}}) \text{ with } h \geq 1 \quad (4.26)$$

and the *inharmonicity coefficient* $\chi_\beta = \beta(n_{\text{Peak}})$ (compare Section 4.1.6) are used as features. By using linear regression, the harmonic magnitudes are interpolated as a decaying linear function over the harmonic index h as

$$a_h \approx \gamma_1 h + \gamma_0. \quad (4.27)$$

The linear slope $\chi_{\text{a,slope}} = \gamma_1$ is used as feature to characterize the spectral magnitude decay over frequency. Using the estimates of the fundamental frequency $f_0(n_{\text{Peak}})$ and of the inharmonicity coefficient $\beta(n_{\text{Peak}})$, the corresponding hypothetical harmonic frequencies $\hat{f}_h(n_{\text{Peak}})$ are computed using (2.7).

The *normalized frequency deviations*

$$\chi_{\Delta,f}(h) = \frac{\hat{f}_h(n_{\text{Peak}}) - f_h(n_{\text{Peak}})}{f_h(n_{\text{Peak}})} \text{ for } h \geq 1 \quad (4.28)$$

between the hypothetical harmonic frequencies and the measured harmonic frequencies $f_h(n_{\text{Peak}})$ in the IF spectrogram have shown to be suitable features for the task of string number classification in [4]. Additional features are obtained by computing the statistical measures minimum, maximum, mean, median, variance, skewness, and kurtosis over the two vectors $\chi_{a,\text{rel}}$ and $\chi_{\Delta,f}$.

Aggregated Timbre Features

In order to characterize the magnitude spectrogram of a note, the frame-wise features *tristimulus*

$$\chi_{\text{harm,tri},1}(n) = \frac{a_0(n)}{\sum_h a_h(n)} \quad (4.29)$$

$$\chi_{\text{harm,tri},2}(n) = \frac{\sum_{h=2}^4 a_h(n)}{\sum_h a_h(n)} \quad (4.30)$$

$$\chi_{\text{harm,tri},3}(n) = \frac{\sum_{h=5}^{10} a_h(n)}{\sum_h a_h(n)} \quad (4.31)$$

$$(4.32)$$

and *spectral irregularity*

$$\chi_{\text{harm,irr}}(n) = \frac{\sum_{h=1}^{10} (a_h - a_{h-1})^2}{\sum_{h=0}^{10} (a_h)^2} \quad (4.33)$$

are computed from the harmonic magnitudes. The features *spectral centroid*

$$\chi_{\text{spec,cent}}(n) = \frac{\sum_k k M(k, n)}{\sum_k M(k, n)}, \quad (4.34)$$

spectral crest factor

$$\chi_{\text{spec,crest}}(n) = \frac{\max_k M(k, n)}{\frac{1}{k_{\max}} \sum_k M(k, n)}, \quad (4.35)$$

differentiated spectral crest-factor

$$\chi_{\text{spec,diff,crest}}(n) = \chi_{\text{spec,crest}}(n) - \chi_{\text{spec,crest}}(n-1), \quad (4.36)$$

spectral roll-off $\chi_{\text{spec,roll}}(n)$, *spectral slope* $\chi_{\text{spec,slope}}(n)$, and *spectral spread* $\chi_{\text{spec,spread}}(n)$ are computed from the magnitude spectrogram as proposed in [62] and [89]. All these frame-wise features are aggregated over the attack and the decay part using the statistical measures explained in Section 4.1.8. As shown in [13], aggregated timbre features perform very well for the classification of different plucking and expression styles.

Instrument Noise

Notes that are played with the plucking styles *slap-thumb* (ST) and *slap-pluck* (SP) as well as the expression style *dead-note* (DN) have a percussive, wide-band characteristic in the attack part of the magnitude spectrogram as shown in Section 2.1.2. In order to compute a feature that measures the presence of wide-band noise between the harmonics in a given frame \tilde{n} , the

harmonic peaks are removed from a spectral frame $M(k, \tilde{n})$ using a spectral template tuned to the fundamental frequency $f_0(\tilde{n})$ and the inharmonicity coefficient $\beta(\tilde{n})$. The remaining spectrogram is stored in $X_N(k, \tilde{n})$. The feature is computed as the energy ratio between the filtered and the original magnitude spectrogram, averaged over the note attack part as

$$\chi_{\text{Noise}} = \frac{1}{n_{\text{Peak}} - n_{\text{On}}} \sum_{n=n_{\text{On}}}^{n_{\text{Peak}}} \frac{\sum_k M_N(k, n)}{\sum_k M(k, n)}. \quad (4.37)$$

Subharmonic Energy

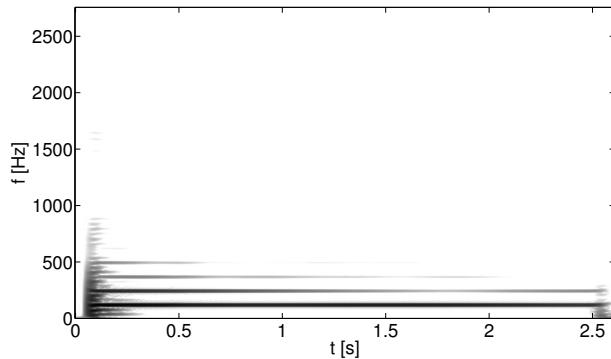


Figure 4.10: STFT spectrogram $M(f, t)$ of a note played with the *harmonics* expression style on the low E-string. It can be observed that note note's fundamental frequency of 123.6 Hz corresponds to the third harmonic of the open string fundamental frequency of 41.2 Hz.

Another group of features characterizes *subharmonic components* in the spectrogram. Figure 4.10 illustrates a note that was played on the low E string ($f_{0,E} = 41.2$ Hz) with the *harmonics* (HA) expression style. The fundamental frequency of the open string can be seen in the beginning of the note decay part (between 0.1 s and 0.25 s). However, due to the string damping at a third of the string length, only the third vibration mode and its octaves remain audible after $t = 0.25$ s. Hence, the perceived fundamental frequency of the note is three times higher ($f_0 = 3f_{0,E} = 123.6$ Hz) than the fundamental frequency of the open string. The harmonics of $f_{0,E}$ can be interpreted as subharmonics in relation to f_0 .

In order to distinguish notes played with the *harmonics* expression style from notes played with other expression styles with the same perceived fundamental frequency, several features are computed to characterize the spectral energy at *subharmonic frequency positions*. In order to extract *likelihood-values* for different vibration modes that indicate the presence of subharmonics, harmonic spectral templates are tuned to different virtual fundamental frequencies $f_{0,\text{virtual}}(m) = f_0/m$ with $m \in [2, 7]$.⁵ At the same time, these spectral templates are modified in such way that they have no spectral peaks at multiples of the “real” fundamental frequency f_0 . The likelihood-value $\chi_{\text{sub},m}$ for $f_{0,\text{virtual}}(m)$ is computed by multiplying the spectrum with the modified spectral templates and using the energy sum ratio as feature similar to (4.37).

⁵The highest vibration mode played with the *harmonics* style in the dataset is $m = 7$.

The third group of features are *string-likeness values*. If harmonics are played on a particular string, then the energy of the harmonic peaks is likely to be filtered out by the spectral template that is tuned to the open string fundamental frequency. Hence, for all four strings of the bass guitar, a likelihood value χ_{string} is computed using spectral templates that are tuned to the open string fundamental frequencies as described before.

Fundamental Frequency Modulation

The three expression styles *vibrato*, *bending*, and *slide* as explained in Section 2.1.2 involve a characteristic modulation of the fundamental frequency over time. Examples for typical fundamental frequency tracks $f_0(n)$ are given for all three styles in Figure 4.11.

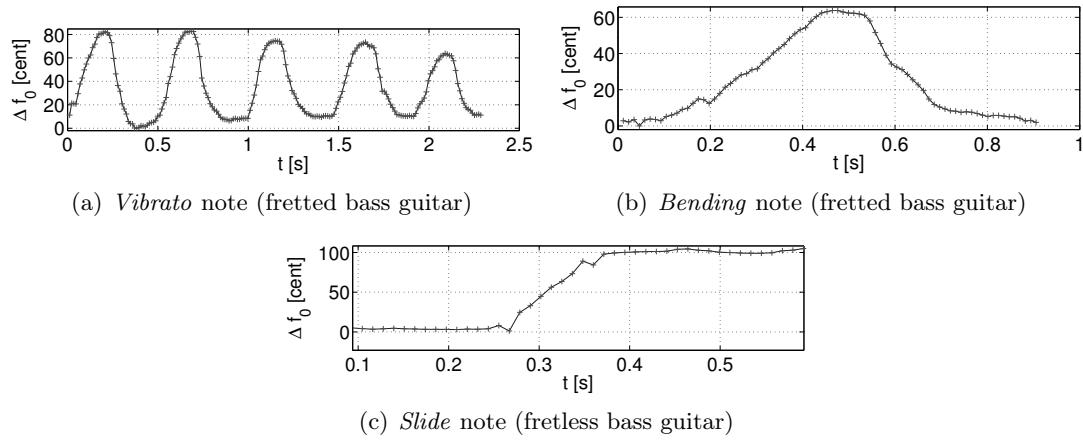


Figure 4.11: Characteristic fundamental frequency tracks for the expression styles *vibrato*, *bending*, and *slide* [7]. Frequency is given as relative frequency difference to the lowest f_0 -value in cent with 100 cents corresponding to one semitone.

Several features are computed to characterize the f_0 track. First, the normalized autocorrelation function $c_{f_0}(\tau)$ is computed over $f_0(n)$ with $c_{f_0}(0) = 1$. Figure 4.12 illustrates $c_{f_0}(\tau)$ for the vibrato note shown in Figure 4.11(a). The *mean modulation frequency* is estimated as

$$\chi_{f,\text{mod}} = 1/\tau_{\max} \quad (4.38)$$

with τ_{\max} being the first non-zero lag of a local maximum in $c_{f_0}(\tau)$ ($\tau > 0$).

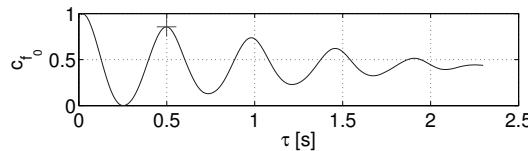


Figure 4.12: Normalized autocorrelation function over $f_0(n)$ [7]. The mean modulation frequency $\chi_{f,\text{mod}}$ is derived from the first non-zero maximum (indicated by a cross).

A dominance-measure

$$\chi_{\text{mod,dom}} = c_{f_0}(\tau_{\max}) \quad (4.39)$$

is computed as feature to measure the *intensity of the modulation*. In order to discriminate between the modulation techniques, the *number of modulation quarter-periods* is estimated as $\chi_{\text{mod,periods}}$. As shown in Figure 4.11, the *slide* technique commonly has one, the *bending* technique has two, and the *vibrato* technique has more than 2 modulation quarter periods.

The *modulation lift* is measured in cent relative to the lowest fundamental frequency value as

$$\chi_{\text{mod,lift}} [\text{cent}] = 1200 \log_2 \left(\frac{\max f_0(n)}{\min f_0(n)} \right) \quad (4.40)$$

Finally, the *overall pitch progression* $\chi_{\text{mod,prog}}$ is measured as the difference between the average fundamental frequency in the last 30 % and the first 30 % of the note frames, also given in cent.

Finally, all extracted features are concatenated to a feature vector $\chi \in \mathbb{R}^{210}$ that represents each note event.

4.1.9 Estimation of Plucking Style & Expression Style

Three statistical models are trained using the approx. 1700 notes from development set *DS-2*. Depending on the classification task, the note parameters plucking style \mathcal{S}_P , expression style \mathcal{S}_E , or string number N_S are used as class labels for the classifier training. In each training procedure, the feature selection method IRMFSP as explained in Section 2.3.2 is applied to reduce the dimensionality of the feature space to $N_D = 50$. Then, a Support Vector Machine (SVM) classifier with a radial basis function (RBF) kernel is trained for each of the three classification tasks. If a bassline is transcribed, the plucking style, expression style, and string number is derived for each note by maximizing the class probability values obtained from the trained classifiers based on the extracted feature vector.

4.1.10 Estimation of String Number & Fret Number

Depending on the expression style \mathcal{S}_E , three cases are distinguished for the estimation of the string number N_S and fret number N_F .

For *dead-notes*, the fret number is not considered to be relevant and the string number is either randomly set for single notes or set to the string number of the closest note played with one of the expression styles *normal*, *vibrato*, *bending*, or *slide*. This procedure results in tablature notations that are easier to play for the bass player.

For notes played with the *harmonics* style, the string number is obtained by maximizing the aforementioned string likelihood values χ_{string} and mode number by maximizing the mode likelihood values $\chi_{\text{sub,m}}$ as explained in Section 4.1.8. Since *harmonics* with a given mode can be played on multiple fret positions, the fret number is set to be close to the fret numbers of previous notes based on the estimated mode \hat{m} .

For all other expression styles, the following steps are performed. First, the string number N_S is classified as explained in the previous section. Subsequently, the fret number is computed based on the note pitch \mathcal{P} and the pitch of the string that was classified $\mathcal{P}_{\text{String}}(N_S)$ as

$$N_F = \mathcal{P} - \mathcal{P}_{\text{String}}(N_S). \quad (4.41)$$

Assuming the standard tuning of the bass guitar, $\mathcal{P}_{\text{String}} = [28, 33, 38, 43]$.

4.1.11 Context-based Error Correction

Two constraints are used to improve the estimation of the instrument-related parameters. First, as shown in [4], the probability values of those strings where the note pitch \mathcal{P} cannot be played are set to zero before the string number is classified. Second, all plucking styles are partitioned into the four classes *finger-style*, *muted*, *picked*, as well as *slap*, which combines *slap-pluck* and *slap-thumb*. It is assumed that only one of the four plucking style classes is used within one bassline. This is a reasonable assumption for most basslines in music practice [182]. Thus, the class probabilities of all plucking style classes are accumulated over all notes of a given bass guitar track in order to determine the most likely plucking style class. Then, all plucking styles \mathcal{S}_P are set according to this class. For the *slap* class, the plucking style is chosen from *slap-pluck* or *slap-thumb* by maximizing the class probability.

4.2 Data Sets

4.2.1 IDMT-SMT-BASS

In 2010, the *IDMT-SMT-BASS* dataset was made available to the research community in [13] and can be accessed online at [2]. The dataset is intended as a public evaluation benchmark for the tasks of playing technique estimation, fretboard position estimation, as well as transcription of bass guitar notes.

The dataset contains isolated note recordings played on three different 4-string electric bass guitars, each with 3 different pickup settings⁶. The notes cover the common pitch range of a 4-string bass guitar from E₁ (41.2 Hz) to G₂ (196.0 Hz). The overall duration of the audio material is approximately 3.6 hours and the dataset consists of around 4300 WAV files.

All 5 plucking styles (FS, MU, PK, ST, and SP) and 6 expression styles (NO, DN, HA, BE, VI, and SL) as well as the 6 expression style sub-classes (BEQ, BES, VIF, VIS, SLD, and SLU) as listed in Table 2.1 are covered by the recorded notes. In real-world recordings, various combinations of playing techniques can frequently be observed. In this data set, the plucking style FS was used when the expression style was varied and the expression style NO was used when the plucking style was varied, respectively.

4.2.2 IDMT-SMT-BASS-SINGLE-TRACKS

In 2013, the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset was published in [9] and made available online at [3]. It contains 17 bass guitar recordings of realistic basslines from the music genres blues, funk, rock, bossa nova, forró, and hip hop. All playing techniques discussed in Section 2.1.2 are represented within the contained basslines. The bass notes cover all four strings of the bass guitar and contain 2, 4, or 8 repetitions of bass patterns of 1, 2, or 4 measures of length with no or minor variations. The basslines are intended to cover most parts of the instrument’s “control

⁶The term pickup setting denotes a specific loudness ratio between the output signals of the two electro-magnetic pickups at the instrument as explained in Section 2.1.1.

Table 4.1: Overview of the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset. The absolute number and percentage of notes from different plucking style, expression style, and string number classes in the dataset are given.

Plucking Styles	FS	MU	PK	ST	SP	Σ
Number of notes	395	216	138	138	56	934
%	41.89	22.91	14.63	14.63	5.94	100
Expression Styles	NO	VI	BE	SL	DN	HA
Number of notes	822	31	8	20	30	32
%	87.17	3.29	0.85	2.12	3.18	3.39
String Number	E	A	D	G		Σ
Number of notes	236	313	256	138		934
%	25.03	33.19	27.15	14.63		100

space” [32]. These properties qualify the dataset to be used for the evaluation of the transcription of the bassline, the retrieval of repeating bass patterns, estimation of playing techniques (plucking styles and expression styles), estimation of the fretboard position, as well as evaluation of the instrument-based audio codec presented in Part III.

Table 4.1 summarizes the composition of the *IDMT-SMT-BASS-SINGE-TRACKS* dataset. It can be seen that no equal distribution concerning the plucking style, expression style, and string number classes can be achieved. However—to the best knowledge of the author—the distribution of plucking styles, expression styles, and string numbers is in accordance to commonly played basslines in music practice.

5 Evaluation

5.1 Estimation of Plucking Styles & Expression Styles from Bass Guitar Notes

Motivation & Goals

The goal of this experiment was to evaluate the discriminative power of the feature set explained in Section 4.1.8 for the classification of plucking and expression styles. The classification task was simplified by focusing on isolated bass guitar note recordings with pitch ground truth annotations first. Hence, estimation errors of pitch, onset, and offset are ruled out in this experiment as potential sources of error for the subsequent feature extraction and classification. The classification of the 5 plucking styles and 5 expression styles given in Table 2.1 (except the *slide* expression style) was performed separately. The results of this study were published in [13].

Dataset

Approximately 4300 notes from the *IDMT-SMT-BASS* dataset presented in Section 4.2.1 were used.

Experimental Procedure

The feature selection algorithm *Inertia Ratio Maximization using Feature Space Projection* (IRMFSP) and the feature space transformation algorithms *Linear Discriminant Analysis* (LDA) and *Generalized Discriminant Analysis* (GDA) were investigated to reduce the dimensionality of the feature space prior to the classification. Furthermore, the performance of the classification algorithms *Support Vector Machines* (SVM) (using the *Radial Basis Function* kernel), *Gaussian Mixture Models* (GMM), *Naive Bayes* (NB), and *k-Nearest Neighbors* (kNN) were compared for the given task. More details about these algorithms can be found in Section 2.3. The mean class accuracy was computed based on a 10-fold stratified cross-validation.

Baseline Experiment

Initially, a baseline experiment was performed as follows. Instead of using the proposed feature set for the classification of plucking and expression styles, Mel-Frequency Cepstral Coefficients (MFCC) [62] were chosen as audio features since they are widely applied for comparable MIR classification tasks such as instrument recognition [71]. Gaussian Mixture Models (GMM) were used as classifiers with a varying number of $N_{\text{Gauss}} \in \{1, 2, 3, 5, 10\}$ Gaussians. The best results are given in Table 5.1.

Table 5.1: Experimental results for the classification of plucking styles and expression styles from isolated note recordings using MFCC features.

Experiment	Number of classes	Highest mean class accuracy
Plucking style classification	5	65.7 % ($N_{\text{Gauss}} = 2$)
Expression style classification	5	67.3 % ($N_{\text{Gauss}} = 3$)

Results & Summary

Table 5.2 shows the classification results for both classification tasks using the proposed feature set with and without prior feature selection and feature space transformation. The mean class accuracy values are given, the standard deviation values can be found in [13]. The parameters of the chosen algorithms, i.e., the number of Gaussians (GMM), the number of nearest neighbors (kNN), the number of selected features (IRMFSP) and γ (GDA), are given in brackets.

The highest mean class accuracy values of 93.3 % and 95.6 % were achieved for the classification of the plucking and expression style, respectively. The combination of IRMFSP for feature selection and GDA for feature space transformation lead to the highest classification scores for most of the classifiers.

5.2 Estimation of Expression Styles Subclasses from Bass Guitar Notes

Motivation & Goals

In this experiment, the classification of the expression styles shown in Table 2.1 was performed on a *class-level* using 4 classes and on a *sub-class level* using 7 classes. Again, isolated bass guitar notes with pitch ground truth annotations were used. The results of this study were published in [7].

Dataset

A dataset containing 156 isolated note recordings for each of the 6 expression style sub-classes *slow vibrato* (VIS), *fast vibrato* (VIF), *quarter-tone bending* (BEQ), *half-tone bending* (BEH), *slide-up* (SLU) and *slide-down* (SLD) was utilized. The expression style *normal* was used as seventh class. Hence, 156 samples of that class were randomly added from the *IDMT-SMT-BASS*. The recordings for the two sub-classes *slide-up* and *slide-down* were recorded on a fretless bass guitar, all others were recorded on a fretted bass guitar. For the classification using the 4 expression style classes *normal* (NO), *vibrato* (VI), *bending* (BE), and *slide* (SL), the corresponding sub-class samples were merged accordingly.

Table 5.2: Mean class accuracy values for plucking and expression style classification from isolated note recordings for different classifiers without and with feature selection (FS) / feature space transformation (FST). The configurations with the best performance are highlighted with bold print.

Plucking Style Classification			
Classifier	Without FS / FST	With FS / FST	Best Configuration
SVM	90.75 %	92.77 %	IRMFSP(80) + GDA (10^{-7})
GMM(2)	70.04 %	92.30 %	IRMFSP(80) + GDA (10^{-7})
GMM(3)	72.61 %	92.32 %	IRMFSP(80) + GDA (10^{-7})
GMM(5)	75.92 %	93.25 %	IRMFSP(100) + GDA(10^{-7})
GMM(10)	79.22 %	92.51 %	IRMFSP(80) + GDA (10^{-7})
NB	66.43 %	91.63 %	IRMFSP(80) + GDA(10^{-7})
kNN(1)	79.62 %	92.34 %	IRMFSP(100) + GDA (10^{-7})
kNN(5)	82.58 %	92.96 %	IRMFSP(80) + GDA (10^{-7})
kNN(10)	82.61 %	92.80 %	IRMFSP(100) + GDA (10^{-7})

Expression Style Classification			
Classifier	Without FS / FST	With FS / FST	Best Configuration
SVM	93.77 %	94.96 %	IRMFSP(100) + GDA(10^{-7})
GMM(2)	77.63 %	95.13 %	No FS + GDA(10^{-15})
GMM(3)	75.09 %	94.78 %	No FS + GDA(10^{-9})
GMM(5)	77.62 %	95.28 %	IRMFSP(100) + GDA(10^{-7})
GMM(10)	82.45 %	95.61 %	IRMFSP (100) + GDA(10^{-7})
NB	72.61 %	95.28 %	IRMFSP(100) + GDA(10^{-7})
kNN(1)	87.35 %	94.79 %	IRMFSP(100) + GDA(10^{-7})
kNN(5)	90.08 %	95.12 %	IRMFSP(100) + GDA(10^{-7})
kNN(10)	90.45 %	94.97 %	IRMFSP(100) + GDA(10^{-7})

Experimental Procedure

In this experiment, a 12-dimensional feature vector was extracted using features related to the fundamental frequency modulation [7]. Due to the small number of features, SVM classifiers with a RBF kernel were used without any prior feature selection or feature space transformation. The evaluation was based on a 10-fold stratified cross validation.

Results & Summary

Table 5.3 and Table 5.4 illustrate the confusion matrices for both classification tasks. The mean class accuracy for the classification of expression styles on a class level is 85.7 %. The results for the class *slide* are almost 100 % and the remaining three classes achieve about 80 % of accuracy. Presumably, the pitch progression feature introduced in Section 4.1.8 allows almost perfect discrimination between the *slide* class and all other classes.

The mean class accuracy for the classification of expression styles on a sub-class level is 81.7 %.

The confusion matrix shown in Table 5.4 shows similar misclassification as in Table 5.3. Due to similar pitch trajectories, misclassification between the slow vibrato class and the semi-tone bending class appear naturally.

Table 5.3: Confusion matrix for expression style classification on class level. All values are given in percent.

	BE	VI	SL	NO
BE	82.9	9.6	4.5	3.0
VI	8.9	79.9	5.0	6.2
SL	0.2	0.6	98.7	0.5
NO	5.8	8.7	4.3	81.2

Table 5.4: Confusion matrix for expression style classification on sub-class level. All values are given in percent.

	BEQ	BEH	VIF	VIS	SLD	SLU	NO
BEQ	76.9	0.8	9.5	6.0	1.6	0	5.2
BEH	7.6	74.6	0	6.9	5.3	0.8	4.8
VIF	5.3	0.7	76.2	5.1	5.0	0	7.6
VIS	9.3	1.6	8	70.5	5.5	0.8	4.2
SLD	0	1.1	0	0.6	97.6	0.6	0.1
SLU	0	0.4	0	0.9	1.3	96.7	0.8
NO	2.3	3.9	8.9	2.4	1.8	1.0	79.7

5.3 Estimation of String Number from Bass Guitar Notes

Motivation & Goals

This experiment was performed to evaluate the discriminative power of the proposed feature set for string number classification. Similar to the previous experiments, isolated note recordings were used. The influence of various factors such as the assignment of different bass guitars towards the training and test set on the overall classification performance was investigated. In [4], the results of this experiment were published for both electric guitar and electric bass guitar. In this section, only the results for the bass guitar recordings will be detailed. The results for the electric guitar having 6 string classes showed very similar tendencies concerning the classification performance.

Dataset

The dataset introduced in [161] was used in this experiment—it contains isolated bass guitar and electric guitar notes both unprocessed and processed with different digital audio effects such as distortion, chorus, or delay. All 1034 unprocessed bass and guitar recordings were used as dataset in the experiment. The samples were recorded using two different bass guitars and two

different electric guitars, each played with two different plucking styles (*picked* and *finger-style*) and recorded with two different pickup settings (pickup close to the instrument neck or body).

Experimental Procedure

The main target of this experiment was to identify, how the following factors affect the performance of the string classification algorithm:

- the separation of the training and test set according to the applied instrument, playing technique, and pickup setting,
- the use of a plausibility filter (compare Section 4.1.11),
- and the use of a aggregation (voting-scheme) over multiple frame-wise classification results (see [4] for details).

The different experimental conditions are illustrated in Table 5.7. The columns “Separated instruments”, “Separated playing techniques”, and “Separated pickup setting” indicate which criteria were applied to separate the samples from training and test set in each configuration. The fifth and sixth column indicate whether the plausibility filter and the frame result aggregation were applied.

LDA was used for feature space transformation. The number of remaining feature dimension was $N = N_{\text{Strings}} - 1 = 3$. The classification was performed using a SVM classifier with a RBF kernel. For the configurations 1 to 5, the number of possible permutations is given in the seventh column of Table 5.7. In each permutation, mean class precision, recall, and F-measure were computed. For the configurations 6.a - 6.c, none of the criteria to separate the training and the test set was applied. Instead, a 10-fold cross-validation was applied and the precision, recall, and F-measure values were averaged over all folds.

Baseline Experiment (MFCC features)

The first baseline experiment was performed in a similar way as shown in Section 5.1—Mel-Frequency Cepstral Coefficients (MFCC) features were used for the given classification task. Similar as to the main experiment, LDA is used for feature space transformation and SVM is used as classifier. The MFCC features were extracted every 10 ms. The feature vectors were classified and evaluated on a frame-level, a 10-fold stratified cross-validation was used for evaluation. It was ensured that frames from the same audio file were not assigned to both the training and the test set in different folds. The classification results were averaged of all folds and are given in Table 5.5.

Table 5.5: Experiment results of the baseline experiment for automatic string classification using MFCC audio features.

Experiment	Number of classes	Mean F-measure \bar{F}
Automatic String Classification	4	46.0 %

Baseline Experiment (Human performance)

In the second baseline experiment, a listening test was conducted to measure the performance of human listeners for the given task of classifying the string number based on isolated bass guitar notes. The study comprised 9 participants, most of them being semi-professional guitar or bass players. To allow for a comparison between the algorithm performance and the human performance, similar test conditions as for experiment 1.6.c. were achieved as follows. The samples were randomly assigned to training and test set—no separation based on playing technique, pickup setting, or instrument was performed. During the training phase, the participants were allowed to listen to an arbitrary number of notes from the training set for each string class. Afterwards, the participants were asked to assign randomly selected samples from the test set to one of the 4 string classes.

Table 5.6: Confusion matrix for human performance for string classification. All values are given in percent. It can be observed that most false classifications are towards adjacent instrument strings.

	E	A	D	G
E	47.0	43.0	9.9	0.1
A	18.7	53.7	26.8	0.8
D	0.8	18.2	64.5	16.5
G	0.7	7.0	35.0	57.3

As it can be seen in the confusion matrix in Table 5.6, human listeners tend to confuse notes between adjacent strings on the instrument. In total, a mean class accuracy of 55.6 % was achieved.

Results & Summary

The results for the automatic classification of different experimental configurations are shown in Table 5.7. Both the plausibility filter as well as the result aggregation step significantly improve the classification results in most configurations. Furthermore, the separation of training and test samples according to instrument, playing technique, and pickup setting has a strong influence on the achievable classification performance. In general, the results obtained for the bass guitar and the electric guitar show the same trends [4]. The highest classification score of $\bar{F} = .93$ was achieved for bass guitar notes using configuration 6.c. In this configuration, the samples in the training and test set were not separated w.r.t. instrument, playing technique, and pick-up setting and the plausibility filter and the result aggregation were used to improve the performance.

5.4 Transcription of Bass Guitar Tracks

Motivation & Goals

In this experiment, the performance of the proposed bass guitar transcription algorithm is evaluated on “real-world” bass guitar tracks and compared to 3 state-of-the-art bass transcription algorithms. The results of this experiment were published in [14].

Table 5.7: Mean class precision \bar{P} , recall \bar{R} , and F-measure \bar{F} for automatic bass guitar string classification for different experimental conditions. The best performance of $\bar{F} = 93.0\%$ (highlighted in bold print) was achieved using the plausibility filter and the frame-wise result aggregation.

Experiment	Separated instruments in training & test set	Separated playing techniques in training & test set	Separated pickup settings in training & test set	Plausibility filter	Result aggregation over 5 frames	No. of Permutations [◊] No. of CV folds*	\bar{P}	\bar{R}	\bar{F}
1.a	x				2 [◊]	85.0	85.0	85.0	
1.b	x			x	2 [◊]	87.0	87.0	87.0	
1.c	x			x	2 [◊]	78.0	78.0	78.0	
2.a	x	x			8 [◊]	86.0	86.0	86.0	
2.b	x	x		x	8 [◊]	87.0	87.0	87.0	
2.c	x	x		x	8 [◊]	88.0	88.0	88.0	
3.a		x	x		8 [◊]	57.0	50.0	49.0	
3.b		x	x	x	8 [◊]	71.0	69.0	69.0	
3.c		x	x	x	8 [◊]	88.0	88.0	88.0	
4.a		x			8 [◊]	60.0	54.0	54.0	
4.b		x		x	8 [◊]	73.0	71.0	72.0	
4.c		x		x	8 [◊]	93.0	93.0	93.0	
5.a			x		8 [◊]	62.0	55.0	54.0	
5.b			x	x	8 [◊]	74.0	71.0	71.0	
5.c			x	x	8 [◊]	92.0	92.0	92.0	
6.a					10*	92.0	92.0	92.0	
6.b				x	10*	93.0	93.0	93.0	
6.c				x	10*	93.0	93.0	93.0	

Dataset

The *IDMT-SMT-BASS-SINGLE-TRACKS* dataset introduced in Section 4.2.1 was used for evaluation. Figure 5.1 illustrates a pitch histogram over the evaluation dataset. The notes with a MIDI pitch above 48 are played with the *harmonics* expression style.

Algorithms

The proposed algorithm explained in Section 4.1 (denoted as **A**) was compared against three state-of-the-art bass transcription algorithms by Ryynänen & Klapuri [153] (**R**), Salamon [155] (**S**), and Dittmar et al. [46] (**D**). Algorithm **S** is limited to a two-octave pitch range between the

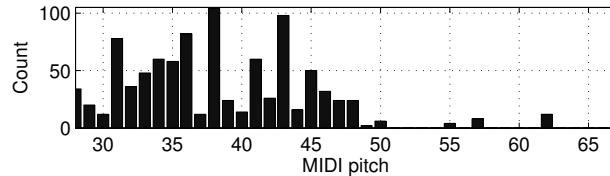


Figure 5.1: Pitch histogram over all notes in the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset.

MIDI pitch values 21 and 45 (fundamental frequency values between 27.5 Hz - 110 Hz).

Experimental Procedure

For all 17 basslines in the dataset, the annotated note parameters onset, offset, and pitch are used as ground truth data.

The algorithms **R**, **D**, and **A** provide note-wise transcription results, hence each detected note is characterized by its onset and offset position as well as its MIDI pitch. The algorithm **S** provides only frame-wise estimates of the fundamental frequency. Therefore, only the frame-wise evaluation measures can be computed here. For the algorithms **R**, **D**, and **A**, frame-wise fundamental frequency values are obtained using the same temporal resolution of $\Delta t = 5.8$ ms as in **S**. All algorithms and the extracted transcription parameters are listed in Table 5.8.

Recall R , precision P , and F-measure F are used as *note-wise evaluation measures*. The recall is defined as the number of correctly transcribed notes divided by the total number of reference notes. The precision is defined as the number of correctly transcribed notes divided by the total number of transcribed notes. The F-measure combines both measures as $F = 2RP/(R + P)$. A note is considered as correctly transcribed, if it can be assigned to a reference note with the same MIDI pitch and if its onset has a maximum absolute deviation to the ground truth of 150 ms as proposed in [152].

As for *frame-wise evaluation measures*, five global measures from the MIREX 2005 competition for melody extraction are used as explained for instance in [156]—Voicing Recall Rate VRC (proportion of correctly detected ground-truth melody frames), Voicing False Alarm Rate $VFAR$ (proportion of ground-truth non-melody frames mistakenly detected as melody frames), Raw Pitch Accuracy RPA (proportion of detected melody frames with the correct pitch), Raw Chroma Accuracy RCA (proportion of detected melody frames with the correct pitch, octave errors are ignored), as well as Overall Accuracy OA (combined performance measure for pitch estimation and voicing detection).

Results & Summary

The frame-wise evaluation measures are shown in Table 5.9. Algorithm **S** outperforms the others in the detection of voiced frames with the highest Voicing Recall Rate of $VRC = 0.934$. In terms of pitch estimation, the proposed algorithm **A** outperforms the others with a Raw Pitch Accuracy of $RPA = 0.765$. However, if the octave information is neglected, algorithm **S** shows the best performance for the Raw Chroma Accuracy with $RCA = 0.82$. Keeping in mind that

Table 5.8: Compared bass transcription algorithms and applicable evaluation measures. All algorithms except **S** allow to compute both frame-wise and note-wise evaluation measures.

Algorithm	Ref.	Evaluation Measures	
		Frame-wise	Note-wise
Ryynänen & Klapuri (R)	[153]	x	x
Salamon (S)	[155]	x	-
Dittmar et al. (D)	[46]	x	x
Abeßer (A - proposed)	[14]	x	x

the algorithm **S** only considers a limited pitch range of two octaves, a better performance of **S** for the Raw Chroma Accuracy is likely if a larger pitch range would be considered.

Table 5.10 illustrates the results of the note-wise evaluation measures. Here, the proposed algorithm **A** clearly outperforms the other two algorithms **R** and **D** in recall ($R = 89.7\%$), precision ($P = 90.8\%$), and F-measure ($F = 90.1\%$). While **R** and **D** show comparable precision values, **R** clearly has the higher recall value in the direct comparison.

The parameters of the proposed algorithm were optimized using isolated bass guitar notes recorded with the same instrument that was used to record the basslines in the evaluation set. Therefore, the obtained results can be interpreted as upper performance limit under idealized conditions since in a real-world application, the bass guitar tracks that are to be transcribed are very likely recorded with a different instrument. However, the conditions can be recreated in music education software, if the transcription algorithm can be adapted to the particular sound of the user's instrument.

Table 5.9: Frame-wise evaluation results for score-level evaluation. The best performing algorithms are indicated in bold print for each evaluation measure.

Algorithm	Evaluation Measures				
	VRC	VFAR	RPA	RCA	OA
R	0.835	0.209	0.696	0.794	0.728
S	0.934	0.296	0.701	0.82	0.698
D	0.741	0.291	0.585	0.624	0.606
A	0.89	0.427	0.765	0.796	0.735

5.5 Estimation of Instrument-level Parameters from Bass Guitar Tracks

Motivation & Goals

Finally, an experiment for the estimation of the instrument-related parameters plucking style (PS), expression style (ES), and string number (SN) from realistic bass guitar recordings was

Table 5.10: Note-wise evaluation results for score-level evaluation. All values are given in percent. As indicated in bold print, the proposed transcription algorithm outperforms the other two algorithms in all three note-wise evaluation measures.

Algorithm	Evaluation Measures		
	<i>R</i>	<i>P</i>	<i>F</i>
R	75.1	84.1	78.7
D	51.2	81.5	59.9
A	89.7	90.8	90.1

performed and published in [14].

Dataset

Similarly to Section 5.4, the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset (compare Section 4.2.1) was used for evaluation.

Experimental Procedure

In order to rule out the transcription step as potential source or error in this experiment, ground truth annotations were used for the note pitch, onset, and offset values. The three models for the automatic classification of string number (SN), plucking style (PS), and expression style (ES) were initially trained with notes from the development set *DS-2* as explained in Section 4.1.1. For the PS classifier, all notes from *DS-2* were selected that were played with the *normal* expression style (NO). For the ES classifier, all notes were used that were played with the *finger-style* (FS) plucking style. For the SN classifier, all notes were used that were not played with the *dead-note* (DN) nor the *harmonics* (HA) expression style. After the complete feature set is extracted, the three aforementioned note parameters are classified and the confusion matrices are obtained.

Results & Summary

The confusion matrices for the estimation of the instrument-related parameters PS, ES, and SN are shown in Table 5.12, Table 5.13, and Table 5.11. For PS and SN classification, a main diagonal is clearly visible—mean class accuracy values of 63.6 % and 75.2 % were achieved. Due to the error correction explained in Section 4.1.11, all basslines in the dataset played with the slap plucking styles *slap-thumb* and *slap-pluck* were correctly identified, since no confusion appears with the other plucking style classes. However, there is a frequent confusion between both slap classes. Notes played with the *finger-style* (FS) style are often confused to the *muted* style. For the ES classification, only the BE class shows satisfying results. The mean accuracy for ES classification is 44.2 %. The other classes—especially NO and DN—show strong confusions towards other classes.

As shown before in Section 5.1 and Section 5.3, the presented approach of feature-based classification of the instrument-related parameters achieved very high classification results if isolated notes are used for training and prediction. Due to the rhythmic structure of the basslines used in this experiment, note durations are much shorter than in the training set. This presumably

caused the strong confusion of FS notes towards the MU expression style class. Also, while the training set contained only 11 different combinations of plucking and expression, the evaluation set included 19 different combinations, which lead to a greater variety in different instrument sounds and which made it more challenging for the classifier models to make the right class predictions. A possible solution to overcome these problems is to include shorter notes as well as notes played with more different combinations of plucking and expression styles into the training set.

Table 5.11: Confusion matrix for string number classification from bass guitar tracks. All values are given in percent.

	E	A	D	G
E	98.8	1.2	0	0
A	15.2	68.7	15.6	0.4
D	1.0	19.4	66.0	13.6
G	0	0	33.0	67.0

Table 5.12: Confusion matrix for plucking style classification from bass guitar tracks. All values are given in percent.

	FS	MU	PK	SP	ST
FS	37.6	62.4	0	0	0
MU	0	100.0	0	0	0
PK	0	39.9	60.1	0	0
SP	0	0	0	81.6	18.4
ST	0	0	0	61.1	38.9

Table 5.13: Confusion matrix for expression style classification from bass guitar tracks. All values are given in percent.

	NO	BE	VI	HA	DN	SL
NO	0.3	54.9	17.2	2.3	0.3	25.0
BE	0	87.5	12.5	0	0	0
VI	0	41.9	51.6	0	0	6.5
HA	16.1	19.4	9.7	48.4	0	6.5
DN	11.1	0	11.1	55.6	22.2	0
SL	0	35.0	10.0	0	0	55.0

6 Summary

In the first part of this thesis, a novel transcription algorithm was presented that is tailored towards the characteristics of the electric bass guitar. The algorithm extracts note parameters on two levels—the score-level and the instrument-level. In Chapter 5 it was shown that the proposed system performed very well for the estimation of plucking style, expression style, as well as string number from isolated note recordings.

Also, given the idealized condition that the same bass guitar was used to record notes in the training and test set, the proposed transcription algorithm could clearly outperform three state-of-the-art bass transcription algorithm on a novel dataset of realistic bass guitar recordings. These idealized conditions can be recreated in music education software, if a transcription algorithm can be adapted to the particular sound of the user’s instrument.

In the remaining two parts of the thesis, the extracted set of note parameters is applied in two application scenarios. First, in Part II, the score-level parameters note pitch, onset, and offset are used to classify the musical genre based on the underlying bassline. Second, in Part III, the complete set of parameters is transmitted to a sound synthesis algorithm in order to re-synthesize the original bassline by simulating the instrument’s sound production mechanisms.

Part II

Application for Music Genre Classification

Preface

In contrast to melody lines and harmonic progressions, basslines have rarely been investigated in the field of MIR. The bass track plays an important role in music genres of different historical epochs from Western classical Baroque music to contemporary genres such as heavy metal or drum'n'bass, as well as genres from various regional traditions from Western European, American, and African countries. In popular music genres, typical bass patterns have evolved over time. The bassline carries important rhythmic and structural information of a song and provides insight to its underlying harmonic progression. Hence, the automatic classification of extracted basslines allows to describe a music piece in terms of harmony, rhythm, and style.

This part of the thesis is structured as follows. In Chapter 7, related work on genre classification using score-based audio features will be reviewed. Amongst others, algorithms will be compared, which extract semantic information from the bassline of a song. In Chapter 8, a mixed set of adapted and newly developed score-based audio features will be presented, which allows for quantifying different musical properties of basslines w.r.t rhythm, tonality, and structure. Also, a novel dataset will be introduced that comprises 520 basslines from 13 different music genres stored in the symbolic MIDI format. Finally, Chapter 9 presents a large-scale genre classification experiment. Three different paradigms will be compared for the task of genre classification: classification based on statistical pattern recognition, classification based on bass pattern similarity, and rule-based classification. Finally, the experimental results will be discussed and conclusions towards future research will be drawn.

Parts of the following sections have been published in [6], [12], [13], and [10].

7 Related Work

7.1 Genre Classification using Score-based Audio Features

Music genre recognition is one of the most popular tasks in the field of MIR. Sturm reviewed over 430 related publications in [163]. This section focuses on publications towards music genre classification using score-based audio features. The main focus is on approaches that analyze the bassline of music pieces.

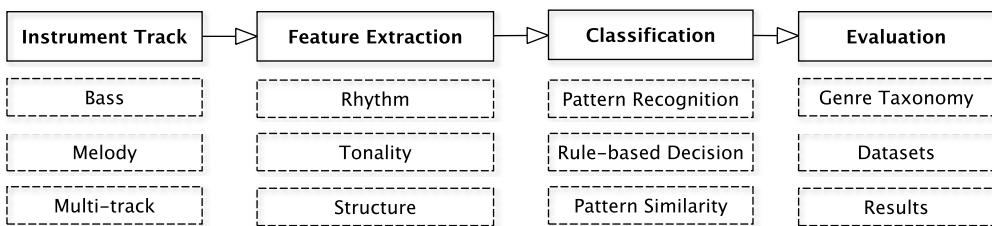


Figure 7.1: Criteria applied to categorize publications towards genre classification using score-based audio features.

As shown in Figure 7.1, different criteria are applied to categorize the selected publications: the instrument track that is analyzed, the proposed set of audio features, the applied classification paradigm, the applied genre taxonomy and dataset size, as well as the achieved classification results. All criteria will be detailed in the following sections.

7.1.1 Instrument Track

Genre classification algorithms that extract audio features from the *bass track* are presented in [5, 10, 12, 99, 157, 169, 170]. Simsekli points out in [157] that “basslines form a bridge between the melody and the rhythm section. Hence they encapsulate both rhythmic, melodic, and harmonic information in a simple, monophonic melody”. Kitarahara discusses that basslines often have a simple, phrase-like structure, which is repeated [99]. Therefore, as will be shown in Section 9.1, the analysis of the bassline of a musical piece is a meaningful approach to classify its genre.

The second group of publications extract features from the *main melody* of a song [15, 33, 40, 42, 94, 112, 140, 174]. The main melody is one of the most memorizable parts of a music piece since it is also commonly repeated. Different publications show that the tonal and rhythmic properties of the melody are often well-discriminative among different music genres. Remarkably, the type of instrument that is used to play the main melody is not considered as additional feature in the literature.

In the third group of publications, *multiple instrument tracks* are analyzed [6, 21, 43, 111, 121].

The score of a music piece can be easily converted to a MIDI file. For the analysis of real audio data, very accurate polyphonic music transcription algorithms are required to extract a symbolic music representation. However, as stated for instance in [102], state-of-the-art polyphonic music transcription algorithms will presumably remain error-prone to a certain extend in the near future.

7.1.2 Feature Extraction

Semantic Levels of Audio Features

In the field of MIR, various *audio-based* features were proposed that can be extracted from the audio signal itself or from a derived time-frequency representation such as the spectrogram [163]. Those features are commonly categorized into three semantic layers: *low-level* features, *mid-level* features, and *high-level* features. Low-level features are extracted on a short time scale (commonly several milliseconds) and relate to simple perceptual signal properties such as the frequency centroid or the loudness. Mid-level features are computed on a larger time scale such as the chromagram [127] or beat histograms [72]. These features capture different aspects from music perception on a higher semantic level [57]. High-level features relate to musical terms such as key, tempo, or time signature and can therefore easily be interpreted by human experts.

Score-based (or symbolic) features can be compared to the audio-based high-level features since they are extracted from a musically meaningful (score-level) representation of music pieces. As discussed in Section 3.1, score parameters such as pitch and onset can either be extracted from audio recordings via a preliminary music transcription step or from existing symbolic audio formats such as MIDI, MusicXML, or Humdrum [119]. In this thesis, the focus is solely on genre classification using score-based features.

Musical Domains

Score-based audio features relate to different musical domains such as tonality, rhythm, structure, and timbre. The first group of features are related to the *tonality* of given melodies or basslines. Features that characterize the melodic shape are used in [5, 99, 140, 169]. The melodic shape or contour describes how the absolute pitch of a melody changes over time. Tzanetakis et al. compute simple histograms from the absolute pitch and the pitch class¹ in [174]. Since pitch histograms change if melodies are transposed², other authors propose to use interval histograms instead [21, 157]. A *scale* defines a sub-set of the 11 chromatic notes, which can be played in a given harmonic context of a chord. The use of different scales is often genre-specific. Therefore, likelihood measures of different scales are extracted using a pattern matching algorithm in [5, 10] and used as features.

The second group of features describe *rhythmic* properties. For this purpose, the distribution of the note parameters onset [140], duration [94], as well as the inter-onset-interval (distance between consecutive note onsets) [112] are analyzed using different statistical measures to extract

¹As will be shown in Section 8.1.1, the pitch class simplifies the absolute pitch representation by neglecting the octave information.

²Transposition denotes a constant shift of all absolute pitch values in a melody according to the given key.

features. More complex rhythmic phenomena such as syncopations and rhythmic subdivisions are investigated in [5].

The third feature group is based on the temporal *structure* of note sequences. Multiple authors exploit the repetitive, pattern-like structure of melodies or basslines. For instance, the similarity between repetitive bass patterns is either used for feature extraction [94, 170] or directly for classifying unknown basslines [10].

Finally, *timbre-related* properties of music pieces are analyzed for feature extraction. As shown in [163], most of the publications focusing on audio-based features try to use timbre descriptors for genre classification tasks. However, symbolic audio formats represent note events on an abstract level without capturing instrument-specific acoustic properties. Nevertheless, the MIDI format allows to annotate the instrument type associated to each MIDI track. The presence of particular instrument or instrument group is used as timbre feature in [21, 121].

Time-windowing & Hybrid Feature Extraction

Most publications extract one set of features over the complete melody or bassline. In contrast, Pérez-Sancho et al. as well as León & Iñesta apply a sliding window to compute local feature values at different time instances [40, 140]. Similarly, Cataltepe et al. convert the melody in a MIDI file into a character string over different time spans within the music piece [33].

Score-based features can be used to complement existing audio-based features in a hybrid feature extraction framework [99, 112, 169]. Here, the audio features describe properties related to rhythm and timbre while the score-based features relate to melodic properties. Some authors follow an inverse approach. Existing MIDI files are re-synthesized to audio files in order to extract audio-based features [33].

7.1.3 Classification

Classification Paradigms

In the literature reviewed in this section, three different classification paradigms are investigated: The first type of classification systems applies *pattern similarity* measures: Melodies and basslines are compared based on their melodic and rhythmic similarity. The second classification paradigm is based on *statistical pattern recognition*: A statistical classifier model assigns a song to a music genre class based on an extracted feature vector. The third type of classifiers is based on *rules* or *expert knowledge*: A set of decision rules is first extracted based on the class distributions in the feature space and then applied for genre classification.

Classification Based on Pattern Similarity The computation of similarity between different melodies is useful for both music retrieval and analysis. First, the melody is converted into a character string by transforming the note-wise score parameter values such as the absolute pitch into a sequence of characters. Different distance measures are applied to measure the similarity between melodies as for instance the Edit Distance [125], the Earth Mover's Distance (EMD), and the derived Proportional Transportation Distance (PTD) [172]. Other similarity measures are derived from the perception-based Implication/Realization (I/R) model [70] or from a graph-based

representation of musical structure [132]. Genre classification based on similarity between melodies is discussed in [6, 15, 33]. The similarity between bass patterns is investigated in [5, 10, 170].

Classification Based on Statistical Pattern Recognition Statistical pattern recognition methods are widely applied for different MIR tasks such as genre, mood, or style classification. After a set of features is extracted, the dimensionality of the feature space is reduced using feature space transformation techniques such as Linear Discriminant Analysis (LDA) [6] and Principal Component Analysis (PCA) [99] or using feature selection techniques such as Inertia Ratio Maximization using Feature Space Projection (IRMFSP) [12] and feature grouping [40].

In the literature discussed in this section, various classification algorithms are used: Support Vector Machines (SVM) [6, 10, 15, 112, 140], Neural Networks (NN) [121], k-Nearest Neighbor [33, 40, 121, 140, 157, 174], Bayesian classifier [40, 99, 140], Self-Organizing Map (SOM) classifier [40], and Multi-layer Perceptron [140]. Machine learning algorithms for classification, feature selection, and feature space transformation that are used in the experiments described in this thesis are detailed in Section 2.3.

Rule-based Classification & Expert Systems The modeling of different genres or styles using a list of rules is more intuitive and comprehensible for humans than using statistical pattern recognition methods [11, 41]. Each rule corresponds to a distinct musical property and expresses a simple feature-value-relation. The automatic learning of these rules is presented for harmony progressions [16], melody characterization [41], as well as for automatic music generation [27]. The Classification and Regression Tree (CART) algorithm was used in [10] to find rules to distinguish 13 different music genres from a global background based on repetitive bass patterns.

Hierarchical Structure of Data Sets & Classification Aggregation

In many publications, data sets with a hierarchical genre taxonomy, i.e., a set of root genres having multiple leaf-genres, are used [33, 43, 121, 157]. In the literature, the aggregation of multiple classifiers is performed via result weighting [94, 111], Bayesian decision frameworks [42], or classifier ensembles [140].

7.1.4 Evaluation

Throughout the literature, *cross-validation* is used to evaluate classification algorithms (compare Section 2.3.1). Given a data set of various item examples (songs, melodies, or basslines) for different music genres, multiple evaluation folds are performed. In each fold, a different set of items is assigned towards the training set, which is used to train the classifier, and the test set, which is classified using the trained classifier. Based on the known genre class labels of the test set items, different evaluation measures are computed for each fold and finally aggregated over multiple folds.

Figure 7.2 gives an overview over the classification results reported in the discussed publications. For each publication, both the average number of class items per genre as well as the average classification accuracy is shown. Two observations can be made that complicate the comparison of the reviewed publications. First, the number of class items per genre strongly varies from 20 [174]

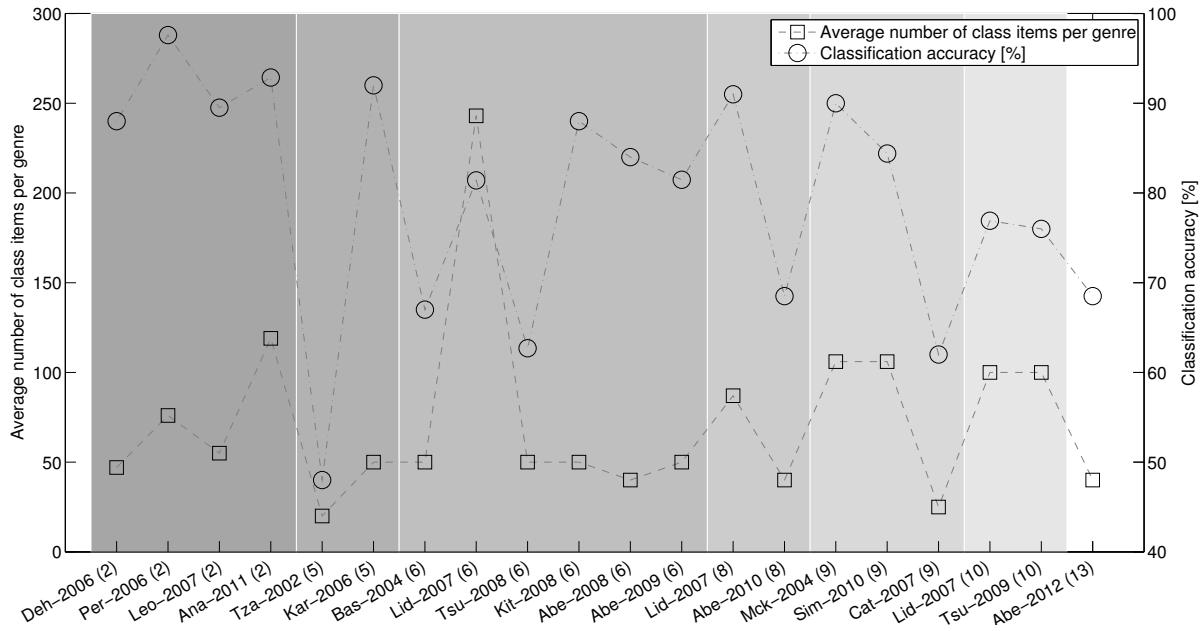


Figure 7.2: Average number of class items per genre and highest classification accuracy for all discussed publications. Publications are ordered by increasing number of genres, which is also given in brackets. Publications with the same number of genres are grouped in the figure by using the same background color. Abbreviations for publications are Deh-2006 [43], Per-2006 [140], Leo-2007 [40], Ana-2011 [15], Tza-2002 [173], Kar-2006 [94], Bas-2004 [21], Lid-2007 [112], Tsu-2008 [169], Kit-2008 [99], Abe-2008 [6], Abe-2009 [12], Lid-2007 [112], Abe-2010 [5], Mck-2004 [121], Sim-2010 [157], Cat-2007 [33], Lid-2007 [112], Tsu-2009 [170], and Abe-2012 [10].

to 243 [112]. In order to obtain a robust genre classification system, the number of class items should be as high as possible in order to best represent a musical genre by means of those examples. Second, the number of genres varied from 2 [15, 40, 43, 140], 5 [94, 174], 6 [6, 12, 21, 99, 112, 169], 8 [5, 112], 9 [33, 121, 157], 10 [112, 170], up to 13 [10]. The classification tasks naturally becomes more difficult with a higher number of classes. Therefore, it can be observed in Figure 7.2 that the classification accuracy values decrease with increasing number of genres.

7.1.5 Conclusion

As shown in [5, 10, 12, 99, 157, 169, 170], the analysis of the bassline leads to promising genre classification results. However, the best results were obtained by analyzing multiple instrument tracks. Hybrid approaches that combine audio-based and score-based features using an intermediate music transcription algorithm obtain high classification results for different audio datasets [112]. The main advantage is that these approaches are not restricted to the analysis of symbolic audio files but instead allow to analyze real audio recordings. Nevertheless, score-based features strongly rely on the quality of the transcription results. Hence, the performance of this hybrid approach depends on further improvements in the field of automatic music transcription.

8 Contribution

8.1 Score-based Audio Features

In this chapter, various *score-based audio features* will be introduced that allow to describe a bassline from three different musical perspectives—rhythm, tonality, and structure. For each perspective, several note representations will be discussed first and then related audio features will be described.

8.1.1 Tonality

Representations

In the following sections, different tonal note representations are explained that will be used throughout this part of the thesis. As an example, Table 8.1 illustrates a funk bassline with the corresponding note representation values.

Table 8.1: Tonal note representations of a funk bassline. Score notation is given on top and different note parameters are given for all notes below.

		Score notation											
Note number	i	1	2	3	4	5	6	7	8	9	10	11	12
Note name		C_3	G_3	A_3	$A\sharp_3$	B_2	C_3	G_3	$A\sharp_3$	A_3	G_3	$A\sharp_2$	B_2
Absolute pitch	$\mathcal{P}(i)$	48	55	57	58	47	48	55	58	57	55	46	47
Pitch class	$\mathcal{P}_{12}(i)$	0	7	9	10	11	0	7	10	9	7	10	11
Relative pitch	$\mathcal{P}_\Delta(i)$	7	2	1	-11	1	7	3	-1	-2	-9	1	-
Chromatic interval	$\mathcal{P}_{\Delta,12}(i)$	7	2	1	-11	1	7	3	-1	-2	-9	1	-
Interval direction	$\mathcal{P}_{\Delta,D}(i)$	1	1	1	-1	1	1	1	-1	-1	-1	1	-
Diatonic intervals	$\mathcal{P}_{\Delta,7}(i)$	5	2	2	-7	2	5	3	-2	-2	-6	2	-

Absolute Pitch The absolute pitch $\mathcal{P} \in \mathbb{Z}^N$ ($0 \leq \mathcal{P}(i) \leq 127$) corresponds to the note's MIDI pitch value. The number of notes in a bassline is denoted as N . For instance, the musical note A_4 (pitch class A in the fourth octave) has a fundamental frequency of $f_0 = 440$ Hz and an absolute pitch value of $\mathcal{P}(i) = 69$.

Pitch Class The pitch class $\mathcal{P}_{12} \in \mathbb{Z}^N$ is computed as

$$\mathcal{P}_{12}(i) = \mathcal{P}(i) \bmod 12 \quad (8.1)$$

and provides an octave-invariant representation of the absolute pitch. The pitch class $\mathcal{P}_{12}(i) = 0$ corresponds to the note name *C*.

Relative Pitch The *relative pitch* $\mathcal{P}_\Delta \in \mathbb{Z}^{N-1}$ (interval) describes the pitch difference in semitones between two consecutive notes.

$$\mathcal{P}_\Delta(i) = \mathcal{P}(i+1) - \mathcal{P}(i) \quad (8.2)$$

Interval Direction The *interval direction* $\mathcal{P}_{\Delta,D} \in \mathbb{Z}^{N-1}$ measures whether two adjacent notes show an increase or decrease in pitch or whether the absolute pitch remains constant.

$$\mathcal{P}_{\Delta,D}(i) = \operatorname{sgn} \mathcal{P}_\Delta(i) \quad (8.3)$$

Chromatic Interval The *chromatic interval* $\mathcal{P}_{\Delta,12} \in \mathbb{Z}^{N-1}$ is obtained by mapping the absolute interval \mathcal{P}_Δ to a range of one octave upwards and downwards as

$$\mathcal{P}_{\Delta,12}(i) = \begin{cases} \mathcal{P}_\Delta(i) \bmod 12 & \text{if } \mathcal{P}_\Delta(i) \geq 0, \\ -(-\mathcal{P}_\Delta(i) \bmod 12) & \text{otherwise.} \end{cases} \quad (8.4)$$

Diatonic Interval The *diatonic interval* is denoted as $\mathcal{P}_{\Delta,7} \in \mathbb{Z}^{N-1}$. For each chromatic interval $\mathcal{P}_{\Delta,12}$, a corresponding diatonic interval $\mathcal{P}_{\Delta,7}$ can be associated as shown in Table 8.2.

Feature Extraction

In this section, all features will be described that are used in the genre classification experiments described in Section 9.1. Similarly to the first part of the thesis, features will be denoted as χ with a corresponding subscript.

Pitch Range The pitch range is computed as

$$\chi_{\text{PitchRange}} = \max \mathcal{P} - \min \mathcal{P}. \quad (8.5)$$

Relative Frequency of Dominant Pitch The *dominant absolute pitch* \mathcal{P}_{dom} value is the most frequently appearing absolute pitch value. It is determined as

$$\mathcal{P}_{\text{dom}} = u^{[\mathcal{P}]}(i_{\max}) \quad (8.6)$$

with

$$i_{\max} = \arg \max_i n^{[\mathcal{P}]}(i). \quad (8.7)$$

Table 8.2: Interval names and corresponding diatonic and chromatic intervals. Since for the analysis of the bassline contours only the pitch distances and not the harmonic function/meaning of the pitches matter, a simplified but unambiguous mapping from chromatic to diatonic interval values is used here.

Interval name	Diatonic interval $\mathcal{P}_{\Delta,7}$	Chromatic interval $\mathcal{P}_{\Delta,12}$
Descending seventh	-7	-11,-10
Descending sixth	-6	-9,-8
Descending fifth	-5	-7,-6
Descending fourth	-4	-5
Descending third	-3	-4,-3
Descending second	-2	-2,-1
Prime	1	0
Ascending second	2	1,2
Ascending third	3	3,4
Ascending fourth	4	5
Ascending fifth	5	6,7
Ascending sixth	6	8,9
Ascending seventh	7	10,11

In case multiple absolute pitch values appear equally often, the lowest of these values is selected as dominant absolute pitch. The relative frequency of the dominant pitch is used as a feature:

$$\chi_{\text{FreqDomPitch}} = p^{[\mathcal{P}]}(i_{\max}). \quad (8.8)$$

A high value of $\chi_{\text{FreqDomPitch}}$ can indicate a rather simple bassline with many pitch repetitions and only a few tonal variations. The auxiliary functions u , n , and p are defined in Section 1.6.

Pedal Tone Some basslines are based on a *pedal tone*. Here, the dominant pitch is one of the lowest pitch values that are present in the bassline. The feature $\chi_{\text{PedalTone}}$ measures this property of the pitch distribution as follows:

$$\chi_{\text{PedalTone}} = \frac{\mathcal{P}_{\text{dom}} - \min \mathcal{P}}{\max \mathcal{P} - \min \mathcal{P}} \quad (8.9)$$

If $\chi_{\text{PedalTone}}$ is small, the dominant pitch is comparably low and the use of a pedal tone in the bassline is more likely.

Tonal Complexity & Pitch Class Entropy Among others things, the *tonal complexity* of a bassline depends on the number of unique pitch class values it contains. In each of the bass patterns investigated in this part of the thesis, the key remains constant and one scale, which can be represented by 7 pitch class values, is used. Simple basslines most consist of the root (and

octave), and the fifth of the present chord. basslines with a higher tonal complexity include also other chord tones such as thirds and sevenths.

In order to derive a quantitative measure for tonal complexity, the flatness of the pitch class distribution $p^{[\mathcal{P}_{12}]}$ of a bassline is measured using the zero-order entropy as

$$\chi_{\text{ChromPitchEntropy}} = \sum_{i=1}^N p^{[\mathcal{P}_{12}]}(i) \log_2 p^{[\mathcal{P}_{12}]}(i). \quad (8.10)$$

High entropy values indicate that the distribution $p^{[\mathcal{P}_{12}]}$ tends towards an equal distribution, which would imply a high number of unique pitch class values.

Interval Properties The absolute interval size is

$$\mathcal{P}_{\Delta,\text{abs}}(i) = |\mathcal{P}_{\Delta}(i)| \text{ with } \mathcal{P}_{\Delta,\text{abs}} \in \mathbb{Z}^{N-1}. \quad (8.11)$$

The mean over $\mathcal{P}_{\Delta,\text{abs}}$ is used as feature to characterize the average absolute interval size within a given bassline. The standard deviation over $\mathcal{P}_{\Delta,\text{abs}}$ is computed as a feature to characterize the fluctuation of the absolute interval size over time. In addition, the number of unique intervals is computed as

$$\chi_{\text{UniqueIntervals}} = \dim(v^{(\mathcal{P}_{\Delta})}). \quad (8.12)$$

Simple basslines such as a repetitive, octave-based minimal techno bass line have a smaller number of unique intervals than more complex basslines from other genres.

Typical Interval Progressions Some interval progressions in basslines are characteristic for certain music genres. The following features capture the relative frequency of constant pitch sequences and chromatic transitions between notes.

A ratio of *constant pitch sequences* is computed as

$$\chi_{\text{ConstPitch}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \delta_i \text{ with } \delta_i = \begin{cases} 1 & , \text{ if } \mathcal{P}(i) \equiv \mathcal{P}(i+1) \\ 0 & , \text{ otherwise.} \end{cases} \quad (8.13)$$

If two adjacent intervals have an absolute size of one semi-tone, the note transition is *chromatic*. basslines in some jazz-related music genres¹ often show chromatic note transitions. Similarly to the previous feature, a feature can be derived as

$$\chi_{\text{ChromNoteTrans}} = \frac{1}{N-2} \sum_{i=0}^{N-2} \delta_i, \quad \delta_i = \begin{cases} 1 & , \text{ if } |\mathcal{P}_{\Delta}(i)| \equiv |\mathcal{P}_{\Delta}(i+1)| \equiv 1 \\ 0 & , \text{ otherwise.} \end{cases} \quad (8.14)$$

¹This bass playing style is commonly referred to as *walking bass*.

Melodic Contour The *melodic contour*, i.e., the melodic shape of a bassline is perceived as fluent if multiple adjacent intervals have the same interval direction. The *ratio of constant direction* is computed as

$$\chi_{\text{ConstIntDir}} = \frac{1}{N-2} \sum_{i=1}^{N-2} \delta_i, \quad \delta_i = \begin{cases} 1 & , \text{ if } \mathcal{P}_{\Delta,D}(i) \equiv \mathcal{P}_{\Delta,D}(i+1) \\ 0 & , \text{ otherwise.} \end{cases} \quad (8.15)$$

The *dominant interval direction* is computed as the ratio between the number of ascending intervals and the total number of ascending and descending intervals:

$$\chi_{\text{DomIntDir}} = \begin{cases} \frac{n^{[\mathcal{P}_{\Delta,D}](1)}}{n^{[\mathcal{P}_{\Delta,D}](3)} + n^{[\mathcal{P}_{\Delta,D}](1)}} & , \text{ if } n^{[\mathcal{P}_{\Delta,D}](3)} + n^{[\mathcal{P}_{\Delta,D}](1)} > 0 \\ 0 & , \text{ otherwise.} \end{cases} \quad (8.16)$$

with

$$w^{[\mathcal{P}_{\Delta,D}]} = [-1, 0, 1].$$

Small feature values indicate mostly descending intervals and vice versa.

Relative Frequency of Diatonic Interval Classes The diatonic interval representation $\mathcal{P}_{\Delta,7}$ corresponds to musical interval labels as shown in Table 8.2. The *diatonic interval class* $\mathcal{P}_{\Delta,7,\text{abs}} \in \mathbb{Z}^{N-1}$ neglects the interval direction:

$$\mathcal{P}_{\Delta,7,\text{abs}}(i) = |\mathcal{P}_{\Delta,7}(i)|. \quad (8.17)$$

The relative frequencies of the 7 different possible values $\mathcal{P}_{\Delta,7,\text{abs}}(i) \in \{1, 2, \dots, 7\}$ are computed as features to characterize the occurring intervals in a bassline:

$$\begin{aligned} \chi_{\text{FreqDiatonicInt}}(i) &= p^{[\mathcal{P}_{\Delta,7,\text{abs}}]}(i) \\ 1 \leq i &\leq 7 \end{aligned} \quad (8.18)$$

Melodic Scale A *scale* is a unique sub-set of the 12 possible pitch class values that fits to a given key. The choice of the applied scale mainly depends on the underlying harmonic progression. However, some scale types are typically applied in certain musical genres.

As shown in the third column of Table 8.3, a scale can be represented by a binary *scale template* $t_s \in \mathbb{Z}^{12}$ with $t_s(i) \in [0, 1]$. This template describes the unique interval structure between adjacent pitch class values of a scale. Pitch class values that are part of the scale are indicated by $t_s(i) = 1$ and all others by $t_s(i) = 0$. In music practice, melodies rarely only consist of inside-scale notes due to local pitch variations. However, the binary template approach is used here as an effective way to compute likelihood values for the 10 different scales listed in Table 8.3 for a given bassline. These likelihood values are used as features.

The first (non-zero) element of a scale template represents the root note pitch class of the scale. The relative positions of the remaining scale pitch classes can be derived from the interval structure of the scale. Since the root note can have any arbitrary pitch class value, the scale template t_s can be modified by a cyclic shift operation by r semitones to obtain a shifted scale

Table 8.3: Investigated scales with corresponding binary scale templates.

Scale index s	Scale name	Scale template t_s
0	Natural minor	$[1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0]$
1	Harmonic minor	$[1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1]$
2	Melodic minor	$[1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1]$
3	Pentatonic minor	$[1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0]$
4	Blues minor	$[1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1]$
5	Whole tone	$[1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0]$
6	Whole tone half tone	$[1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1]$
7	Arabian	$[1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1]$
8	Minor gypsy	$[1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0]$
9	Hungarian gypsy	$[1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1]$

template $t_{s,r}$. By applying this operation, the interval structure remains the same and the pitch class value of the root note is adopted to the predominant chord or key.

A template-matching approach is used to compute a likelihood value of a scale s for a given bassline. Based on the relative frequencies of all pitch class values $p^{[\mathcal{P}_{12}]}$ in a given bassline, a likelihood measure $\gamma_{s,r}$ can be computed for each rotated version of each scale as

$$\gamma_{s,r} = \left\langle t_{s,r} \mid p^{[\mathcal{P}_{12}]} \right\rangle \quad (8.19)$$

with $0 \leq r \leq 11$

and $\langle \cdot \rangle$ denoting the scalar product. The likelihood measures $\chi_{\text{Scale}}(s)$ for each scale template are computed by maximizing $\gamma_{s,r}$ over all 12 possible rotations as

$$\chi_{\text{Scale}}(s) = \max_{0 \leq r \leq 11} \gamma_{s,r} \quad (8.20)$$

with $0 \leq s \leq 9$

and used as feature.

8.1.2 Rhythm

Representations

In the first part of this thesis, the note onset \mathcal{O} and duration \mathcal{D} was measured in seconds based on a *physical time representation*. In this part, a *musical time representation* is used instead. The onset and duration is measured in fractions of musical bar lengths. This representation is tempo-independent and allows to compare music pieces of different tempo values.² The *MIDI toolbox* [48] is used to extract the score parameters \mathcal{O} and \mathcal{D} directly from MIDI files.

²Ambiguities could result from time signature changes and tempo changes with an octave relationship. However, this is not a problem for the analyzed basslines.

Table 8.4: Rhythmic note representations of a funk bassline. Score notation is given on top and different note parameters are given for all notes below.

													
Note number	i	1	2	3	4	5	6	7	8	9	10	11	12
Note name		C_3	G_3	A_3	$A\sharp_3$	B_2	C_3	G_3	$A\sharp_3$	A_3	G_3	$A\sharp_2$	B_2
Onset	$\mathcal{O}(i)$	0	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{7}{16}$	$\frac{7}{8}$	1	$1\frac{1}{4}$	$1\frac{7}{16}$	$1\frac{10}{16}$	$1\frac{3}{4}$	$1\frac{7}{8}$	$1\frac{15}{16}$
Relative onset	$\mathcal{O}_1(i)$	0	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{7}{16}$	$\frac{7}{8}$	0	$\frac{1}{4}$	$\frac{7}{16}$	$\frac{10}{16}$	$\frac{3}{4}$	$\frac{7}{8}$	$\frac{15}{16}$
Bar number	$\mathcal{B}(i)$	0	0	0	0	0	1	1	1	1	1	1	1
Duration	$\mathcal{D}(i)$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{3}{32}$	$\frac{3}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$
Inter-onset-interval	$\mathcal{O}_\Delta(i)$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{3}{16}$	$\frac{3}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{16}$	-

In the following sections, all note representations related to rhythm will be explained, which are used for feature extraction in this part of the thesis. In Table 8.4, these representations are given for the same example bassline as previously used in Table 8.1.

Relative Onset The *relative onset* $\mathcal{O}_1 \in \mathbb{R}^N$ neglects the bar number and only takes the relative position of a note within its bar into account:

$$\mathcal{O}_1(i) = \mathcal{O}(i) \bmod 1 \quad (8.21)$$

Bar Number The onset of each note can be associated to one bar of the underlying rhythmic structure. The bar number $\mathcal{B} \in \mathbb{R}^N$ is computed as

$$\mathcal{B}(i) = \lfloor \mathcal{O}(i) \rfloor \quad (8.22)$$

using zero-based indexing. The length of a bassline in bars is denoted as N_B .

Inter-Onset-Interval The *inter-onset interval* (IOI) $\mathcal{O}_\Delta \in \mathbb{R}^{N-1}$ is the distance between two consecutive note onsets.

$$\mathcal{O}_\Delta(i) = \mathcal{O}(i+1) - \mathcal{O}(i) \quad (8.23)$$

Metric Level A *metric level* $l \in \mathbb{Z}$ defines a segmentation of a bar into multiple equidistant beats. Table 8.5 shows the beat duration \mathcal{D}_l that correspond to different metric levels for different time signatures $\frac{n}{d}$. The first metric level ($l = 1$) can be interpreted as *beat-level*, the metric levels $l = 2$ and $l = 3$ can be interpreted as first and second *sub-beat level*³. For instance, given a $\frac{4}{4}$ time signature, the first metric level is the quarter-note level, the second metric level is the eighth-note level and so forth.

³The sub-beat level is often referred to as tatum level in the literature.

Table 8.5: Beat durations for different metric levels and different time signatures.

Time signature	Beat duration \mathcal{D}_l in metric level l		
	$l = 1$	$l = 2$	$l = 3$
$\frac{3}{4}, \frac{4}{4}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{6}{8}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$

In each bar, a metric level is defined by $n l$ beats with n denoting the numerator of the time signature. The beats have the relative onset values

$$\mathcal{O}_{1,l}(i) = \frac{i-1}{n l} \text{ with } 1 \leq i \leq n l. \quad (8.24)$$

As will be shown in Section 8.1.2, each note can be assigned to one metric level $l_M(i)$ with $l_M \in \mathbb{Z}^N$.

Pre-Processing

Note Quantization Note quantization refers to the mapping of arbitrary note onset values towards a given metric level. Musical notes can be quantized in time onto the nearest beat positions $\mathcal{O}_{1,l}$ of a given metric level l . The i -th note is mapped to the beat index

$$b_l(i) = \arg \min_{1 \leq b \leq d_l} |\mathcal{O}_1(i) - \mathcal{O}_{1,l}(b)| \text{ with } b_l \in \mathbb{Z}^{d_l}. \quad (8.25)$$

If the note onset is exactly between two beat positions, it is quantized onto the first one. All beats of a metric level can be either classified as *on-beats* or *off-beats* based on their index as

$$b_{\text{On},l}(i) = 2^k - 1, \quad (8.26)$$

$$b_{\text{Off},l}(i) = 2^k, \quad (8.27)$$

$$1 \leq k \leq \log_2(d_l).$$

This definition holds true for time signatures with an even-numbered numerator. On-beats correspond to metrically strong positions in a bar.

Note Mapping Each note of a given bassline can be assigned to one metric level. First, a *mapping cost* value $c_M(i, l)$ is computed based on the smallest distance between the relative onset $\mathcal{O}_1(i)$ of the i -th note and the relative onset values $\mathcal{O}_{1,l}(k)$ of all beats that correspond to the metric level l :

$$c_M(i, l) = \min_{1 \leq k \leq d_l} |\mathcal{O}_1(i) - \mathcal{O}_{1,l}(k)|. \quad (8.28)$$

The metric level $l_M(i)$ of each note is computed by minimizing the mapping costs:

$$l_M(i) = \arg \min_{1 \leq l \leq 4} c_M(i, l) \quad (8.29)$$

For reasons of simplification, the smallest metric level that is considered here is $l = 4$. If multiple metric levels have the lowest cost value $\min_l c_M(i, l)$, the smallest metric level is selected as $l_M(i)$.

Feature Extraction

Tempo The *IDMT-SMT-BASS-GENRE-MIDI* dataset, which is used in the genre classification experiments in this part of the thesis, will be explained in Section 8.2. It contains MIDI files from 13 different music genres including tempo annotations. The tempo of each bassline in beats per minute (bpm) is directly used as feature χ_{Tempo} .⁴ Figure 8.1 illustrates the tempo distribution over basslines from different music genres in the dataset. It can be observed that the tempo is a discriminative feature among the genre classes.

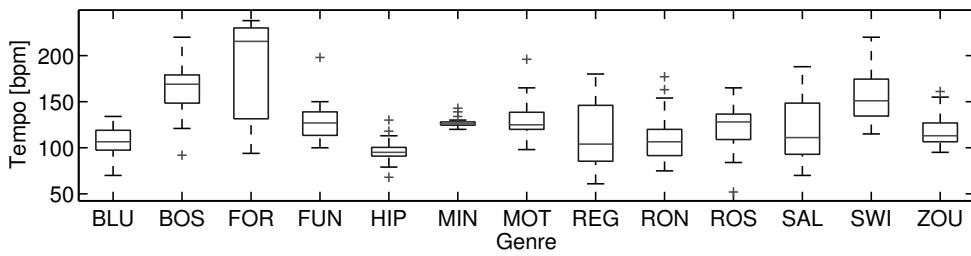


Figure 8.1: Boxplot of tempo values of all bass patterns in the *IDMT-SMT-BASS-GENRE-MIDI* data set for different music genres. The central mark is the median, the edges of the box are the 25th and 75th percentiles. The tempo values are given in beats per minute (bpm). The genre abbreviations are explained in Table 8.6. It can be observed that the tempo is a well-discriminating feature among the genre classes.

Dominant Metric Level The *dominant metrical level* is defined as metrical level whom at least 80 percent of all notes are assigned to:

$$\chi_{\text{DomMetLev}} = u^{[l_M]}(i_{\max}) \text{ with} \quad (8.30)$$

$$p^{[l_M]}(i_{\max}) \geq 0.8 \quad (8.31)$$

In case $p^{[l_M]}(i_{\max}) \geq 0.8$ is fulfilled by multiple metric levels, the smallest metric level is chosen as dominant metric level.

In Figure 8.3, a salsa bassline is shown in the first row. In the second, third, and fourth row, the metric levels $l_M \in \{3, 2, 1\}$ are illustrated as metric positions in the bar. For each metric level, the plus signs indicate on which metric position a corresponding note in the bass exists. For the given example, the dominant metric level is $\chi_{\text{DomMetLev}} = 3$ (sixteenth-notes) since all notes can be assigned to it.

⁴The tempo is obtained from the MIDI files using the *gettempo* function of the MIDI Toolbox for Matlab [48].



Figure 8.2: Excerpt of a salsa bassline in $\frac{4}{4}$ time signature (first row). Here, notes of the bassline, which refer to on-beat positions of the metric level $l_M = 3$ (sixteenth notes) are indicated by a plus sign (+), notes on off-beat positions are indicated by an accent sign (.'). In the second to fourth row, metrical levels $l_M = 3$, $l_M = 2$ (eighth notes), and $l_M = 1$ (quarter notes, beats) are shown. Plus signs indicate that notes in the bassline exist at this metric position.

Average Metric Rating Based on the metric level $l_M(i)$ of each note, an inverse rating is computed to emphasize the number of notes on strong metric positions:

$$\chi_{\text{MetricRating}} = \frac{1}{N} \sum_{i=1}^N \frac{1}{l_M(i)} \quad (8.32)$$

High values of $\chi_{\text{MetricRating}}$ indicate that the majority of all notes of an instrument track are played on strong metric positions and therefore correspond to low metric levels.

Note Density The number of notes in each of the N_B bars is stored in $N_{NB} \in \mathbb{Z}^{N_B}$. The mean and standard deviation are computed as features over N_{NB} in order to measure the average note density and the fluctuation of the note density over time.

On-beat Accentuation In Western European music genres, most notes are located at on-beat positions. In contrast, in Latin American music styles, notes are often played on off-beat positions due to *note syncopation*, as will be explained in the next paragraph. To measure the degree of on-beat accentuation, different metric levels $l \in \{0, 1, 2, 3, 4\}$ are investigated. Given the most common $\frac{4}{4}$ time signature, these metric levels correspond to the note durations $D_l \in \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$. For each metric level l and each bar b , the ratio between the number notes $n_{NB, On, l}(b)$ that are mapped to on-beat positions and the total number of notes $n_{NB}(b)$ in that bar are computed.

$$\delta_{On, l}(b) = \frac{n_{NB, On, l}(b)}{n_{NB}(b)} \quad (8.33)$$

The mean and standard deviation over each $\delta_{On, l}$ are computed as features to capture the degree of on-beat accentuation as well as its fluctuation over time for all metric levels $l \in \{0, 1, 2, 3, 4\}$.

In Figure 8.3, the salsa bassline from the previous note example is shown again. A metric level of $l = 3$ (sixteenth note) is used as a reference. Notes on on-beat positions are indicated by a plus sign (+) and notes on off-beat positions are indicated by an accent sign ('). Here, the number of notes located on on-beat positions are $n_{NB,On,3}(1) = 5$ for the first bar and $n_{NB,On,3}(2) = 2$ for the second bar.



Figure 8.3: Same bassline as in Figure 8.3 is shown. The dominant metric level is $l = 3$, the corresponding sixteenth notes are shown in the second row. Notes on on-beat positions are indicated by a plus sign (+) and notes on off-beat positions are indicated by an accent sign (').

Syncopated Note Sequences Syncopated note sequences are characterized by notes that are shifted from strong metric positions (on-beats) to weak metric positions (off-beats). Rhythms with syncopation can be found especially in Latin American music genres such as salsa or bossa nova. Examples for note sequences with and without syncopation are given in Figure 8.4.

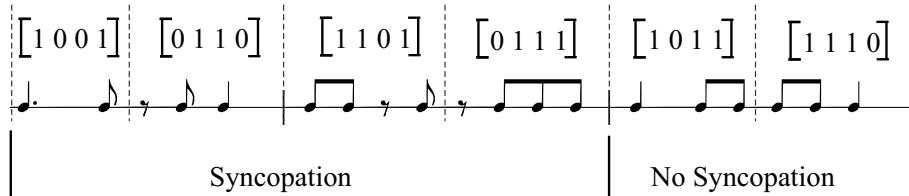


Figure 8.4: Syncopated and non-syncopated note sequences for the metric level $l = 2$ (eight-notes) [10]. Corresponding binary beat sequences are shown above the score. The dotted lines are added for visualization purpose.

Syncopated note sequences are detected as follows. First, each note of a bassline is quantized towards the closest beat positions of a given metric level l as explained in Section 8.1.2. Then, a binary value $\delta_{\text{Beat}}(b) \in [0, 1]$ is assigned to each beat as follows:

$$\delta_{\text{Beat}}(b) = \begin{cases} 1 & , \text{ if at least one note was quantized to that beat and} \\ 0 & , \text{ otherwise.} \end{cases} \quad (8.34)$$

In Figure 8.5, the previously shown salsa bassline is encoded as binary beat sequence for the metric level $l = 3$ (sixteenth-note level).

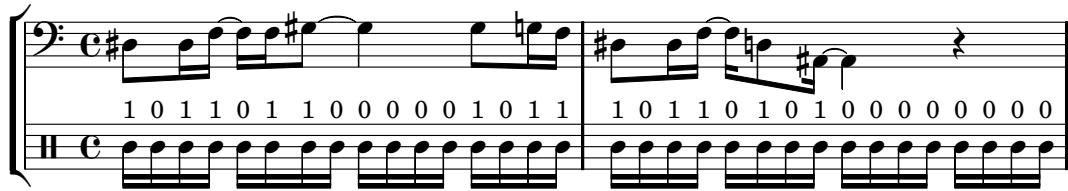


Figure 8.5: Previously shown salsa bassline encoded as binary beat sequence based on sixteenth-notes (metric level $l = 3$).

As shown in Figure 8.4, syncopated note sequences are indicated by the following binary beat-sequences $\delta_{\text{Sync},k} \in \mathbb{Z}^4$:

$$\begin{aligned}\delta_{\text{Sync},1} &= [1, 0, 0, 1] \\ \delta_{\text{Sync},2} &= [0, 1, 1, 0] \\ \delta_{\text{Sync},3} &= [1, 1, 0, 1] \\ \delta_{\text{Sync},4} &= [0, 1, 1, 1]\end{aligned}\tag{8.35}$$

The number of beat sequences that occur in a bassline for a given metric level l is counted as $N_{\text{Sync}}(l)$. Syncopated note sequences are retrieved for the metric levels $l \in \{1, 2, 3, 4\}$.

The degree of syncopation is computed as the ratio between the number of syncopated note sequences $N_{\text{Sync}}(l) \in \mathbb{Z}$ and the overall number of beats in a bassline for each metric level:

$$\chi_{\text{Sync}}(l) = \frac{N_{\text{Sync}}(l)}{N_B d l}.\tag{8.36}$$

8.1.3 Structure

Representations



Figure 8.6: Funk bass pattern example.

Bass Patterns & Subpatterns A bass pattern Θ is a (repetitive) sequence of notes played by the bass instrument. In the following, each bass pattern is represented either using the relative onset times \mathcal{O} (denoted as $\Theta_{\mathcal{O}}$) or the absolute pitch values \mathcal{P} (denoted as $\Theta_{\mathcal{P}}$) of the notes it contains. As an example, the funk bass pattern shown in Figure 8.6 can be represented as

$$\begin{aligned}\Theta_{\mathcal{O}} &= [0, \frac{4}{16}, \frac{6}{16}, \frac{7}{16}, \frac{14}{16}, \dots] \text{ or} \\ \Theta_{\mathcal{P}} &= [48, 55, 57, 58, 48, \dots].\end{aligned}$$

Rhythmic and Tonal Similarity between Patterns Given two patterns Θ_m and Θ_n based on one of the abovementioned representations, the Edit distance $d_E(\Theta_m, \Theta_n)$ can be used to compute their similarity. The Edit distance between two character strings is the accumulated number of character insertions, deletions, and replacements, which are necessary to convert one string into the other [75]. A similarity value can be defined as

$$S_E(\Theta_m, \Theta_n) = 1 - \frac{d_E(\Theta_m, \Theta_n)}{d_{E,\max}} \quad (8.37)$$

with $S_E(\Theta_m, \Theta_n) \in \mathbb{R}$ and $0 \leq S_E(\Theta_m, \Theta_n) \leq 1$.

The scaling factor $d_{E,\max}$ is the number of notes in the longer pattern.

Feature Extraction

Tonal & Rhythmic Similarity between note sequences In order to describe the repetitive structure of a bass pattern, it can be segmented into (non-overlapping) subpatterns of l_{sub} bars length. Depending on the subpattern representation (Θ_P or Θ_O), the rhythmic similarity $S_{E,Rhy}$ and tonal similarity $S_{E,Ton}$ between adjacent subpatterns is computed using the Edit distance as described in the previous section.

The tonal and rhythmic similarity between adjacent subpatterns is computed and averaged over all subpattern pairs to derive the features $\chi_{\text{TonSim}}(l_{\text{sub}})$ and $\chi_{\text{RhySim}}(l_{\text{sub}})$. Both features are computed for the three subpattern lengths $l_{\text{sub}} \in \{1, 2, 4\}$ since these lengths appear most frequently in the dataset. These features for instance allow to discriminate between simple minimal techno bass patterns, which are repeated every bar and more complex funk basslines, which are repeated every 4 bars.

8.2 Data Sets

8.2.1 IDMT-SMT-BASS-GENRE-MIDI

Since 2009, a data set of MIDI files was assembled that contains typical bass patterns from various music genres. The extraction of repetitive bass patterns from original bass lines is not within the scope of this thesis. MIDI files were used in order to omit the error-prone transcription step discussed in Part I of this thesis. The dataset was revised over time (versions V1 [12] and V2 [5]) until a final set (V3) including 520 basslines from 13 music genres was set up and used for the genre classification experiments described in this part of the thesis as well as in [10].

Table 8.6 gives an overview over all three versions w.r.t to the number of basslines and the number of covered music genres. For each music genre, the regional origin and the approximate time of the first genre audio recordings are provided as additional information. As it can be seen, the music genres included in the data set cover a large geographic and historic selection of music cultures. The distribution of tempo values was shown before in Figure 8.1.

The first two version V1 and V2 comprised basslines from 6 and 8 different genres, respectively. These basslines have been entirely taken from instructional bass literature [150, 182]. For the third version V3, the collection was extensively revised. All genre-sets were modified and five new sets

Table 8.6: Music genres included in the three versions (V1, V2, and V3) of the *IDMT-SMT-BASS-GENRE-MIDI* data set. For each music genre, the regional origin, the approximate time of first audio recordings, and the abbreviation as used in this thesis are given.

Music genre	Origin	First rec.	Abbr.	Number of basslines		
				V1	V2	V3
africa	(various)	-	AFR	-	40	-
blues	USA	1912	BLU	50	40	40
bossa nova	Brasil	1958	BOS	-	-	40
funk	USA	1960s	FUN	50	40	40
forró	Brasil	1900	FOR	-	-	40
hip-hop	USA	1970s	HIP	-	-	40
latin	(various)	-	LAT	50	-	-
metal & hard rock	GB, USA	1968	MHR	50	-	-
minimal techno	USA, Germany	1994	MIN	-	-	40
motown	USA	1960	MOT	-	-	40
nineties rock	USA, GB	1990s	RON	-	-	40
pop	GB, USA, others	1960s	POP	50	-	-
reggae	Jamaica	1960s	REG	-	40	40
rock	GB, USA	1960s	ROC	-	40	-
salsa & mambo	Cuba	1930s	SAL	-	40	40
seventies rock	USA, GB	1970s	ROS	-	-	40
soul & motown	USA	1940 / 1960s	SOU	-	40	-
swing	USA	1920s	SWI	50	40	40
zouglou	Cote d'Ivoire	1995	ZOU	-	-	40
				Σ	300	320
						520

were added. Since instructional literature is not available for some of the music genres, various basslines have been manually transcribed from real audio recordings and were added to the data set. The audio recordings associated to each genre were selected by a musicologist. Throughout this thesis, the third version V3 will be referred to as *IDMT-SMT-BASS-GENRE-MIDI*. The dataset was not published due to copyright reasons.

9 Evaluation

9.1 Automatic Music Genre Classification based on Repetitive Bass Patterns

The experiments described in this section were conducted in collaboration with Hanna Lukashevich (Semantic Music Technologies group, Fraunhofer IDMT) and Paul Bräuer (piranha womex AG, Berlin). The results were partly published in [10].

Motivation & Goals

Despite the large number of musical instruments that are played in different music genres, a bass instrument such as the bass guitar or the double bass is almost always present in a musical ensemble [186]. The bassline provides both a rhythmic and harmonic foundation for the remaining instruments.

As discussed before, basslines can be represented in a compact form as repetitive *bass patterns*. This procedure has two advantages. First, a bass pattern is a robust representation of a repetitive bassline since it neglects local rhythmic and tonal variations. Second, bass patterns usually have a length of 2, 4, or 8 bars. Hence, the extraction of score-based features such as presented in Section 8.1 can be accelerated considerably.

The main goal of this experiment is to investigate, whether the music genre of a musical piece can be automatically retrieved from its underlying bass patterns. Three different paradigms will be compared for this classification task:

1. classification based on statistical pattern recognition,
2. classification based on a rule-based decision tree, and
3. classification based on the similarity between bass patterns.

Dataset

For this experiment, the *IDMT-SMT-BASS-GENRE-MIDI* data set described in Section 8.2 was used. The dataset contains MIDI files of 520 bass patterns—40 bass patterns for each of the 13 music genres from different cultural, historical, and regional origins.

Baseline Experiment

As a baseline experiment, the *jSymbolic* software [120] published by McKay and Fujinaga in [122] was used to extract score-based audio features from all 520 basslines. 22 features as listed in Table 9.1 were selected and extracted for each bassline in the data set.

As classifiers, a SVM with RBF kernel function was used and a 20-fold cross-validation was performed as explained in Section 2.3 without any prior feature selection or feature space transformation. A mean class accuracy of 49.6 % was achieved over 13 genre classes, the confusion matrix is shown in Table 9.2.

Table 9.1: Selected audio features from the *jSymbolic* software that were extracted for the baseline experiment. Details can be found in [122].

Number	Feature Name
1	Average Melodic Interval
2	Average Note Duration
3	Chromatic Motion
4	Initial Tempo
5	Maximum Note Duration
6	Melodic Fifths
7	Melodic Octaves
8	Melodic Thirds
9	Melodic Tritones
10	Minimum Note Duration
11	Most Common Melodic Interval
12	Most Common Pitch Class
13	Note Density
14	Number of Strong Pulses
15	Pitch Class Variety
16	Pitch Variety
17	Range
18	Repeated Notes
19	Rhythmic Variability
20	Stepwise Motion
21	Variability of Note Duration
22	Pitch Class Distribution

Experimental Procedure

The experimental procedure of the main experiment is illustrated in Figure 9.1. From the MIDI files containing the bass patterns, the note parameters absolute pitch \mathcal{P} , note onset \mathcal{O} , and note duration \mathcal{D} are extracted. For each bass pattern, all score-based audio features described in Section 8.1 are extracted and concatenated to a feature vector χ .

Classification based on Pattern Recognition

The first classification paradigm is the most commonly used classification approach in MIR. Here, a SVN classifier model as explained in Section 2.3.3 was used. The evalation was performed using

Table 9.2: Confusion matrix for genre classification using SVM classifier and the *jSymbolic* audio features. All values are given in percent. The mean class accuracy is $\bar{A} = 49.6\%$. The music genre abbreviations are explained in Table 8.6.

	BLU	BOS	FOR	FUN	HIP	MIN	MOT	REG	RON	ROS	SAL	SWI	ZOU
BLU	60	5	0	0	2.5	0	7.5	5	5	5	5	5	0
BOS	0	70	12.5	0	0	5	5	2.5	0	2.5	2.5	0	0
FOR	0	15	65	0	0	0	2.5	0	0	10	5	2.5	0
FUN	2.5	0	0	50	10	2.5	15	0	15	0	5	0	0
HIP	7.5	2.5	2.5	17.5	37.5	10	0	10	2.5	2.5	5	0	2.5
MIN	5	5	0	5	7.5	57.5	7.5	5	5	0	2.5	0	0
MOT	10	2.5	0	17.5	0	15	17.5	2.5	7.5	12.5	10	0	5
REG	0	2.5	2.5	2.5	2.5	5	2.5	35	7.5	20	10	0	10
RON	7.5	0	0	12.5	5	7.5	5	17.5	30	2.5	7.5	2.5	2.5
ROS	12.5	2.5	0	7.5	5	5	17.5	10	10	22.5	7.5	0	0
SAL	0	5	0	7.5	5	10	5	2.5	10	0	52.5	0	2.5
SWI	0	0	2.5	0	0	2.5	0	0	0	2.5	0	92.5	0
ZOU	0	0	0	0	7.5	2.5	12.5	5	2.5	15	0	0	55

a 52-fold stratified cross-validation.

Classification based on a Decision Tree

The Classification and Regression Tree (CART) algorithm introduced in Section 2.3.3 is used in this experiment with a subsequent optimal pruning strategy as proposed in [26]. The optimal parameters for the stopping rules such as for instance a minimum number of items per node to be still considered for splitting are determined experimentally. The generalization properties of the decision tree are controlled in a cross validation scenario, where the tree is pruned to a certain level in order to prevent overfitting the training data.

Classification based on Bass Pattern Similarity

The third classification paradigm is based on the assumption that bass patterns from the same music genre share common rhythmic and tonal properties and therefore show a certain degree of similarity among each other. The tonal and rhythmic representations of bass patterns as introduced in Section 8.1.3 are applied here.

First, the problem of transposition and its effect on the tonal similarity between bass patterns will be discussed. Second, different similarity measures will be derived from the Edit distance and pair-wise note similarities. Finally, different aggregation strategies will be presented, which are evaluated and compared in the genre classification experiment.

Transposition If a melody is transposed, the absolute pitch values of all notes in the melody are shifted by a constant value. Transposition allows to notate a melody in a different musical key. At the same time, transposition does not affect the interval structure between the notes and therefore should not have an influence on the tonal similarity.

In order to compensate for a potential pitch transposition between two bass patterns $\Theta_{P,1}$ and

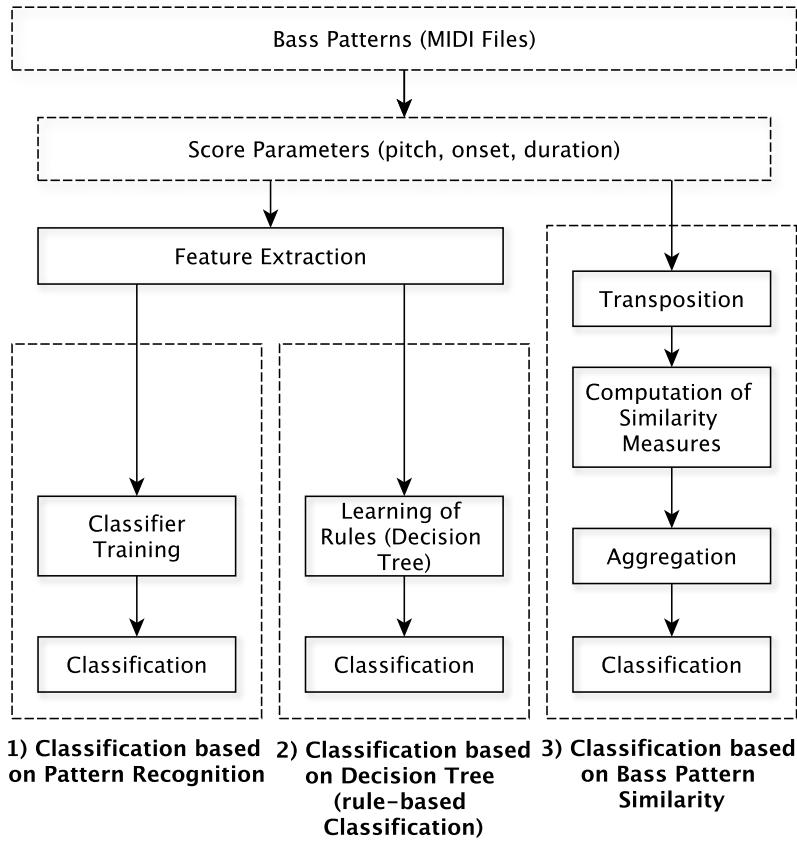


Figure 9.1: Three different classification paradigms evaluated for automatic genre classification based on repetitive bass patterns [10]—classification based pattern recognition (SVM classifier), classification based on a rule-based decision tree, and classification based on similarity between bass patterns.

$\Theta_{\mathcal{P},2}$, the dominant pitch values $\mathcal{P}_{\text{dom},1}$ and $\mathcal{P}_{\text{dom},2}$ of both patterns are aligned as

$$\mathcal{P}_2(i) \leftarrow \mathcal{P}_2(i) + \mathcal{P}_{\text{dom},1} - \mathcal{P}_{\text{dom},2} \quad (9.1)$$

before the tonal similarity measures are computed. The notion of transposition between patterns requires that both patterns share a certain amount of similarity. \mathcal{P}_2 denotes the absolute pitch values, and N_2 the number of notes in the second pattern.

Similarity Measures Based on Edit Distance Between Bass Patterns Four different similarity measures are derived from the Edit distance S_E and used in the classification experiments—a tonal similarity measure $S_{E,T}$, a rhythmic similarity measure $S_{E,R}$, as well as two combined

similarity measures $S_{E,RT,mean}$ and $S_{E,RT,max}$:

$$S_{E,T} = S_E(\Theta_{P,1}, \Theta_{P,2}) \quad (9.2)$$

$$S_{E,R} = S_E(\Theta_{O,1}, \Theta_{O,2}) \quad (9.3)$$

$$S_{E,RT,mean} = \frac{1}{2} (S_{E,T} + S_{E,R}) \quad (9.4)$$

$$S_{E,RT,max} = \max(S_{E,T}, S_{E,R}) \quad (9.5)$$

Similarity Measures Based on a Pairwise Distance Between Bass Patterns The *pairwise similarity* between two bass patterns is computed as follows: First, the number of notes $N_{1,2}$ in pattern Θ_1 are counted, for which at least one note in pattern Θ_2 exists with the same absolute pitch or the same note onset, respectively. $N_{2,1}$ is compute vice versa. The pairwise similarity is computed as

$$S_P(\Theta_1, \Theta_2) = \frac{1}{2} \left(\frac{N_{1,2}}{N_1} + \frac{N_{2,1}}{N_2} \right). \quad (9.6)$$

Five different similarity measures derived from the pairwise similarity S_P are investigated in the experiments:

$$S_{P,T} = S_P(\Theta_{P,1}, \Theta_{P,2}) \quad (9.7)$$

$$S_{P,R} = S_P(\Theta_{O,1}, \Theta_{O,2}) \quad (9.8)$$

$$S_{P,RT,mean} = \frac{1}{2} (S_{P,T} + S_{P,R}) \quad (9.9)$$

$$S_{P,RT,max} = \max(S_{P,T}, S_{P,R}) \quad (9.10)$$

The fifth similarity measure $S_{P,RT}$ is computed similar to $S_{P,R}$ and $S_{P,T}$, however, only those notes were counted for $N_{1,2}$ and $N_{2,1}$ that have both the same absolute pitch and the same note onset.

Aggregation Strategies In the previous section, different similarity measures were discussed that allow to compute a tonal or rhythmic similarity between bass patterns. In this section, two strategies are compared to classify the music genre of an unknown bass pattern based on its similarity towards a set of known bass patterns:

- Pattern-wise classification
- Classification via bar-wise aggregation

Pattern-wise Classification Given a dataset of N_G genre classes, each class c is represented by a set of associated bass patterns $\{\Theta_c(i)\}$ with i denoting the pattern index. The number of patterns per genre is $N_P(c)$. Given an unknown pattern Θ , a likelihood measure $L(c|\Theta)$ can be defined as the highest similarity between Θ and all patterns associated to the genre c as

$$L(c|\Theta) = \max_{1 \leq i \leq N_P(c)} S(\Theta, \Theta_c(i)) \text{ for } 1 \leq c \leq N_G. \quad (9.11)$$

Finally, the genre of the unknown pattern is classified by maximizing the genre likelihood measure.

$$\hat{c}(\Theta) = \arg \max_{1 \leq c \leq N_G} L(c|\Theta) \quad (9.12)$$

In case two or more genres are associated with the same likelihood-value, the classified genre is randomly selected from the most likeliest candidates.

In the following, l_Θ denotes the length of a pattern in bars. The *IDMT-SMT-BASS-GENRE-MIDI* dataset contains bass patterns that are between 4 and 16 bars long. The following procedure is applied to compute the similarity between two patterns of arbitrary lengths:

1. If necessary, each pattern is elongated by adding its first bar to achieve a pattern length that equals a power of two ($l_\Theta \equiv 2^i$ with $i \in \mathbb{Z}$).
2. If the lengths of the patterns are not equal, the shorter pattern is shifted across the longer pattern with a step-size of two as shown for an example of two patterns with the lengths $l_{\Theta,1} = 4$ and $l_{\Theta,2} = 8$ in Figure 9.2.
3. For each shift, the similarity between the shorter pattern and the corresponding sub-pattern (indicated by grey filling in the figure) of the longer pattern is computed.
4. Finally, all similarity values are averaged to an overall similarity score $S(\Theta_1, \Theta_2)$.

Figure 9.2 illustrates the proposed comparison approach for two bass patterns, which are 4 and 7 bars long. For the second pattern, the appended eighth bar, which equals the first bar of the pattern, is illustrated using a dashed edge.

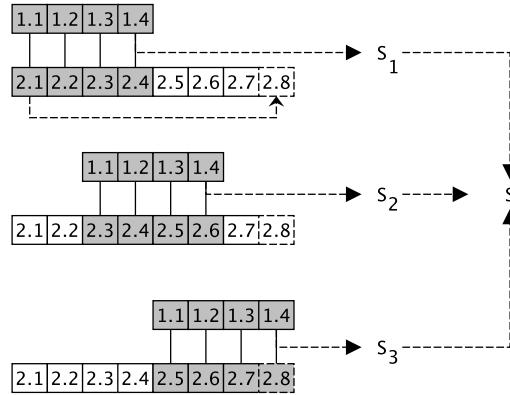


Figure 9.2: Comparing two patterns of different lengths $l_{\Theta,1} = 4$ and $l_{\Theta,2} = 7$. Each square represents a sub-pattern of one bar length. First, the patterns are elongated to a pattern length that is a power of two. Then, the patterns are compared using a hopsize of 2 bars and finally, an average similarity value is computed.

Classification via Bar-wise Aggregation Each pattern Θ can be split into a set of $l_{\Theta,\text{sub}}$ sub-patterns $\{\Theta_{\text{sub}}(i)\}$ of one bar length. Again, given a dataset of N_G genre classes, each genre c is represented by a set of associated sub-patterns $\{\Theta_{\text{sub},c}(k)\}$.

The bar-wise aggregation strategy is based on the similarity between sub-patterns. Hence, given an unknown pattern Θ , the likelihood measure $L(c|\Theta_{\text{sub}}(i))$ between its i -th sub-pattern and the genre class c is computed as

$$L(c|\Theta_{\text{sub}}(i)) = \max_{1 \leq k \leq l_{\Theta,\text{sub},c}} S(\Theta_{\text{sub}}(i), \Theta_{\text{sub},c}(k)), \quad 1 \leq c \leq N_G \quad (9.13)$$

with $l_{\Theta,\text{sub},c}$ denoting the number of (known) sub-patterns assigned to the genre class c . The likelihood measure between the full pattern $L(c|\Theta)$ is computed by averaging the likelihood measures of all of its sub-patterns for each genre class as

$$L(c|\Theta) = \frac{1}{l_{\Theta,\text{sub}}} \sum_{i=1}^{l_{\Theta,\text{sub}}} L(c|\Theta_{\text{sub}}(i)) \quad (9.14)$$

The final genre classification of the pattern is performed as previously shown in (9.12).

Results & Summary

Comparison of Classification Approaches based on Bass Pattern Similarity

Table 9.3: Performance of different configurations of similarity measures and aggregation strategies in terms of classification accuracy—mean (MN), standard deviation (SD), minimum (MIN), and maximum (MAX) [10]. All values given in percent.

Similarity measure	Aggregation strategy							
	Pattern-wise				Bar-wise			
	MN	SD	MIN	MAX	MN	SD	MIN	MAX
$S_{E,T}$	26.0	15.5	3.8	51.0	9.0	12.6	0	40.0
$S_{E,R}$	33.5	22.3	10.7	93.3	16.3	18.9	1.7	73.3
$S_{E,RT,\text{mean}}$	37.9	24.6	7.5	97.5	13.1	17.2	0	58.3
$S_{E,RT,\text{max}}$	35.8	22.0	13.2	93.3	15.7	17.9	1.0	67.8
$S_{P,T}$	21.7	8.5	7.9	35.4	7.8	12.5	0	44.5
$S_{P,R}$	33.5	22.9	12.4	93.3	16.1	19.7	0	75.3
$S_{P,RT}$	29.9	17.6	5.4	57.5	12.0	21.0	0	75.8
$S_{P,RT,\text{mean}}$	38.9	23.9	7.5	97.5	14.4	25.5	0	92.5
$S_{P,RT,\text{max}}$	38.6	23.6	10.0	97.5	12.1	23.1	0	85.0

In order to identify the optimal configuration of similarity measure and aggregation strategy, a 10-fold cross-validation experiment was performed using the *IDMT-SMT-BASS-GENRE-MIDI*

dataset. Table 9.3 shows the mean, standard deviation, minimum, and maximum over all class-wise accuracy values (denoted by MN, SD, MIN, and MAX) for all configurations. The following conclusions can be drawn:

1. The pattern-wise classification outperformed the bar-wise classification approach. Hence, full (repetitive) bass patterns seem to better represent the music genre than their sub-patterns.
2. The pair-wise similarity measures and the similarity measures based on the edit distance performed comparably well in terms of mean class accuracy.
3. The similarity measures that combine the tonal and rhythmic similarity between bass patterns achieved the highest mean accuracy values. In particular, the similarity measure $S_{P,RT,Mean}$ that computes the mean between the tonal and rhythmic pair-wise similarity performed best with 38.9 % of mean accuracy.
4. A large variance among the classification performance for the 13 genres can be observed. Given the optimal configuration, the mean accuracy values strongly varied from 97.5 % for swing down to 7.5 % for seventies rock (compare upper confusion matrix in Table 9.4). Satisfying results over 60 % accuracy were only achieved for the genres swing, blues, and bossa nova. The reason for that could be that these genres are well-characterized by only a few typical *prototypical* bass-patterns, which are typically used with only minor tonal and rhythmic variations. The bass patterns associated to other genres seem to be too diverse in order to be classified appropriately using this approach.

Performance Comparison of Different Classification Approaches

Table 9.4 summarizes the confusion matrices for the best configuration of each of the three proposed classification approaches. The following conclusions can be drawn:

1. The rule-based classification approach achieved a mean class accuracy of 64.8 % and outperformed classification based on pattern similarity and statistical pattern recognition by 25.9 % and 9.4 %, respectively. The random classification accuracy baseline for 13 classes is around 7.7 %.
2. The SVM classifier achieved a mean class accuracy of 55.4 % using the proposed feature set. In the baseline experiment, the feature set extracted using the *jSymbolic* software could only achieve 49.6 % of mean class accuracy. However, when comparing the confusion matrices in Table 9.2 and Table 9.4, similar results can be observed for the different genre classes. For instance, both classifiers perform comparably well for the genres blues (BLU), bossa nova (BOS), funk (FUN), minimal techno (MIN), swing (SWI), and zouglo (ZOU). At the same time, the classification performance for motown (MOT), reggae (REG), seventies rock (ROS), and nineties rock (RON) is rather poor.
3. The class accuracy values strongly vary over all 13 classes, especially for the classification based on pattern similarity and statistical pattern recognition. Low classification rates strongly correspond to *eclectic styles* [10], i.e. music styles, which by their own history and logic are a camouflage of older styles. In bossa nova (BOS) for example, Jazz and Latin

styles – especially samba – are blended. However, the two styles are still recognizable. The results confirm the ambiguous nature of the music genre classification task [163].

4. Many confusions between music genres can be explained from a musicological point of view. The observed genre confusions can be separated into *corresponding confusions* and *non-corresponding confusions* [10]. As regards the non-corresponding confusions, the classification is simply mistaken. Some confusions on the other hand correspond to, i.e., reflect, ambiguities in the musical reality. For example basslines of blues music have been mistaken as rock (ROS, RON) or motown (MOT), but almost never as one of the four different Latin or Afro-Caribbean music styles (SAL, ZOU, BOS, FOR). Other than mistakes or misclassification, these highly corresponding confusions are valuable information in a lot of use cases such as musicological research on genre similarity or distinguished classification of fusion styles for online shops and distributors.
5. The performance of the SVM classifier is in a lot of cases but not always worse than the pruned-tree approach. Regarding the harmonically more complex genre of bossa nova, the SVM approach even beats the pruned tree by 10% of accuracy. While the pruned tree approach has the best classification rates, it is worst regarding the rate of non-corresponding confusions.

Performance Comparison to the State-of-the-art

As discussed in Section 7.1, no comparable classification experiment with 13 genres was presented in the literature. Tsunoo et al. amongst others performed a genre classification experiment with 10 classes on audio data by comparing bass patterns. In [171], the authors report a genre classification accuracy of 39.8 % by solely using bass patterns for the GTZAN dataset that includes songs from the 10 music genres blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock.

In order to compare the proposed algorithm with this publication, a reduced version of the *IDMT-SMT-BASS-GENRE-MIDI* was created by removing bass patterns of the three Latin American music genres forró, salsa-mambo, and zouglo. Using the same set of features as before, a mean class accuracy of 56.5 % was achieved. The two results indicate that genre classification based on transcribed bass patterns (given as MIDI files) results in better accuracy values than audio-based genre classification. However, the two datasets do not cover the same music genres. As discussed before, the bassline is a discriminative property for many music genres. At the same time, some music genres can only be classified using other properties such as rhythm or timbre.

Table 9.4: Genre classification confusion matrices for the three classification paradigms. All values in the confusions matrices are given in percent. The decision tree approach outperformed the classification using SVM and classification based on pattern similarity by 9.4 % and 25.9 % in mean class accuracy \bar{A} . It can be observed that the classification performance strongly depends on the music genre.

Classification based on bass pattern similarity: $\bar{A} = 38.9 \%$														
BLU BOS FOR FUN HIP MIN MOT REG RON ROS SAL SWI ZOU														
BLU	60	2.5	0	0	2.5	5	7.5	2.5	10	2.5	0	7.5	0	
BOS	5	67.5	15	0	0	0	2.5	0	2.5	7.5	0	0	0	
FOR	5	12.5	40	2.5	2.5	12.5	5	0	5	2.5	2.5	10	0	
FUN	25	0	2.5	32.5	2.5	2.5	5	2.5	10	5	7.5	0	5	
HIP	12.5	5	5	12.5	15	12.5	7.5	10	12.5	0	0	2.5	5	
MIN	6.3	0	1.3	7.5	10	36.3	6.3	5	17.5	2.5	0	5	2.5	
MOT	26.7	0	0	5	2.5	0	27.5	6.7	8.3	18.3	2.5	0	2.5	
REG	12.5	0	5	2.5	7.5	2.5	17.5	25	5	7.5	2.5	2.5	10	
RON	27.5	4.2	0	1.7	2.5	5	2.5	5	24.2	20	2.5	5	0	
ROS	5	7.5	5	7.5	0	7.5	22.5	10	17.5	7.5	0	7.5	2.5	
SAL	10	0	4.2	12.5	2.5	6.7	10	2.5	2.5	7.5	36.7	0	5	
SWI	2.5	0	0	0	0	0	0	0	0	0	0	97.5	0	
ZOU	5	0	2.5	6.3	11.3	8.8	5	5	15	0	5	0	36.3	
Classification using a rule-based decision tree (CART): $\bar{A} = 64.8 \%$														
BLU BOS FOR FUN HIP MIN MOT REG RON ROS SAL SWI ZOU														
BLU	70	0	0	5	0	0	10	7.5	5	0	2.5	0	0	
BOS	0	67.5	12.5	0	0	5	2.5	0	0	2.5	0	0	10	
FOR	2.5	10	77.5	0	0	0	2.5	0	0	0	5	0	2.5	
FUN	2.5	0	0	77.5	0	2.5	10	0	0	2.5	2.5	0	2.5	
HIP	7.5	2.5	0	5	50	0	0	10	0	2.5	12.5	2.5	7.5	
MIN	5	0	0	0	0	60	7.5	0	2.5	5	10	0	10	
MOT	5	0	0	0	0	0	72.5	2.5	2.5	10	2.5	0	5	
REG	10	2.5	0	2.5	2.5	0	2.5	55	0	0	7.5	0	17.5	
RON	17.5	7.5	0	0	0	0	5	17.5	35	5	10	0	2.5	
ROS	10	0	0	5	0	0	5	7.5	12.5	45	10	0	5	
SAL	5	0	0	5	0	0	0	2.5	2.5	2.5	65	0	17.5	
SWI	0	0	0	0	0	0	2.5	0	2.5	0	2.5	92.5	0	
ZOU	2.5	0	0	2.5	2.5	0	5	2.5	0	5	5	0	75	
Classification using an SVM classifier: $\bar{A} = 55.4 \%$														
BLU BOS FOR FUN HIP MIN MOT REG RON ROS SAL SWI ZOU														
BLU	60	0	0	2.5	2.5	2.5	10	2.5	7.5	7.5	0	5	0	
BOS	0	77.5	17.5	0	0	0	0	0	0	2.5	2.5	0	0	
FOR	0	25	57.5	0	0	5	5	0	0	2.5	2.5	0	2.5	
FUN	5	0	0	67.5	2.5	0	5	7.5	7.5	2.5	2.5	0	0	
HIP	2.5	0	2.5	0	45	17.5	5	7.5	0	10	7.5	0	2.5	
MIN	0	0	2.5	0	12.5	67.5	0	2.5	5	5	5	0	0	
MOT	17.5	0	2.5	12.5	2.5	7.5	20	2.5	7.5	17.5	5	0	5	
REG	7.5	2.5	0	7.5	10	0	7.5	40	10	2.5	0	0	12.5	
RON	5	2.5	2.5	12.5	0	10	12.5	10	37.5	5	0	2.5	0	
ROS	10	2.5	0	5	10	2.5	15	10	7.5	27.5	2.5	2.5	5	
SAL	0	2.5	5	2.5	12.5	2.5	2.5	2.5	5	0	55	0	10	
SWI	2.5	0	0	0	2.5	0	0	0	2.5	0	92.5	0		
ZOU	0	0	0	2.5	7.5	0	7.5	0	2.5	7.5	0	72.5		

10 Summary

As shown in this part of the thesis, the bassline itself is a successful discriminator among different music genres. An algorithm that combines score-based audio features extracted from the bassline with a rule-based classification strategy achieved a mean class accuracy of 64.8 % for 13 genre classes. The rule-based approach facilitates the analysis of the genre classification results by musicologists since the automatically extracted rules are simple feature-value relationships that can easily be interpreted. This approach can forge a bridge between automatic analysis methods as presented here and their application for musicological research. The automatic analysis of large datasets can reveal new insights into musical properties of music genres.

By reducing the dataset to 10 music genres, a mean class accuracy of 56.5 % could be achieved using the proposed set of features in combination with a SVM classifier. A state-of-the-art system for genre classification that analyzes bass patterns in audio files achieves a mean accuracy of 39.8 % on a dataset with the same number of music genres [171]. Even though the genres in both datasets are not completely the same, the results indicate that a genre classification system that investigates the bass line of a song can achieve higher performance, if features can be extracted on a transcription of the repetitive bass patterns instead of a spectrogram representation of the audio signal.

The dataset used in the genre classification experiments does not contain playing technique annotation since it was assembled from various sources with different amounts of annotations as explained in Section 8.2. Especially the applied plucking styles can be a useful indicator towards the music genre and should be incorporated in future research. For instance, the slap techniques *slap-thumb* and *slap-pluck* are typically used in funk music, the *picked* style is typically used in rock and metal related genres.

In the future, a *hybrid classification approach* that combines the advantages of all three discussed classification paradigms could be investigated. When source separation and music transcription allow to robustly transcribe different instrument track from a mixed audio signal, score-based features could be extracted on a track level allowing to obtain genre classification results that are associated to the individual instruments. This *multiple-expert* genre classification strategy could potentially reveal a more detailed stylistic description of a given piece of music.

Part III

Application for Sound Synthesis & Parametric Audio Coding

Preface

In the past decades, various algorithms were proposed for the *analysis* and *synthesis* of musical instrument recordings. As discussed in the first part of this thesis, automatic music transcription aims to represent musical instrument recordings using a parametric description of all musical note events that were played. The extracted score-level and instrument-level parameters capture the most salient perceptual properties such as loudness, pitch, and timbre.

For the purpose of audio synthesis, *physical modeling* algorithms allow to mimic the sound production of different instrument types. For string instruments, these algorithms rely on a simple one-dimensional model of the string vibration. Various model extensions were proposed in the literature to simulate different playing techniques, the acoustic properties of the instrument body, or coupling effects between multiple vibrating strings.

In this part of the thesis, two main contributions will be presented. First, a novel *physical modeling algorithm* for *bass guitar synthesis* will be described in Section 12.1. It implements 11 different bass guitar playing techniques and can therefore be used to synthesize basslines from a large variety of musical styles. Furthermore, as will be shown in Section 12.1.2, the proposed synthesis algorithm can be tuned towards the sonic properties of a particular bass guitar instrument in order to improve the quality of the re-synthesized basslines.

Second, in Section 12.2, a *parametric audio coding framework* will be presented that combines the analysis algorithms discussed in the first part of this thesis with the bass guitar synthesis algorithm. The coding framework relies on the assumption that the extracted note-wise parameters are sufficient to represent the acoustic properties of the original bass guitar track. In comparison to conventional audio coding schemes, a note-wise parametrization allows to transmit the instrument recording with significantly lower bit-rates. The results of a MUSHRA listening test showed that an audio coding scheme based on the presented algorithms offers a higher perceived sound quality in comparison to conventional coding schemes when set to very low bitrates.

Parts of the following sections have been previously published in [104] and [9].

11 Related Work

After a brief discussion of sampling-based synthesis and virtual instruments in Section 11.1, physical modeling algorithms for string instrument synthesis will be reviewed in Section 11.2 and Section 11.3. Finally, different parametric audio coding approaches will be compared in Section 11.4.

11.1 Digital Music Synthesis

Nowadays, most commercially available synthesis algorithms are based on *sampling*. The sampling of real instruments allows to achieve a high sound quality. At the same time, the required amount of data storage is high since the sampled instrument must be recorded with a large variety of dynamic levels and playing techniques to allow for a naturally sounding re-synthesis. Within the last decade, the constant rise in performance of modern computer hardware with regards to processor performance and storage capacity lead to a wide application of sampling-based algorithms in music production software.

Sound synthesis algorithms are usually encapsulated in *virtual instruments*, which are available as stand-alone software or as plug-ins for commercial Digital Audio Workstations (DAW). Popular examples of virtual bass guitar simulations are BROOMSTICKBASS by Bornemark [25], SCARBEE PRE-BASS by Native Instruments [81], and Trillian by Spectrasonics [159].

Virtual instruments are commonly triggered (or “played”) by using external controlling devices such as a MIDI keyboards. These devices translate physical or haptic gestures like pressing a key into a corresponding control signal that is sent to the synthesis algorithm. Janer proposes three components for a real-time capable virtual musical instrument: the synthesis engine, the input controller, and the mapping interface [83]. Popular digital formats such as MIDI or OSC can be used as protocols to transmit control commands.

However, a major challenge is to develop controller interfaces that are similar to the haptic properties of real instruments in order to create a greater acceptance among musicians [36]. Laurson et al. emphasize that the input parameter for a synthesis algorithm must capture both score parameters (related to “common [music] notation”) as well as instrument specific expressions [107].

In the following section, the *physical modeling* approach for sound synthesis will be detailed. Other synthesis methods such as additive and subtractive sound synthesis, FM synthesis, or wavetable synthesis will not be considered here. The interested reader is referred to [164].

11.2 Physical Modeling

Physical modeling algorithms are based on mathematical and physical models that describe the sound production of musical instruments. Depending on the type of instrument, the proposed models show different complexities. The vibration of strings is commonly implemented using a one-dimensional model. In contrast, vibrating drum membranes are approximated using two-dimensional models [55].

The *digital waveguide* algorithm is commonly applied to simulate the sound production of string instruments [158]. As detailed in Section 2.1.3, d'Alembert found that the vibration of a plucked string can be modeled as a superposition of two waves that propagate in opposite directions across the string and that are reflected at the string ends. Smith proposed in [158] that the traveling-wave solutions can be modeled digitally by sampling the elongation of the vibrating string at fixed geometric positions. The wave propagation into both directions is modeled by two parallel chains of delay units, which are referred to as *delay lines*. The reflection and the damping losses of the string vibration are simulated using digital filters.

The basic waveguide model has several limitations. First, as discussed in Section 2.1.3, the harmonic envelopes are often amplitude-modulated due to string beating or show a non-exponential decay. These phenomena cannot be modeled appropriately using the basic modeling approach described above. Second, non-harmonic signal components such as attack transients, which have a wide-band noise-like characteristic, are hard to include into the model. However, these signal components are characteristic to the timbre of musical instruments.

11.3 Synthesis Model Extensions and Calibration

In order to model expressive playing on string instruments, various extensions of the basic digital waveguide algorithm are presented in the literature. As discussed in Section I, the goal of the parametrization process is to find a compact representation of a musical instrument recording. In order to allow a realistic sound synthesis, the synthesis algorithm must be able to reproduce the most important timbre variations of the instrument that result from using different playing techniques.

In this section, the calibration of sound synthesis models by means of automatic parameter estimation will be discussed. Most of the reviewed publications focus on the guitar. However, due to the similar sound production, these proposed synthesis algorithms are also relevant for bass guitar synthesis. Erkut describes the model calibration process as the “inverse problem to physical modeling” [50]. The first step is to identify “physical correlates to complex performance gestures” [162]. These correlates such as the plucking force of a guitarist playing a note on a string determine the signal parameters that are to be automatically estimated.

Harmonic Envelopes & Inharmonicity

The first significant parameter that affects the synthesis is the decay time of the note envelope. Plucking styles such as muted play or finger-style play allow musicians to vary the amount of string damping. The correlate within the synthesis algorithm is the gain of the loop filter, which is usually aligned according to the estimated slope of the note envelope. Erkut et al. propose a

calibration scheme for a guitar synthesis algorithm based on pitch-synchronous STFT of recorded notes [50]. The harmonics are individually modeled as “time-varying sinusoidal components with individual magnitude, frequency, and phase trajectories”. Laurson et al. argue that the synthesis of fast note sequences requires a different excitation function to get a brighter plucking sound and a lower degree of damping [108].

The inharmonic relationship between the harmonic frequencies, which is often characteristic for string instruments, affects the instrument timbre as discussed in Section 2.1.3. Inharmonicity is commonly incorporated in the synthesis models to achieve a more natural sound as for instance in [178] and [177].

Plucking Styles

The synthesis model is usually initialized with characteristic excitation functions in order to mimic different plucking styles. The excitation function can be extracted for instance by inverse filtering a recorded note with the synthesis model [91, 176]. Lee et al. compare further approaches to estimate the excitation function in [109].

Musicians most often use their fingers or a plastic plectrum to play a string instrument. Cuzzuoli and Lombardo propose to separately model three phases during the finger-string interaction while the string is plucked: excitation, release, and damping [37]. As explained in Section 2.1.2, the finger-string interaction is modeled as two consecutive gestures in this thesis—the plucking style and the expression style. Germain & Evangelista model the plectrum-string interaction in [63], which is characteristic for the plucking style *picked*. In [113], Lindroos et al. propose a three-part excitation function that allows to simulate different dynamic levels, different plucking angles of the plectrum, two-stage note decay, and the effect of magnetic pickups on the electric guitar.

For the bass guitar, the two techniques slap-thumb and slap-pluck are very common. As explained in Section 2.1.2, either the thumb of the playing hand is hammered on the string or the string is plucked very strongly. Rank and Kubin [146], Trautmann & Rabenstein [168], as well as Janer et al. [83] propose synthesis algorithms to simulate the (characteristic) collision between the string and the fretboard that is caused by the large string elongation. In a similar fashion, models for the collision between strings and the fretboard of the guitar [53] and for the interaction between the player’s fingers and the string [37] are proposed in the literature.

Expression Styles

Concerning the expression techniques listed in Table 2.1, guitar synthesis models were proposed that incorporate frequency modulation techniques such as vibrato [107], harmonics [135], and dead-notes [53]. As discussed in Section 2.1.2, the expression styles bending, vibrato, or slide result in a time-varying fundamental frequency of the played note. Due to the spatial sampling of the vibrating string, only a limited number of different fundamental frequency values can be simulated using the basic waveguide model. This problem is solved by using additional *fractional delay filters* in cascade to the delay line elements [92]. Karjalainen et al. shows in [93] that both delay lines of the digital waveguide model can be reduced to a *single delay-loop* model as illustrated in Figure 11.1. The delay line z^{-D} and the fractional delay filter $H_F(z)$ are cascaded with a loop

filter $H_L(z)$. The magnitude response of $H_L(z)$ is designed to best simulate the energy decay rates of the guitar note harmonics.

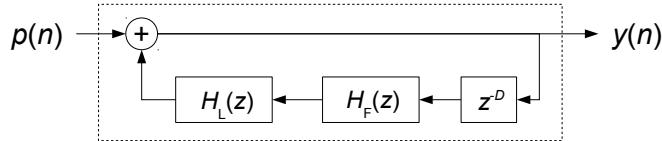


Figure 11.1: Single delay-loop model for physical modeling of string instruments consisting of a delay line z^{-D} , a fractional delay filter $H_F(z)$, and a loop filter $H_L(z)$.

The synthesis model is usually assumed to be a linear time-invariant (LTI) system. This assumption has to be relaxed for the case of time-varying fundamental frequency. However, Erkut et al. report that they found no negative effect in terms of audible artifacts [50].

11.4 Parametric Audio Coding Schemes

In many application scenarios such as audio analysis, transmission, or synthesis, a compact representation of audio signals is required. The main goal of *parametric audio coding* is to compress an audio signal by describing it using a suitable set of parameters. These parameters capture the most important perceptual properties of the original signal. *Hybrid audio coding schemes* exist that transmit the parametric information along with the conventionally encoded audio data in order to improve the decoding result. Two popular examples are the spectral band replication (SBR), where the upper frequency range is reconstructed based on the encoded lower frequency range, or MPEG Surround, where only a stereo signal is conventionally coded and the interchannel correlations between the stereo channels and the surround channels are transmitted as parameters. In this thesis, a pure parametric audio coding approach will be followed.

Speech coding is one of the earliest approaches to parametric audio coding. It is based on a source model of the human vocal tract. The vocal cord is modeled with an FIR filter and the excitation signal of the glottis is approximated with pulses or noise. An audio recording of speech can be compressed by using the estimated model parameter as its representation [88]. This way, the audio signal can be stored and transmitted with very low bit-rates. The estimated model parameters are not only used for speech coding but also for speech recognition applications.

For more general audio signals, the aforementioned approach is less effective due to the large variety of musical instrument timbres. Therefore, instrument-specific models of sound production must be applied to efficiently encode and reproduce the recorded sound. For example, Arnold and Schuller propose a parametric instrument codec for guitar recordings in [17]. The extracted parameters are the excitation function, which is influenced by the plucking style, the gain factor, which controls the note decay, the note onset position, the note fundamental frequency, and the loop filter parameters of the waveguide synthesis model. A dataset of monophonic melodies recorded on acoustic and electric guitars was used as stimuli in a MUSHRA listening tests and compared to the audio codecs HE-AAC, Ogg Vorbis, and AMR-WB+. The results show similar preference among all codecs. However, in comparison to the other audio codecs, the proposed parametric codec requires a significantly smaller bit-rate of 2.4 kbit/s.

The most popular digital music description languages are Musical Instrument Digital Interface (MIDI) and MusicXML. However, due to several limitations, they are not suitable to store the full set of note parameters that was proposed in the first part of this thesis. MIDI cannot store the information about the instrument-level parameters playing techniques and fretboard position. MusicXML is also too restrictive to be adapted to the proposed instrument-level parameters. Also, the note loudness information is only stored on a very coarse scale, which is not sufficient to capture expressive dynamic variations on the instrument. In [95], the Hypermedia/Time-based Structuring Language (HyTime) as presented in [64] was identified as a promising alternative to annotated note events in string instrument recordings. Both time and frequency can be annotated on an absolute scale or relative to given reference values. Hence, the fundamental frequency can be described as a time-continuous parameter, which allows to precisely capture notes with vibrato, bending, and slide. Instrument-specific parameters such as playing techniques and fretboard position can be added as lookup tables and used for annotation. A detailed comparison among music description languages such as MIDI, MusicXML, and HyTime is provided in [95].

Three additional description languages are proposed in the literature with special focus on guitar synthesis. Karjalainen et al. present the *Guitar Control Language* to describe a guitar recording as a series of events, which are represented by a set of instrument-specific parameters. The authors provide an extensive list of guitar playing techniques, each technique is described in terms of performance gestures and emerging sound characteristics [91]. The *Expressive Notation Package* (ENP) is presented by Laurson et al. in [107]. This notation language extends regular music notation by a set of expressions that range from playing techniques such as vibrato, note parameters such as dynamic, to parameters describing the playing gesture such as plucking position and string number. Finally, the *ZIPI Music Parameter Description Language*, which is proposed by McMillen et al. in [123], allows to encode timbre properties such as brightness, roughness, and attack to characterize individual notes or group of notes. Järveläinen investigates, which of the parameters pitch, loudness, and timbre are the most salient ones from a perceptual perspective [85]. The author estimates also perceptual tolerance thresholds for different parameters.

In the MPEG-4 standard, the basic idea is to transmit audio data not as a sampled waveform but in a more efficient, object-based parametric representation. The *Harmonic and Individual Lines and Noise* (HILN) algorithm approximates a given audio signal using sinusoidal signals and noise [142]. The algorithm is incorporated into the second version of MPEG-4. Individual harmonic components are described by magnitude and frequency. Grouped harmonic components such as musical notes are described by the fundamental frequency, the spectral envelope of the harmonics, and the magnitude of the fundamental frequency. To improve the overall sound quality, the estimated sinusoidal components are subtracted from the original signal and the remaining signal is modeled using a linear filter. It is shown in [142] that the HILN algorithm leads to comparable audio quality as transform-based coding algorithms such as AAC. At the same time, HILN allows to change the parameters pitch and tempo for the individual signals on the decoder side.

12 Contribution

The contributions described in the following sections result from the collaboration with Patrick Kramer, Christian Dittmar, and Gerald Schuller and were previously published in [103], [104] and [9]. In Section 12.1, a novel physical modeling algorithm will be introduced that mimics the sound production of an electric bass guitar and implements all discussed plucking and expression styles. In Section 12.2, a parametric instrument codec will be discussed, which combines the analysis and re-synthesis of a bass guitar track.

12.1 Physical Modeling Algorithm for Realistic Bass Guitar Synthesis

The proposed algorithm to synthesize bass guitar tracks is illustrated as a flowchart in Figure 12.1. Its components will be detailed in the following sections. The synthesis model can be seen as the counterpart to the bass guitar transcription algorithm presented in Chapter 4 since it takes all extracted note-event parameters such as onset, offset, pitch, and playing techniques as input and generates a synthesized version of the original bass guitar recording. Sound examples of the proposed synthesis algorithm are available at [105].

12.1.1 Waveguide Model

The proposed bass guitar synthesis algorithm is based on the digital waveguide approach discussed in Section 11.2. The vibrating string is sampled at equally-spaced string positions and the corresponding deflection values are stored. The basic waveguide model consists of two *delay lines* that simulate two waves traveling into opposite directions on the string as explained in Section 2.1.3.

The electric bass guitar string has a rigid termination at the bridge and at the nut. Therefore, both delay lines are connected on their ends. The wave reflection is modeled by inverting the sign of the string deflection values at the delay line end points. The overall decay of the simulated string is adjusted by the damping factor $g < 1$ of the *loss filter*, which can be set as an input parameter of the model. This allows to synthesize different amounts of string damping that go along with different plucking styles such as *finger-style* (FS) and *muted* (MU). The loss filter also simulates the frequency losses of the signal over time. A parametric zero phase FIR filter is used for this purpose as proposed in [179]. Since the bass guitar strings show different amounts of frequency losses over time, different parameters were chosen for the loss filter depending on the string number of the synthesized note.

In order to implement arbitrary fundamental frequency values, a simple linear interpolation of fractional delay values is used. This allows to tune the waveguide model to f_0 values that are not

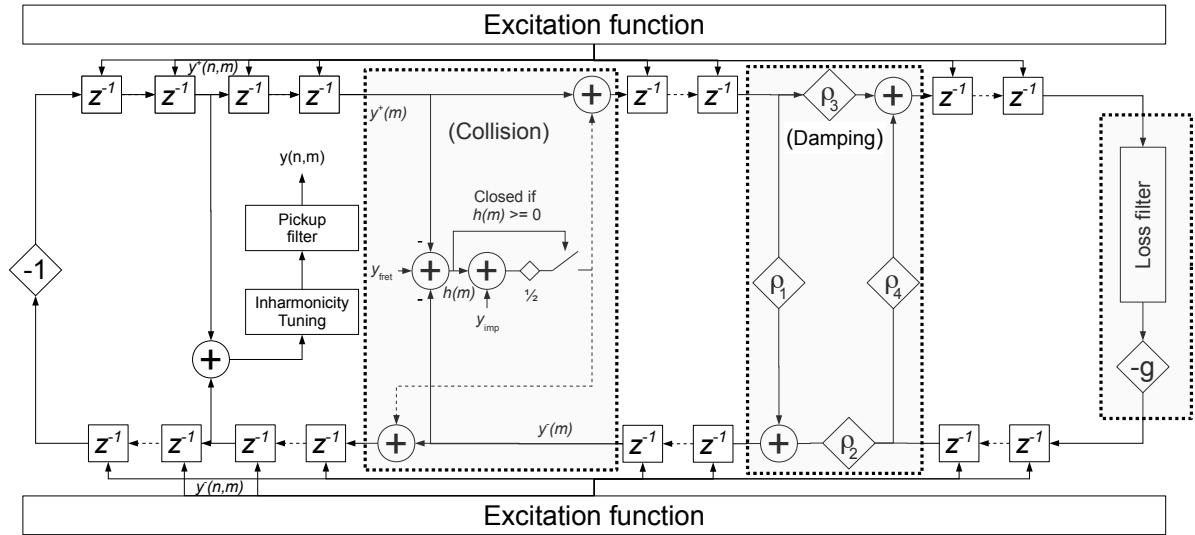


Figure 12.1: Proposed waveguide model for bass guitar synthesis including a collision and damping point, a loss filter, inharmonicity tuning, and a pickup filter [9].

a partial of the sampling frequency f_s . The linear interpolation offers two advantages. First, it is computationally efficient. Second, as will be shown in Section 13.1, listening tests revealed that this approach is sufficient to model time-varying f_0 values that are characteristic for the playing techniques *bending* (BE), *slide* (SL), and *vibrato* (VI).

Excitation Function

Depending on the plucking style, both delay lines of the waveguide model are initialized with an *excitation function*. The excitation function predefines the timbre and the spectral envelope of the resulting tone. This function represents the string displacement at the time of plucking, it is influenced by the plucking style, position, and intensity.

Figure 12.2 gives three examples for excitation functions. The *finger-style* plucking style results in a triangular excitation function with the maximum string displacement at the plucking point. The effect of using a plastic pick for the plucking style *picked* is simulated by a sharper peak at the plucking position. The *slap thumb* plucking style is characterized by an initial velocity peak caused by hammering the thumb on the string. The velocity function can be translated into a displacement function by integration.

The *absolute plucking position* is considered to be constant for arbitrary string number and fret number values. The physical length of the vibrating string defines the fundamental frequency f_0 . Since the waveguide model only simulates the vibrating part of the string, the *relative plucking position* within the displacement function changes for different f_0 .

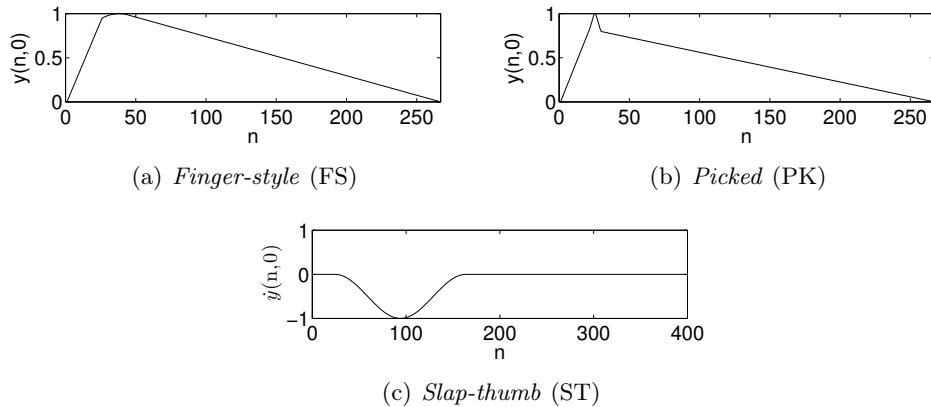


Figure 12.2: Model excitation functions for the plucking styles *finger-style* (FS) and *picked* (PK) as well as initial velocity function for the plucking style *slap-thumb* (ST). n denotes the delay line element.

Electro-magnetic Pickups

As discussed in Section 2.1.1, electro-magnetic pickups capture the string vibration on electric bass guitars and convert it into an electric signal. In order to generate the model output signal, the string deflection is captured at the delay line element that corresponds to the geometric position of the pickup on the real instrument. The relative pickup position is computed similarly to the relative plucking position in dependence of the vibrating string length.

The instrument output signal is influenced by the response of the inner tone circuitry including the pickup [93]. As detailed in [103], a SPICE simulation of the circuitry was performed based on the internal properties resistance, capacitance, and inductance of the pickup, a fixed tone circuit, and a connected load that includes the capacitance of the instrument cable. An FIR filter was implemented based on the simulation results and included into the waveguide model to filter the output signal $y(n, m)$.

Fret Collision

If the plucking styles *slap-thumb* (SP) and *slap-pluck* (SP) are used, the string collides with the frets on the instrument neck due to its high displacement as discussed in Section 2.1.2. The method presented by Evangelista in [52] is applied to simulate the string-fret collision in the waveguide model. A scattering junction is located between both delay lines at all fret positions. In every sampling step m , the string displacement $y_{in}(m)$ is computed as the sum of both delay lines and compared to the string-fret distance y_{fret} .

$$h(m) = y_{fret} - y_{in}(m) = y_{fret} - (y^+(m) + y^-(m)). \quad (12.1)$$

If $h(m) \geq 0$, a collision between the string and the fret is detected. In case of a collision, a fraction of the delay line elements that travel into the scattering junction is reflected. The string displacement is held fixed at $y_{in}(m) = y_{fret}$ until the string moves back again ($h(m) < 0$). The

collision is assumed to be not completely inelastic. The bouncing back of the string from the fret is simulated by adding a constant impulse y_{imp} to the string displacement function for a short period of the collision time.

The output values of the scattering junction $y_{\text{out}}^-(m)$ and $y_{\text{out}}^+(m)$ in the moment of a collision are given by

$$\begin{bmatrix} y_{\text{out}}^-(m) \\ y_{\text{out}}^+(m) \end{bmatrix} = S_c \begin{bmatrix} y_{\text{in}}^-(m) \\ y_{\text{in}}^+(m) \end{bmatrix} + \frac{y_{\text{fret}} + y_{\text{imp}}}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (12.2)$$

with the scattering matrix

$$S_c = \frac{1}{2} \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}. \quad (12.3)$$

Depending on the current fretboard position, up to 20 frets are simulated in that manner. The distance between the frets are computed with respect to the real geometry of the instrument neck. The fret height values are set according to a slightly bowed instrument neck.

Finger Damping

The *harmonic* (HA) and *dead-note* (DN) expression styles are characterized by damping the string at a certain fretboard position. The damping that results from the punctual finger-string interaction is simulated using a wave digital resistor as proposed by Pakarinen in [135]. The resistor is connected to the two delay lines with a three-port junction and is defined by the damping factors ρ_1 to ρ_4 . If the damping is activated, only those standing waves keep sounding (after some cycles) that have a node $y_{\text{in}}(m) = 0$ at the damping point.¹ The output values $y_{\text{out}}^-(m)$ and $y_{\text{out}}^+(m)$ of the wave digital filter junction are given by

$$\begin{bmatrix} y_{\text{out}}^-(m) \\ y_{\text{out}}^+(m) \end{bmatrix} = \begin{bmatrix} \rho_1 & \rho_2 \\ \rho_3 & \rho_4 \end{bmatrix} \begin{bmatrix} y_{\text{in}}^+(m) \\ y_{\text{in}}^-(m) \end{bmatrix}. \quad (12.4)$$

The damping factors ρ_1 to ρ_4 are set in accordance to the given expression style.

12.1.2 Tuning of the Model Parameters

In this section, two approaches for tuning the bass guitar synthesis model proposed in Section 12.1.1 will be presented. The tuning process aims at reproducing the sonic properties of the bass guitar² that was used to record the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset detailed in Section 4.2.2. By better reproducing the sound of the original instrument, the perceptual quality of the synthesis algorithm is expected to improve.

First, isolated note recordings that cover the full fretboard range (all four strings, open string up to the 12th fret position) were analyzed. These notes were taken from the *IDMT-SMT-BASS* dataset (see Section 4.2.1). Three steps are performed to tune the synthesis algorithm: tuning of the temporal loss parameter, tuning of the frequency loss filter, and tuning of the inharmonicity of the synthesized notes.

¹This corresponds to the sound production principle of the *harmonics* (HA) expression style as discussed in Section 2.1.2.

²Fame Baphomet 4 NTB, string gauges 1.05 mm (E string), 0.85 mm (A string), 0.65 mm (D string), and 0.45 mm (G string).

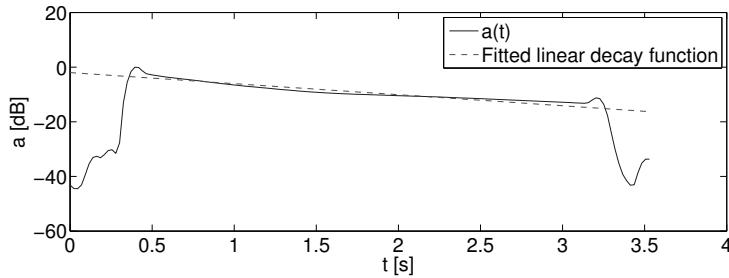


Figure 12.3: Normalized magnitude envelope of bass guitar note with fitted linear decay function over time.

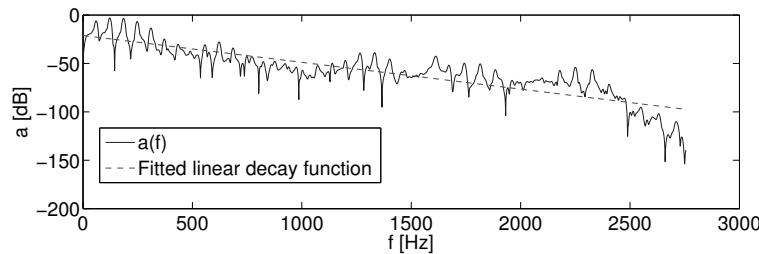


Figure 12.4: Normalized magnitude envelope of bass guitar note with fitted linear decay function over frequency.

Tuning of the Loss Filters

As described in Section 12.1.1, the simulation of the string vibration losses is realized by two components, a damping factor g and an FIR filter. The damping factor g determines the overall temporal decay of each note. To tune this parameter, the decay rate α_t of the spectral envelope over time is extracted using linear regression over the magnitude spectrogram (in dB) in the note decay part. In Figure 12.3, a normalized magnitude envelope of a bass guitar note and the corresponding fitted linear decay function is shown. The decay rate is estimated for all notes over the complete instrument fretboard. The damping factor g in the waveguide model is computed as $g = 10^{\frac{-\alpha_t NT}{20}}$ with N denoting the number of delay elements and T denoting the time of one sample.

The loss filter is a zero-phase low-pass FIR filter. It introduces a faster decay to higher harmonics to reproduce the natural faster damping of higher frequencies due to string stiffness. The applied filter allows to adjust the increase in decay towards higher frequencies with only one parameter [179]. To tune the FIR filter, the decay rate α_f of the spectral envelope over frequency is estimated. Figure 12.4 shows the magnitude decay over frequency for the same bass guitar note as shown in Figure 12.3. Then, the time gradients $\frac{\partial \alpha_f}{\partial t}$ of every string-fret combination are estimated and the frequency loss filter is tuned accordingly by minimizing the distance of this gradient between synthesized and original decay.

12.1.3 Inharmonicity

As explained in Section 2.1.3, the inharmonicity of string instruments needs to be considered, especially for the high string diameter values of the bass guitar [55]. Therefore, the following approach is included in the synthesis model in order to simulate the inharmonic relationship between the harmonic frequencies. From the analysis stage explained in the first part of this thesis, both the inharmonicity coefficient β and the fundamental frequency f_0 are given for each note. In the synthesized note, the center frequencies f_h of the harmonic series are located at $f_h = (h + 1)f_0$ with $h \geq 0$. The inharmonicity tuning aims to emulate the frequency relationship $\hat{f}_h = (h + 1)f_0\sqrt{1 + \beta(h + 1)^2}$.

First, the band signal of each harmonic component is extracted using STFT, binary masking, and inverse STFT. The binary mask for each harmonic component is centered around its ideal frequency f_h with a bandwidth of f_0 . During the spectral masking, the mirror spectra are set to zero which yields the analytic signal [98] after applying the inverse STFT. The analytic signal contains the original signal in its real part and a version of the original signal with constant phase shift of 90° in its imaginary part. Second, the analytic signal is modulated with a complex exponential $\exp(j2\pi(\hat{f}_h - f_h))$ in order to shift the band signal upwards by the deviation between the inharmonic frequency \hat{f}_h and the ideal frequency f_h . This procedure is repeated for each harmonic component and the shifted spectral bands are superimposed to get an inharmonic signal with the original timbre qualities. In the literature, an alternative method to incorporate inharmonicity into physical modeling algorithms is to use an additional allpass filter [148]. This approach was not investigated here.

12.2 Parametric Model-based Audio Coding of Bass Guitar Tracks

12.2.1 Overview

The proposed coding scheme consists of an encoder, which deals with the parameter extraction as explained in the first part of the thesis, and a decoder, which re-synthesizes an audio signal based on the transmitted parameters. Figure 12.5 gives an overview. As mentioned before, the focus is on isolated bass guitar tracks. The analysis of mixed audio signals would require an additional source separation algorithm.

In addition to the transmission and coding of the audio data, the coder also allows for *generating score or tablature notation* as detailed in Section 2.2. Also, since the bass guitar track is represented by a note-wise parametrization, these parameters can be altered in order to change the stylistic properties of a bass line before it is re-synthesized. This *alteration of note-wise parameters* could for instance change the plucking and expression styles, but also slightly change the note onsets in order to influence the *micro-timing* of a bass line.

12.2.2 Parameter & Bitrate

Table 12.1 summarizes all parameters, which need to be transmitted for each note event. Based on the ranges of values and the necessary precision, the number of quantization steps can be derived. In total, 82 bits are necessary to encode all parameters to describe one note event.

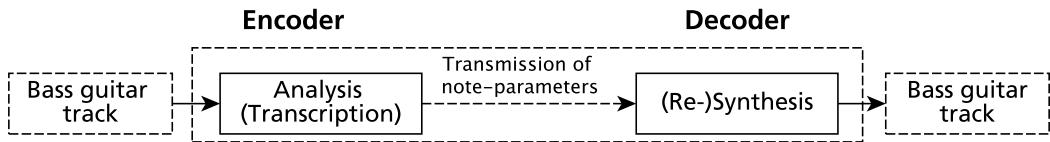


Figure 12.5: Flowchart of proposed parametric bass guitar coder.

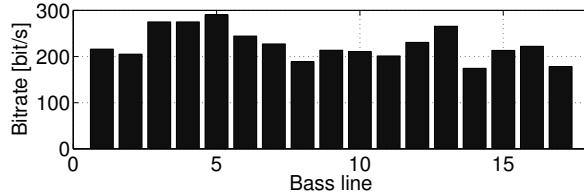


Figure 12.6: Bitrates using the proposed instrument coder for all 17 bass lines from the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset.

Table 12.1: Transmitted parameters and their quantization in the proposed bass guitar coder.

Parameter	Range	Resolution	Quantization bits (steps)
Loudness \mathcal{L}	$[0, 127 \text{ dB}]$	0.1 dB	11 (2048)
Plucking Style \mathcal{S}_P	$[1, 5]$	1	3 (8)
Expression Style \mathcal{S}_E	$[1, 6]$	1	3 (8)
String Number \mathcal{N}_S	$[1, 4]$	1	2 (4)
Fret Number \mathcal{N}_F	$[0, 24]$	1	5 (32)
Onset \mathcal{O}	$[0, 30 \text{ s}]$	0.01 s	12 (4096)
Duration \mathcal{D}	$[0, 20 \text{ s}]$	0.01 s	11 (2048)
Magnitude decay α_t	$[0, 127 \text{ dB/s}]$	1 dB/s	7 (128)
Fundamental frequency f_0	$[41.2 \text{ Hz}, 382.0 \text{ Hz}]$	0.1 Hz	12 (4096)
Modulation frequency $\chi_{\text{f,mod}}$	$[0, 12 \text{ Hz}]$	0.1 Hz	7 (128)
Modulation lift $\chi_{\text{mod,lift}}$	$[0, 500 \text{ cent}]$	1 cent	9 (512)
\sum			82 bit/Note

In Figure 12.6, the achieved bitrates for all bass lines from the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset are illustrated. The bit-rate depends mainly on the tempo and the note density over time, the average bit-rate that can be achieved for this dataset is 225.4 bit/s. In comparison to conventional perceptual audio coding schemes, only a fraction of the bit-rate is necessary. This improvement comes with a strong specialization towards one particular musical instrument. Hence, the proposed parametric audio codec can only be applied to encode isolated bass guitar tracks.

12.2.3 Application for Polyphonic Music

In order to apply the proposed parametric coding approach for multi-instrumental, polyphonic music, a specific instrument coder must be designed for each instrument. The note parameters have to be adjusted to the typical sound production and playing techniques. In order to estimate the required bit-rate to encode polyphonic music pieces with the proposed parametric coding approach, seven realistic multi-track MIDI files were obtained from [151]. For the sake of simplicity, the bit-rate of 82 bit/Note is assumed for all instrument tracks. As shown in Table 12.2, the MIDI files have an average number of 4.7 instrument tracks. The estimated bit-rate is around 2.25 kbit/s.

Table 12.2: Number of tracks, note density per second, and required bit-rate to encode multi-track MIDI files.

File	Number of instrument tracks	Note density	Bit-rate
BLUES.MID	4	34.53 notes/s	2.83 kbit/s
COUNTRY.MID	5	36.04 notes/s	2.96 kbit/s
HIP_HOP.MID	5	29.12 notes/s	2.39 kbit/s
JAZZ.MID	4	20.13 notes/s	1.66 kbit/s
LATIN.MID	7	37.9 notes/s	3.11 kbit/s
REGGAE.MID	5	13.29 notes/s	1.09 kbit/s
ROCK.MID	3	20.97 notes/s	1.72 kbit/s
Average	4.7143	27.42 notes/s	2.25 kbit/s

13 Evaluation

The experiments described in this section were conducted in collaboration with Patrick Kramer, Christian Dittmar (Semantic Music Technologies group, Fraunhofer IDMT), and Gerald Schuller (Technische Universität Ilmenau) and were previously published in [104] (Section 13.1) and [9] (Section 13.2 and Section 13.3).

13.1 Perceptual Audio Quality of Synthesized Basslines

Motivation & Goals

As discussed in Section 11.3, no bass guitar synthesis algorithm based on physical modeling exists that allows to synthesize different playing techniques and could be used for a comparative evaluation. However, when considering the application of the proposed synthesis algorithm within an instrument codec as discussed in Section 12.2, existing conventional audio codecs can be used to compare the perceptual quality of the resulting audio signals. In this experiment, a MUSHRA listening test (Multi Stimulus test with Hidden Reference and Anchor) was performed to evaluate the perceptual quality of the re-synthesized basslines. The test was performed throughout all playing techniques separately.

Dataset & Participants

The dataset used in this experiment was assembled by Patrick Kramer in [103]. As shown in Table 13.1, nine basslines with different playing techniques were included. Twelve participants ranging from average music listeners to professional bass guitar players took part in the listening test.

For each bassline, the order of the stimuli was randomized. The participants were allowed to listen to each stimuli as often as they wanted. Each stimulus had to be rated on a scale between 0 for poor audio quality and 100 for excellent audio quality.

Table 13.1: Plucking styles (PS) and expression styles (ES) used in the bass lines used as stimuli of the first listening test.

Bassline	1	2	3	4	5	6	7	8	9
Plucking Style	FS	FS	FS	FS	FS	MU	PK	ST	SP
Expression Style	BE	DN	HA	NO	VI	NO	NO	NO	NO

Experimental Procedure

In the set of stimuli of the listening test, the original bassline recordings were included as hidden reference (“Reference”). Low-pass filtered versions ($f_{\text{cut}} = 3.5 \text{ kHz}$) were used as hidden anchors (“3.5 kHz LP”). The next three stimuli were different versions of the original bassline recordings processed by several low-bit audio codecs. In particular, the codecs AMR-WB+ (at 6.0 kbit/s), HE-AAC (at 14.1 kbit/s), and Ogg Vorbis (at 17.2 kbit/s) were chosen. Maximum compression settings were used to achieve the lowest possible bit-rate for each codec. The last stimulus was the re-synthesized bassline using the algorithm described in Section 12.1 without additional model tuning. Table 13.2 provides an overview over all stimuli used in the first listening test.¹

Table 13.2: Stimuli used in the first MUSHRA listening test.

Stimulus	Type	Low-pass filtered	Processed with Audio Codec	Synthesized Audio	Label
1	Original				Reference
2	Original	x			3.5 kHz LP
3	Original		x		AMR-WB+
4	Original		x		HE-AAC
5	Original		x		Ogg Vorbis
6	Synthesized			x	Synthesis

Results & Summary

The results of the listening test are illustrated in Figure 13.1. The first nine columns show the mean ratings and 95 % confidence intervals for the nine basslines. The last column shows the averaged results.

The reference signal and the anchor signal were rated as “excellent” and “good”, respectively. The average ratings for the three audio codecs are all in the upper “poor” region while the proposed bass guitar codec achieved around 55 % in the “fair” region.

Especially for the basslines 1, 4, 5, and 6 that are played with the plucking style *finger-style* (FS) and *muted* (MU), the synthesized basslines clearly surpass the stimuli processed by the audio codecs. The basslines 3, 7, 8, and 9 relate to the *harmonics* (HA), *picked* (PK), *slap-pluck* (SP), and *slap-thumb* (ST) techniques, which are all characterized by high frequency spectral energy in the audio signal.² For these techniques, the listening test still shows higher ratings for the synthesized basslines compared to the audio codec versions. However, the improvement is smaller than for the FS and MU technique. For the second bassline, which includes the percussive *dead-note* technique, the synthesized version was rated worst.

In summary, the average results show that the proposed instrument audio codec for isolated bass guitar recordings outperformed three different audio codecs in terms of perceptual quality, if

¹ The abbreviations for the plucking and expression styles are explained in Table 2.1.

²This results from higher fundamental frequencies (HA), sharp attacks (PK), and from the collision between the string and the fretboard (SP, ST) as detailed in Section 2.1.2.

those are set to very low bit rates. At the same time, the instrument codec allows for a significantly lower bit rate 0.06 kbit/s.

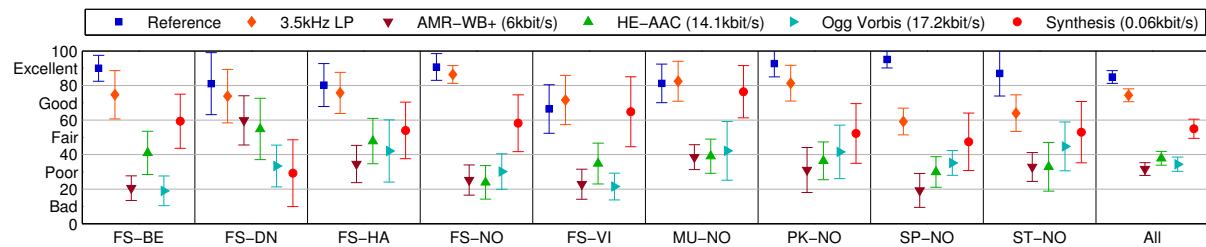


Figure 13.1: The results of the first MUSHRA listening test for the different stimuli described in Table 13.2. Mean ratings with 95 % confidence intervals are given [103, 104].

13.2 Perceptual Improvements by Applying Model Tuning to Specific Instruments

Motivation & Goals

In Section 12.1.2, different methods were proposed to tune the parameters of the proposed synthesis algorithm to better match the sonic properties of a particular instrument. In this listening test, the influence of those methods on the perceptual quality of synthesized basslines was investigated.

Dataset & Participants

17 recorded basslines from the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset presented in Section 4.2.2 were used as reference audio data. The basslines cover various combinations of bass guitar playing technique as detailed in Table 13.3. Nine participants took part in both tests, most of whom are mostly semi-professional musicians.

Table 13.3: Plucking styles (PS) and expression styles (ES) used in the basslines of the *IDMT-SMT-BASS-SINGLE-TRACKS* dataset [9].

Experimental Procedure

Similar to the previous experiment, a MUSHRA listening test was performed. Table 13.4 details all stimuli that were used.

Table 13.4: Stimuli used in the MUSHRA listening test described in Section 13.2

Stimulus	Audio type	Low-pass filtered	Tuning of loss filters	Inharmonicity Label
1	Original			Reference
2	Original	x		3.5 kHz LP
3	Synthesized			ICASSP 2012
4	Synthesized		x	Optimized
5	Synthesized		x	Optimized + Inharmonicity

In addition to the reference signal and anchor signal, three different synthesis model configurations were used to create additional stimuli. The first configuration “ICASSP 2012” is the initial version of the synthesis model used in Section 13.1 and presented in [104]. This model can be considered as un-tuned model as its parameters were empirically adjusted by only a small set of reference bass guitar recordings. The second configuration “Optimized” is tuned to the sonic properties of the bass guitar that was used to record the reference basslines used in the listening test. Here, the filter parameters of the synthesis model were optimized based on given single-note recordings as described in Section 12.1.2. The third configuration “Optimized + Inharmonicity” is the tuned model with an additional inharmonicity tuning as described in Section 12.1.3.

Results & Summary

The results of the listening test are shown in Figure 13.2. The first 17 columns illustrate the mean ratings and 95 % confidence intervals for the 17 basslines. In the last column, the averaged results are given.

While the reference was consistently rated as “excellent”, the low-pass filtered anchor was rated as “good”. For the basslines 7, 13, and 16, which were played using the slap techniques slap-pluck (SP) and slap-thumb (ST), the anchor ratings were significantly lower. Since the slap technique is characterized by typical high frequency attack transients, the low-pass filtering impairs the audio quality to a stronger extend.

The results for the synthesis model configurations show that the improvements in the perceived audio quality of the re-synthesized basslines are very small. It can be seen in the final column that the optimized synthesis algorithms with and without inharmonicity show no significant improvement to the baseline model. All three synthesized versions achieved an average rating of around 40. Hence, the perceptual quality can be described between “poor” and “fair”.

Interviews with the listening test participants confirmed that the overall “synthetic” impression of the synthesized bass-lines still “masks” the perceptual improvements by the proposed model tuning approaches. Additionally, two aspects were mentioned, which were not considered in the

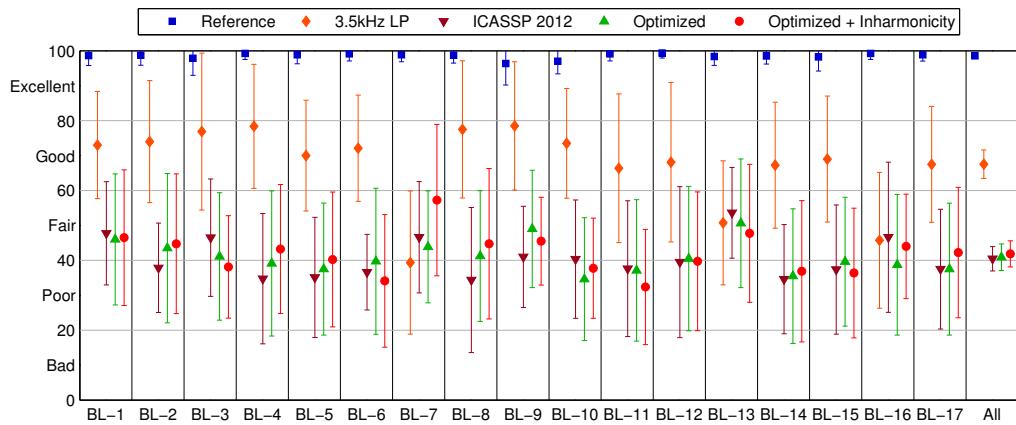


Figure 13.2: The results of the MUSHRA listening test for the different stimuli described in Table 13.4 for the 17 basslines. Mean ratings with 95 % confidence intervals are given [9].

re-synthesis: dynamic variations within the basslines and the typical noise caused by hand slides over the strings when the fretboard position is changed.

Future research must identify aspects of the instrument sound production that are still not captured by the synthesis model. At the current implementation stage, the estimation of temporal and spectral decay parameters as well as the inharmonicity coefficient from single notes can be considered as less important to the overall perceptual quality of the synthesis.

13.3 Importance of Plucking Styles and Expression Styles on the Perceptual Quality of Synthesized Basslines

Motivation & Goals

As discussed in Section 2.1.2, the sound production on string instruments such as the bass guitar can be separated into two physical gestures—the plucking style and the expression style. These gestures have a strong influence on the attack part and decay part of the played note, respectively. In this listening test it was investigated, how important the correct parameter values for plucking style and expression style are to achieve a realistic re-synthesis of a given bassline.

Dataset & Participants

Again, the IDMT-SMT-BASS-SINGE-TRACKS data set was used as reference audio data. The participants of this experiment were the same as for the previous experiment.

Experimental Procedure

Six stimuli were used in this listening test as shown in Table 13.5. Again, the hidden reference signal (“Reference”) and the low-pass filtered anchor signal (“3.5 kHz LP”) were chosen as in the

previous experiment. Additionally, four synthesized versions were generated: In the first version (“Optimized + Inharmonicity”), all plucking style and expression style parameters were used as given in the ground truth annotations. In the second version (“PS:FS”), the plucking styles of all notes were unified and set to the *finger-style* (FS) technique. The expression styles were not modified. In the third version (“ES:NO”), the expression styles of all notes were unified and set to *normal* (NO) and the plucking styles remain as in the reference annotations. Finally, in the fourth version (“PS:FS ES:NO”), both the plucking styles and expression styles were set to FS and NO, respectively.

Table 13.5: Stimuli used in the MUSHRA listening test described in Section 13.3

Stimulus	Audio type	Low-pass filtered	Tuning of loss	Inharmonicity filters	Plucking & expression styles	Label
1	Original				all PS, all ES	Reference
2	Original	x			all PS, all ES	3.5 kHz LP
3	Synthesis		x	x	all PS, all ES	Optimized + Inharmonicity
4	Synthesis		x	x	PS = FS, all ES	PS:FS
5	Synthesis		x	x	all PS, ES = NO	ES:NO
6	Synthesis		x	x	PS = FS, ES = NO	PS:FS ES:NO

Results & Summary

The results of the second experiment are shown in Figure 13.3.

The ratings of the reference and the anchor are comparable to the first experiment. Again, the perceived audio quality of the slap basslines is more strongly affected by the low-pass filtering.

The averaged results in the final column indicate that changing the plucking style of all notes decreases the audio quality of the synthesized basslines stronger than changing the expression styles. Presumably, this is because a large fraction of the notes in realistic bass-lines have no particular expression style. For the IDMT-SMT-BASS-SINGLE-TRACKS dataset, around 87% of all notes have the expression style *normal* (NO) as shown in Table 4.1. Therefore, removing all vibrato, bending, slide, harmonic, and dead note techniques affects only a small fraction of the notes in the basslines. On the other hand, if a bassline has a different plucking style than the *finger-style* technique (FS), changing the plucking style affects all notes in the given bassline. The negative effect of changing the plucking style to *finger-style* is most prominent for the basslines 7, 13, 14, and 16 that incorporate either the *slap*-techniques ST and SP or the *picked* technique (PK).

In summary, plucking styles have a higher importance for re-synthesizing string instrument recordings than the expression styles have. A reliable parametrization and modeling of the attack

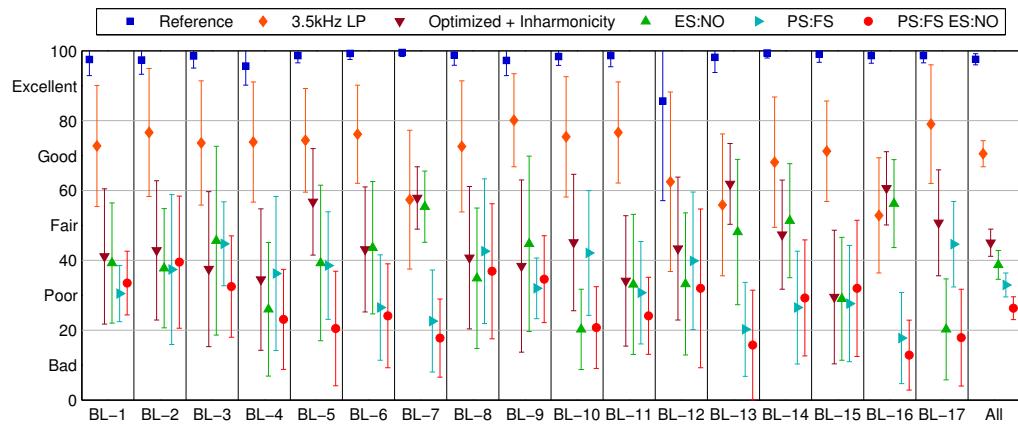


Figure 13.3: The results of the MUSHRA listening test for the different stimuli described in Table 13.5 for the 17 basslines. Mean ratings with 95 % confidence intervals are given [9].

part of instrument notes (which is mainly influenced by the plucking style) is therefore crucial to obtain realistic synthesis results.

14 Summary

In the third part of this thesis, a novel sound synthesis algorithm was proposed that allows to re-synthesize bass guitar recordings based on a compact set of note parameters. The algorithm implements 11 different plucking and expression styles. Furthermore, it can be tuned to the sonic properties of a particular bass guitar. The combination of the bass guitar transcription algorithm presented in the first part of this thesis and the synthesis algorithm as a parametric audio coder was discussed. In comparison to conventional audio coding algorithms, the parametric coding approach requires only a fraction of the bit-rate. At the same time, the algorithm is currently specialized and limited towards isolated bass guitar tracks.

A first listening test revealed that the proposed synthesis algorithm outperforms conventional audio coding algorithms at very low bit-rate settings in terms of perceptual quality. In a second test, the participants reported that the overall “synthetic” sound impression of the re-synthesized bass lines still masks the perceptual improvements due to the model tuning. Future work must address the simulation of physical phenomena such as string beating and noise-like attack transients. Also, external factors such as the sound of the amplifier, loudspeaker, and additional audio effects need to be considered in the synthesis algorithm. In a third listening test, it was observed that the realistic synthesis of plucking styles is even more important than the synthesis of expression styles.

List of Figures

1.1	Flowchart illustrating the structure of this thesis.	2
2.1	Electric bass guitar with instrument parts.	11
2.2	Logarithmic spacing of frets on the string instrument neck.	12
2.3	Plucking hand positioning and an example note for the <i>finger-style</i> (FS) plucking style.	15
2.4	Plucking hand positioning and an example note for the <i>picked</i> (PK) plucking style.	16
2.5	Plucking hand positioning and an example notes for the <i>slap-pluck</i> (SP) and <i>slap-thumb</i> (ST) plucking styles.	17
2.6	Plucking hand positioning and an example note for the <i>muted</i> (MU) plucking style.	18
2.7	Playing hand positioning and two example notes for the <i>dead-note</i> (DN) and <i>harmonics</i> (HA) expression styles.	19
2.8	String bending using the playing hand applied for the <i>vibrato</i> (VI) and <i>bending</i> (BE) expression styles.	20
2.9	Reflection of a traveling wave at a fixed end.	21
2.10	Initial string displacement and vibration modes if string is plucked in the middle.	22
2.11	Two-stage envelope model.	24
2.12	Score and tablature representation of a bassline.	25
3.1	Algorithmic steps of bass transcription algorithms.	37
3.2	Categorization criteria of related work towards the estimation of the playing technique and fretboard position from string instrument recordings.	41
4.1	Processing flowchart and all algorithmic steps of the proposed bass guitar transcription algorithm.	51
4.2	Piano-roll notation of a bassline.	52
4.3	STFT magnitude spectrogram and IF spectrogram with dB magnitude scale for a bass guitar note played with the <i>vibrato</i> expression style.	54
4.4	IF spectrogram, onset detection function, and detected onset positions for a bassline excerpt.	55
4.5	Pitch detection accuracy as a function of number of spectral templates peaks.	56
4.6	Harmonic spectral template based on a logarithmic frequency axis.	56
4.7	STFT magnitude spectrogram, tracked fundamental frequency course, as well as onset and offset times for a bass note played with the <i>vibrato</i> expression style.	57
4.8	Atom function used for spectral envelope modeling.	59
4.9	STFT spectrogram of a bass guitar note played with the <i>vibrato</i> expression style: original (left) and modeled (right).	60

4.10	STFT spectrogram of a note played with the <i>harmonics</i> expression style.	63
4.11	Characteristic fundamental frequency tracks for the expression styles <i>vibrato</i> , <i>bending</i> , and <i>slide</i>	64
4.12	Normalized autocorrelation function over $f_0(n)$	64
5.1	Pitch histogram over all notes in the <i>IDMT-SMT-BASS-SINGLE-TRACKS</i> dataset.	76
7.1	Criteria applied to categorize publications towards genre classification using score-based audio features.	87
7.2	Performance comparison of genre classification algorithms based on score-based audio features.	91
8.1	Tempo distribution over the genres in the <i>IDMT-SMT-BASS-GENRE-MIDI</i> data set.	101
8.2	Different metric levels in a salsa bassline.	102
8.3	On-beat and off-beat positions in a salsa bassline.	103
8.4	Syncopated and non-syncopated note sequences.	103
8.5	Salsa bassline encoded as binary beat sequence based on sixteenth-notes.	104
8.6	Funk bass pattern example.	104
9.1	Three classification paradigms evaluated for genre classification.	110
9.2	Proposed approach for similarity computation between two bass patterns of different lengths.	112
11.1	Single delay-loop model for physical modeling of string instruments.	126
12.1	Flowchart of the proposed bass guitar synthesis algorithm.	130
12.2	Model excitation functions for the plucking styles <i>finger-style</i> (FS) and <i>picked</i> (PK) as well as initial velocity function for the plucking style <i>slap-thumb</i> (ST). .	131
12.3	Normalized magnitude envelope of bass guitar note with fitted linear decay function over time.	133
12.4	Normalized magnitude envelope of bass guitar note with fitted linear decay function over frequency.	133
12.5	Flowchart of proposed parametric bass guitar coder.	135
12.6	Bitrates using the proposed instrument coder for all 17 bass lines from the <i>IDMT-SMT-BASS-SINGLE-TRACKS</i> dataset.	135
13.1	MUSHRA test results for the first test.	139
13.2	MUSHRA test results for the second test.	141
13.3	MUSHRA test results for the third test.	143

List of Tables

1.1	Type of audio data used in the three parts of this thesis.	3
1.2	Indexing symbols used in this thesis.	8
1.3	Symbols used in this thesis.	8
1.3	Symbols used in this thesis.	9
2.1	Proposed taxonomy of bass guitar playing techniques.	14
4.1	Overview of the <i>IDMT-SMT-BASS-SINGLE-TRACKS</i> dataset.	67
5.1	Experimental results for the classification of plucking styles and expression styles from isolated note recordings using MFCC features.	70
5.2	Classification results for plucking and expression style classification from isolated note recordings.	71
5.3	Confusion matrix for expression style classification on class level.	72
5.4	Confusion matrix for expression style classification on sub-class level.	72
5.5	Experiment results of the baseline experiment for automatic string classification using MFCC audio features.	73
5.6	Confusion matrix for human performance for string classification.	74
5.7	Results of string number classification for different experimental conditions. . . .	75
5.8	Compared bass transcription algorithms and applicable evaluation measures. All algorithms except S allow to compute both frame-wise and note-wise evaluation measures.	77
5.9	Frame-wise evaluation results for score-level evaluation. The best performing algorithms are indicated in bold print for each evaluation measure.	77
5.10	Note-wise evaluation results for score-level evaluation.	78
5.11	Confusion matrix for string number classification from bass guitar tracks. . . .	79
5.12	Confusion matrix for plucking style classification from bass guitar tracks. . . .	79
5.13	Confusion matrix for expression style classification from bass guitar tracks. . . .	79
8.1	Tonal note representations of a funk bassline.	93
8.2	Interval names and corresponding diatonic and chromatic intervals.	95
8.3	Investigated scales with corresponding binary scale templates.	98
8.4	Rhythmic note representations of a funk bassline. Score notation is given on top and different note parameters are given for all notes below.	99
8.5	Beat durations for different metric levels and different time signatures.	100
8.6	Overview of music genres in the <i>IDMT-SMT-BASS-GENRE-MIDI</i> data set. . . .	106

9.1	Selected audio features from the <i>jSymbolic</i> software that were extracted for the baseline experiment.	108
9.2	Confusion matrix for genre classification using the <i>jSymbolic</i> audio features and SVM classifier.	109
9.3	Classification performance of different configurations of similarity measures and aggregation strategies.	113
9.4	Genre classification confusion matrices for the three classification paradigms.	116
12.1	Transmitted parameters and their quantization in the proposed bass guitar coder.	135
12.2	Number of tracks, note density per second, and required bit-rate to encode multi-track MIDI files.	136
13.1	Plucking styles (PS) and expression styles (ES) used in the bass lines used as stimuli of the first listening test.	137
13.2	Stimuli used in the first MUSHRA listening test.	138
13.3	Plucking styles and expression styles used in the basslines of the <i>IDMT-SMT-BASS-SINGLE-TRACKS</i> dataset.	139
13.4	Listening test stimuli for the second MUSHRA listening test.	140
13.5	Stimuli used in the third listening test.	142

Acronyms

AMR-WB+	Extended Adaptive Multi-Rate-Wideband
CART	Classification and Regression Tree
EM	Expectation-Maximization
EMD	Earth Mover's Distance
ENP	Expressive Notation Package
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FM	Frequency Modulation
FS	Feature Selection
FST	Feature Space Transformation
GDA	Generalized Discriminant Analysis
GMM	Gaussian Mixture Models
HE-AAC	High Efficiency Advanced Audio Coding
HFC	High-Frequency Content
HILN	Harmonic and Individual Lines plus Noise
HMM	Hidden Markov Models
HPCP	Harmonic Pitch Class Profile
HPSS	Harmonic-Percussive Source Separation
HyTime	Hypermedia/Time-based Structuring Language
IDMT	Fraunhofer Institute for Digital Media Technology
IF	Instantaneous Frequency
IRMFSP	Inertia Ratio Maximization with Feature Space Projection
kNN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
LTI	Linear Time-Invariant
MFCC	Mel-Frequency Cepstral Coefficients
MIR	Music Information Retrieval
MIDI	Musical Instrument Digital Interface
MIREX	Music Information Retrieval Evaluation eXchange
ML	Maximum Likelihood
MPEG	Moving Picture Experts Group
MUSHRA	MULTiple Stimuli with Hidden Reference and Anchor
MusicXML	Music Extensible Markup Language
NB	Naive Bayes
NMF	Non-negative Matrix Factorization

NTB	Neck Through Body
OA	Overall Accuracy
OSC	Open Sound Control
PCA	Principal Component Analysis
PDF	Probability Density Function
PLCA	Probabilistic Latent Component Analysis
PreFEst	Predominant-F0 Estimation
RBF	Radial Basis Function
RCA	Raw Chroma Accuracy
RPA	Raw Pitch Accuracy
RWC	Real-world Computing Database
SOM	Self-organizing Maps
SBR	Spectral Bandwidth Replication
SPICE	Simulation Program with Integrated Circuit Emphasis
STFT	Short-time Fourier Transform
SVM	Support Vector Machine
VFAR	Voicing False Alarm Rate
VRC	Voicing Recall Rate

References

- [1] Toshihiko Abe, Takao Kobayashi, and Satoshi Imai, “Robust Pitch Estimation with Harmonics Enhancement in Noisy Environments based on Instantaneous Frequency”, in *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP)*, Philadelphia, PA, USA, 1996. 53
- [2] Jakob Abeßer, “IDMT-SMT-Bass - An audio database for bass transcription and signal processing”, http://www.idmt.fraunhofer.de/en/business_units/smt/published_datasets.html (Accessed: 05.10.2014). 66
- [3] Jakob Abeßer, “IDMT-SMT-BASS-SINGLE-TRACK - An audio database for bass transcription, pattern retrieval, and playing technique estimation”, http://www.idmt.fraunhofer.de/en/business_units/smt/published_datasets.html (Accessed: 05.10.2014). 66
- [4] Jakob Abeßer, “Automatic String Detection for Bass Guitar and Electric Guitar”, in *From Sounds to Music and Emotions - 9th International Symposium, CMMR 2012, London, UK, June 19-22, 2012, Revised Selected Papers*, Mitsuko Aramaki, Mathieu Barthet, Richard Kronland-Martinet, and Sølvi Ystad, Eds., vol. 7900, pp. 333–352, London, UK, 2013. Springer. 48, 49, 51, 61, 62, 66, 72, 73, 74
- [5] Jakob Abeßer, Paul Bräuer, Hanna Lukashevich, and Gerald Schuller, “Bass Playing Style Detection Based on High-level Features and Pattern Similarity”, in *Proceedings of the 11th International Society of Music Information Retrieval (ISMIR)*, pp. 93–98, Utrecht, Netherlands, 2010. 87, 88, 89, 90, 91, 105
- [6] Jakob Abeßer, Christian Dittmar, and Holger Großmann, “Automatic Genre and Artist Classification by Analyzing Improvised Solo Parts from Musical Recordings”, in *Proceedings of the 3rd Audio Mostly Conference on Interaction with Sound*, pp. 127–131, Piteå, Sweden, 2008. 85, 87, 90, 91
- [7] Jakob Abeßer, Christian Dittmar, and Gerald Schuller, “Automatic Recognition and Parametrization of Frequency Modulation Techniques in Bass Guitar Recordings”, in *Proceedings of the 42nd Audio Engineering Society (AES) International Conference on Semantic Audio*, pp. 1–8, Ilmenau, Germany, 2011. 18, 43, 51, 64, 70, 71
- [8] Jakob Abeßer, Klaus Frieler, Martin Pfleiderer, and Wolf-Georg Zaddach, “Introducing the Jazzomat project - Jazz solo analysis using Music Information Retrieval methods”, in *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Marseilles, France, 2013. 36

- [9] Jakob Abeßer, Patrick Kramer, Christian Dittmar, and Gerald Schuller, “Parametric Audio Coding of Bass Guitar Recordings using a Tuned Physical Modeling Algorithm”, in *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, 2013. 13, 66, 121, 129, 130, 137, 139, 141, 143
- [10] Jakob Abeßer, Hanna Lukashevich, and Paul Bräuer, “Classification of Music Genres based on Repetitive Basslines”, *Journal of New Music Research*, vol. 41, no. 3, pp. 239–257, 2012. 85, 87, 88, 89, 90, 91, 103, 105, 107, 110, 113, 114, 115
- [11] Jakob Abeßer, Hanna Lukashevich, Christian Dittmar, Paul Bräuer, and Fabienne Krause, “Rule-based classification of musical genres from a global cultural background”, in *Proceedings of the 7th International Symposium on Computer Music Modeling and Retrieval (CMMR)*, Málaga, Spain, 2010. 90
- [12] Jakob Abeßer, Hanna Lukashevich, Christian Dittmar, and Gerald Schuller, “Genre Classification using Bass-Related High-Level Features and Playing Styles”, in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 453–458, Kobe, Japan, 2009. 85, 87, 90, 91, 105
- [13] 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)*, pp. 2290–2293, Dallas, USA, 2010. 13, 43, 51, 62, 66, 69, 70, 85
- [14] Jakob Abeßer and Gerald Schuller, “Instrument-centered Music Transcription of Bass Guitar Tracks”, in *Proceedings of the AES 53rd Conference on Semantic Audio*, pp. 166–175, London, UK, 2014. 48, 51, 74, 77, 78
- [15] Yoko Anan, Kohei Hatano, Hideo Bannai, and Masayuki Takeda, “Music Genre Classification using Similarity Functions”, in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 693–698, Miami, USA, 2011. 87, 90, 91
- [16] Amélie Anglade, Rafael Ramirez, and Simon Dixon, “Genre Classification using Harmony Rules Induced from Automatic Chord Transcriptions”, in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 669–674, Kobe, Japan, 2009. 90
- [17] Mirko Arnold and Gerald Schuller, “A Parametric Instrument Codec for Very Low Bitrates”, in *Proceedings of the 125th Audio Engineering Society (AES) Convention*, pp. 427–433, San Francisco, CA, USA, 2008. 126
- [18] Ana M. Barbancho, Anssi Klapuri, Lorenzo J. Tardón, and Isabel Barbancho, “Automatic Transcription of Guitar Chords and Fingering from Audio”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 915–921, 2011. 48, 49
- [19] Isabel Barbancho, Ana M. Barbancho, Simone Sammartino, and Lorenzo J. Tardón, “Pitch and Played String Estimation in Classic and Acoustic Guitars”, in *Proceedings of the 126th Audio Engineering Society (AES) Convention*, Munich, Germany, 2009. 48, 49

- [20] Isabel Barbancho, Cristina de la Bandera, Ana M. Barbancho, and Lorenzo J. Tardón, “Transcription and Expressiveness Detection System for Violin Music”, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 189–192, Taipei, Taiwan, Apr. 2009. 45, 46
- [21] Roberto Basili, Alfredo Serafini, and Armando Stellato, “Classification of Musical Genre: A Machine Learning Approach”, in *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, pp. 7–18, Barcelona, Spain, 2004. 87, 88, 89, 91
- [22] G. Baudat and F. Anouar, “Generalized Discriminant Analysis Using a Kernel Approach”, *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000. 27
- [23] Emmanouil Benetos and Simon Dixon, “Polyphonic Music Transcription using Note Onset and Offset Detection”, in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 37–40, Praha, Czech Republic, 2011. 23
- [24] Carsten Bönsel, *Implementierung eines Videoanalyseverfahrens zur automatischen Erkennung der Position der Greifhand in Gitarrenaufnahmen*, Bachelor thesis, Technische Universität Ilmenau, Germany, 2012. 43
- [25] Bornemark, “BROOMSTICKBASS”, <http://www.bornemark.se/bb> (Accessed: 29.11.2013). 123
- [26] Leo Breiman, Jerome Friedman, Charles J. Stone, and R. A. Olshen, *Classification and Regression Trees*, Chapman and Hall/CRC, 1st edition, 1984. 29, 109
- [27] Roberto Bresin, “Articulation Rules for Automatic Music Performance”, in *Proceedings of the International Computer Music Conference (ICMC)*, pp. 294–297, Havana, Cuba, 2001. 90
- [28] Anne-Marie Burns and Marcelo Wanderley, “Computer Vision Method for Guitarist Fingering Retrieval”, in *Proceedings of the Sound and Music Computing Conference (SMC)*, pp. 1–7, Marseille, France, 2006. 48, 49
- [29] Anne-Marie Burns and Marcelo Wanderley, “Visual Methods for the Retrieval of Guitarist Fingering”, in *Proceedings of the Sound and Music Computing Conference (SMC)*, pp. 196–199, Paris, France, 2006. 48, 49
- [30] Estefanía Cano, Christian Dittmar, and Gerald Schuller, “Efficient Implementation of a System for Solo and Accompaniment Separation in Polyphonic Music”, in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO 2012)*, pp. 285–289, Bucharest, Romania, 2012. 36
- [31] Alfonso Pérez Carrillo, Jordi Bonada, Esteban Maestre, Enric Guaus, and Merlijn Blaauw, “Performance Control Driven Violin Timbre Model Based on Neural Network”, *IEEE Transactions on Audio Speech and Language Processing*, vol. 20, no. 3, pp. 1007–1021, 2012. 45

- [32] Alfonso Perez Carrillo and Marcelo M. Wanderley, “Learning and Extraction of Violin Instrumental Controls from Audio Signal”, in *Proceedings of the 2nd International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM)*, pp. 25–30, Nara, Japan, 2012. 42, 45, 46, 67
- [33] Zehra Cataltepe, Yusuf Yaslan, and Abdullah Sonmez, “Music Genre Classification Using MIDI and Audio Features”, *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, 2007. 87, 89, 90, 91
- [34] Chih-Chung Chang and Chih-Jen Lin, “LIBSVM: A Library for Support Vector Machines”, Tech. Rep., Department of Computer Science, National Taiwan University, Taipei, Taiwan, 2011. 28
- [35] Yin-Lin Chen, Tien-Ming Wang, Wei-Hsiang Liao, and Alvin W. Y. Su, “Analysis and Trans-Synthesis of Acoustic Bowed-String Instrument Recordings - A Case Study using Bach Cello Suites”, in *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, pp. 63–67, Paris, France, 2011. 35
- [36] Marcelo Cicconet, *The Guitar as a Human-Computer Interface*, PhD thesis, Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, 2010. 12, 43, 123
- [37] Giuseppe Cuzzucoli and Vincenzo Lombardo, “A Physical Model of the Classical Guitar, Including the Player’s Touch”, *Computer Music Journal*, vol. 23, no. 2, pp. 52–69, 1999. 13, 125
- [38] Alain de Cheveigné and Hideki Kawahara, “YIN, a fundamental frequency estimator for speech and music”, *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917, 2002. 44
- [39] H. G. de Dayan and A. Behar, “The Quality Of Strings For Guitars: An Experimental Study”, *Journal of Sound and Vibration*, vol. 64, no. 3, pp. 421–431, 1979. 22
- [40] Pedro J. Ponce de Léon and José M. Iñesta, “Pattern Recognition Approach for Music Style Identification Using Shallow Statistical Descriptors”, *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 37, no. 2, pp. 248–257, 2007. 87, 89, 90, 91
- [41] Pedro J. Ponce de Léon, David Rizo, and José Manuel Iñesta, “Towards a human-friendly melody characterization by automatically induced rules”, in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pp. 437–440, Vienna, Austria, 2007. 90
- [42] Christopher DeCoro, Zafer Barutcuoglu, and Rebecca Fiebrink, “Bayesian Aggregation for Hierarchical Genre Classification”, in *Proceedings of the 8th International Symposium on Music Information Retrieval (ISMIR)*, pp. 77–80, Vienna, Austria, 2007. 87, 90
- [43] Morteza Dehghani and Andrew M. Lovett, “Efficient Genre Classification using Qualitative Representations”, in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pp. 353–354, 2006. 87, 90, 91

- [44] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm”, *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 39, pp. 1–38, 1977. 28
- [45] Christian Dittmar, Estefanía Cano, Jakob Abeßer, and Sascha Grollmisch, “Music Information Retrieval Meets Music Education”, in *Multimodal Music Processing. Dagstuhl Follow-Ups*, Meinard Müller, Masataka Goto, and Markus Schedl, Eds., vol. 3, pp. 95–120. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Germany, 2012. 25, 36, 52
- [46] Christian Dittmar, Karin Dressler, and Katja Rosenbauer, “A Toolbox for Automatic Transcription of Polyphonic Music”, *Proceeding of the Audio Mostly Conference*, pp. 58–65, 2007. 37, 38, 39, 40, 75, 77
- [47] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, Wiley-Interscience, 2nd edition, Nov. 2000. 28
- [48] Tuomas Eerola and Petri Toivainen, *MIDI Toolbox: MATLAB Tools for Music Research*, University of Jyväskylä, 2004. 98, 101
- [49] Valentin Emiya, Roland Badeau, and Bertrand David, “Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, Aug. 2010. 35
- [50] Cumhur Erkut, Matti Karjalainen, and Mikael Laurson, “Extraction of Physical and Expressive Parameters for Model-based Sound Synthesis of the Classical Guitar”, in *Proceedings of the 108th Audio Engineering Society (AES) Convention*, pp. 19–22, 2000. 43, 44, 124, 125, 126
- [51] Slim Essid, *Classification automatique des signaux audio-fréquences: reconnaissance des instruments de musique*, PhD thesis, Université Pierre et Marie Curie, Paris, France, 2005. 27
- [52] Gianpaolo Evangelista, “Physically Inspired Playable Models of Guitar, a Tutorial”, in *Proceedings of the 4th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pp. 1–4, Limassol, Cyprus, 2010. 131
- [53] Gianpaolo Evangelista and Fredrik Eckerholm, “Player-Instrument Interaction Models for Digital Waveguide Synthesis of Guitar: Touch and Collisions”, *IEEE Transactions on Audio Speech and Language Processing*, vol. 18, no. 4, pp. 822–832, 2010. 125
- [54] Xander Fiss and Andres Kwasinski, “Automatic Real-Time Electric Guitar Audio Transcription”, in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 373–376, Praha, Czech Republic, 2011. 35, 48, 49
- [55] Neville H. Fletcher and Thomas D. Rossing, *The Physics Of Musical Instruments*, Springer, New York, London, 2nd edition, 1998. 14, 21, 22, 23, 24, 36, 41, 124, 134
- [56] Richard Mark French, *Engineering the Guitar - Theory and Practice*, Springer Science+Business Media, 2009. 11

- [57] Anders Friberg and Anton Hedblad, “A Comparison of Perceptual Ratings and Computed Audio Features”, in *Proceedings of the 8th Sound and Music Computing Conference (SMC)*, pp. 122–127, Padova, Italy, 2011. 88
- [58] Benoit Fuentes, Roland Badeau, and Gaël Richard, “Adaptive Harmonic Time-Frequency Decomposition of Audio Using Shift-Invariance PLCA”, in *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 401–404, Praha, Czech Republic, 2011. 39
- [59] Keinosuke Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 2nd edition, Sept. 1990. 27
- [60] A. Galembo and A. Askenfelt, “Measuring inharmonicity through pitch extraction”, *Speech Transmission Laboratory. Quarterly Progress and Status Reports (STL-QPSR)*, vol. 35, no. 1, pp. 135–144, 1994. 23
- [61] Alexander Galembo and Anders Askenfelt, “Signal representation and estimation of spectral parameters by inharmonic comb filters with application to the piano”, *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 197–203, Mar. 1999. 23
- [62] Geoffroy Peeters, “A large set of audio features for sound description (similarity and classification) in the CUIDADO project”, Tech. Rep., Ircam, Analysis/Synthesis Team, Paris, France, 2004. 62, 69
- [63] Francois Germain and Gianpaolo Evangelista, “Synthesis of Guitar by Digital Waveguides: Modeling the Plectrum in the Physical Interaction of the Player with the Instrument”, in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, vol. 3, pp. 25–28, New Paltz, NY, USA, 2009. 125
- [64] Charles F. Goldfarb, “HyTime: A standard for structured hypermedia interchange”, *IEEE Computer magazine*, vol. 24, no. 8, pp. 81–84, 1991. 127
- [65] Dmitry O. Gorodnichy and Arjun Yogeswaran, “Detection and tracking of pianist hands and fingers”, in *Proceedings of the 3rd Canadian Conference on Computer and Robot Vision*, pp. 1–8, Toronto, Canada, 2006. 47
- [66] Masataka Goto, “A Real-Time Music-Scene-Description System - Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals”, *Speech Communication*, vol. 43, no. 4, pp. 311–329, Sept. 2004. 25, 36, 37, 38, 39, 40
- [67] Masataka Goto, “Development of the RWC music database”, in *Proceedings of the 18th International Congress on Acoustics*, pp. 553–556, Kyoto, Japan, 2004. 41, 46
- [68] Masataka Goto, “AIST Annotation for the RWC Music Database”, in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pp. 359–360, Victoria, Canada, 2006. 41
- [69] Elaine Gould, *Behind Bars. The Definitive Guide to Music Notation*, Faber Music, 2011. 25

- [70] Maarten Grachten, Josep-Lluis Arcos, and Ramon Lopez de Mantaras, “Melodic Similarity: Looking for a Good Abstraction Level”, in *Proceedings of the 5th International Conference in Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004. 89
- [71] Mikus Grasis, Jakob Abeßer, Christian Dittmar, and Hanna Lukashevich, “A Multiple-Expert Framework for Instrument Recognition”, in *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Marseilles, France, 2013. 57, 69
- [72] Matthias Gruhne, Christian Dittmar, and Daniel Gärtner, “Improving Rhythmic Similarity Computation by Beat Histogram Transformations”, in *Proceedings of the 10th International Society for Music Information Retrieval (ISMIR)*, pp. 177–182, Kobe, Japan, 2009. 88
- [73] Enric Guaus and Josep Lluis Arcos, “Analyzing left hand fingering in guitar playing”, in *Proceedings of the 7th Sound and Music Computing Conference (SMC)*, pp. 284—290, Barcelona, Spain, 2010. 43
- [74] Enric Guaus, Tan Ozaslan, Eric Palacios, and Josep Lluis Arcos, “A Left Hand Gesture Caption System for Guitar Based on Capacitive Sensors”, in *Proceedings of the 10th International Conference on New Interfaces for Musical Expression (NIME)*, pp. 238—243, Sydney, Australia, 2010. 43, 44
- [75] Dan Gusfield, *Algorithms on strings, trees, and sequences: computer science and computational biology*, Cambridge University Press, Cambridge, UK, 1st edition, 1997. 105
- [76] Tilo Hähnel and Axel Berndt, “Expressive Articulation for Synthetic Music Performances”, in *Proceedings of the 10th International Conference on New Interfaces for Musical Expression (NIME)*, pp. 277–282, Sydney, Australia, 2010. 46
- [77] Stephen W. Hainsworth and Malcolm D. Macleod, “Automatic bass line transcription from polyphonic music”, in *Proceedings of the International Computer Music Conference (ICMC)*, pp. 431–434, La Habana, Cuba, 2001. 36, 38, 39, 40
- [78] Hank Heijink and Ruud G. J. Meulenbroek, “On the Complexity of Classical Guitar Playing: Functional Adaptations to Task Constraints.”, *Journal of Motor Behavior*, vol. 34, no. 4, pp. 339–351, Dec. 2002. 47
- [79] Matthew Hodgkinson, Joseph Timoney, and Victor Lazzarini, “A Model of Partial Tracks for Tension-Modulated Steel-String Guitar Tones”, in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFX-10)*, pp. 1–8, Graz, Austria, 2010. 23
- [80] Alex Hrybyk and Youngmoo Kim, “Combined Audio and Video for Guitar Chord Identification”, in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 159–164, Utrecht, Netherlands, 2010. 48, 49
- [81] Native Instruments, “SCARBEE PRE-BASS”, <http://www.native-instruments.com/#/de/products/producer/powerd-by-kontakt/scarbee-pre-bass/> (Accessed: 29.11.2013). 123

- [82] Ra Inta, *The Acoustics of the Steel String Guitar*, PhD thesis, The University of New South Wales, Sydney, Australia, 2007. 24
- [83] Jordi Janer, “Voice-controlled plucked bass guitar through two synthesis techniques”, in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pp. 132–135, Vancouver, Canada, 2005. 123, 125
- [84] E. V. Jansson, “Acoustics for the guitar player”, in *Function, Construction, and Quality of the Guitar*, E. V. Jansson, Ed., pp. 7–26. Royal Swedish Academy of Music, Stockholm, 1983. 14
- [85] Hanna Järveläinen, *Perception of Attributes in Real and Synthetic String Instrument Sounds*, PhD thesis, Helsinki University of Technology, 2003. 18, 24, 127
- [86] Hanna Järveläinen, Vesa Välimäki, and Matti Karjalainen, “Audibility of the timbral effects of inharmonicity in stringed instrument tones”, *Acoustics Research Letters Online*, vol. 2, no. 3, pp. 79, 2001. 23
- [87] Hanna Järveläinen, Tony Verma, and Vesa Välimäki, “Perception and Adjustment of Pitch in Inharmonic String Instrument Tones”, *Journal of New Music Research*, vol. 31, no. 4, pp. 311–319, 2003. 23
- [88] N. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, Englewood Cliffs, 1984. 126
- [89] Kristoffer Jensen, *Timbre Models of Musical Sounds*, PhD thesis, University of Copenhagen, 1999. 62
- [90] Richard Johnston, *The Bass Player Book: Equipment, Technique, Styles and Artists*, Backbeat Books, 2000. 11
- [91] Matti Karjalainen, Teemu Mäki-Patola, Aki Kanerva, Antti Huovilainen, and Pekka Jänis, “Virtual Air Guitar”, in *Proceedings of the 117th Audio Engineering Society (AES) Convention*, pp. 1–19, San Francisco, CA, USA, 2004. 22, 24, 25, 43, 125, 127
- [92] Matti Karjalainen, Vesa Välimäki, and Zoltán Jánosy, “Towards High-Quality Sound Synthesis of the Guitar and String Instruments”, in *Proceedings of the International Computer Music Conference (ICMC)*, pp. 56–63, Tokyo, Japan, 1993. 125
- [93] Matti Karjalainen, Vesa Välimäki, Tero Tolonen, and Tero Tolonen, “Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and Beyond”, *Computer Music Journal*, vol. 22, no. 3, pp. 17–32, 1998. 125, 131
- [94] Ioannis Karydis, Alexandros Nanopoulos, and Yannis Manolopoulos, “Symbolic musical genre classification based on repeating patterns”, in *Proceedings of the 1st Audio and Music Computing for Multimedia Workshop (AMCCM)*, pp. 53–58, Santa Barbara, CA, USA, 2006. ACM Press. 87, 88, 89, 90, 91

- [95] Christian Kehling, *Entwicklung eines parametrischen Instrumentencoders basierend auf Analyse und Re-Synthese von Gitarrenaufnahmen*, Diploma thesis, Technische Universität Ilmenau, 2013. 127
- [96] Chutisant Kerdvibulvech and Hideo Saito, “Vision-Based Detection of Guitar Players’ Fingertips Without Markers”, in *Proceedings of the 4th International Conference on Computer Graphics, Imaging and Visualization (CGIV)*, pp. 419–428, Bangkok, Thailand, 2007. 48, 49
- [97] Chutisant Kerdvibulvech and Hideo Saito, “Vision-Based Guitarist Fingering Tracking Using a Bayesian Classifier and Particle Filters”, *Advances in Image and Video Technology*, pp. 625–638, 2007. 48, 49
- [98] Frederick W. King, *Hilbert Transforms*: Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2009. 134
- [99] Tetsuro Kitahara, Yusuke Tsuchihashi, and Haruhiro Katayose, “Music Genre Classification and Similarity Calculation Using Bass-Line Features”, in *Proceedings of the 10th IEEE International Symposium on Multimedia*, pp. 574–579, Dec. 2008. 87, 88, 89, 90, 91
- [100] Anssi Klapuri, “Multiple Fundamental Frequency Estimation Based on Harmonicity and Spectral Smoothness”, *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003. 23
- [101] Anssi Klapuri, “Multiple Fundamental Frequency Estimation by Summing Harmonic Amplitudes”, in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pp. 216–221, 2006. 38, 39
- [102] Anssi Klapuri and Manuel Davy, Eds., *Signal Processing Methods for Music Transcription*, Springer Science+Business Media, 2006. 33, 35, 88
- [103] Patrick Kramer, *Entwicklung eines Verfahrens zur automatischen Klangsynthese von Bassnoten unter Berücksichtigung typischer Spieltechniken des E-Basses*, Diploma thesis, Technische Universität Ilmenau, 2011. 129, 131, 137, 139
- [104] Patrick Kramer, Jakob Abeßer, Christian Dittmar, and Gerald Schuller, “A Digital Waveguide Model of the Electric Bass Guitar Including Different Playing Techniques”, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 353–356, Kyoto, Japan, 2012. 121, 129, 137, 139, 140
- [105] Patrick Kramer, Jakob Abeßer, Christian Dittmar, and Gerald Schuller, “A Digital Waveguide Model of the Electric Bass Guitar including Different Playing Techniques - Sound Samples”, http://www.idmt.fraunhofer.de/en/business_units/smt/published_datasets.html (Accessed: 05.10.2014), 2012. 129
- [106] Arvindh Krishnaswamy and Julius O. Smith, “Inferring Control Inputs to an Acoustic Violin from Audio Spectra”, in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 3–6, Baltimore, MD, USA, 2003. 45, 46, 49, 50

- [107] Mikael Laurson, Jarmo Hiipakka, Cumhur Erkut, Matti Karjalainen, Vesa Välimäki, and Mika Kuuskankare, “From expressive notation to model-based sound synthesis: a case study of the acoustic guitar”, in *Proceedings of the 1999 International Computer Music Conference (ICMC)*, Beijing, China, 1999. 123, 125, 127
- [108] Mikael Laurson, Vesa Välimäki, and Henri Penttinens, “Simulating Idiomatic Playing Styles in a Classical Guitar Synthesizer: Rasgueado as a Case Study”, in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, pp. 1–4, Graz, Austria, 2010. 44, 125
- [109] Nelson Lee, Zhiyao Duan, and Julius O. Smith, “Excitation Signal Extraction for Guitar Tones”, in *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, 2007. 125
- [110] Nicolas Leroy, Emmanuel Flety, and Frederic Bevilacqua, “Reflective optical pickup for violin”, in *Proceedings of the International Conference on New interfaces for Musical Expression (NIME)*, pp. 204–207, Paris, France, 2006. 45
- [111] Thomas Lidy, Rudolf Mayer, Andreas Rauber, Pedro J. Ponce de Léon, Antonio Pertusa, and José Manuel Iñesta, “A Cartesian Ensemble of Feature Subspace for Music Categorization”, in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 279–284, Utrecht, Netherlands, 2010. 87, 90
- [112] Thomas Lidy, Andreas Rauber, Antonio Pertusa, and José Manuel Iñesta, “Improving Genre Classification by Combination of Audio and Symbolic Descriptors using a Transcription System”, in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pp. 61–66, Vienna, Austria, 2007. 87, 88, 89, 90, 91
- [113] Niklas Lindroos, Henri Penttinens, and Vesa Välimäki, “Parametric Electric Guitar Synthesis”, *Computer Music Journal*, vol. 35, no. 3, pp. 18–27, 2011. 13, 125
- [114] Alex Loscos, Ye Wang, and Wei Jie Jonathan Boo, “Low Level Descriptors for Automatic Violin Transcription”, in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pp. 164–167, Victoria, Canada, 2006. 35, 45, 46
- [115] Robert Macrae and Simon Dixon, “Guitar Tab Mining, Analysis and Ranking”, in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 453–458, Miami, USA, 2011. 26
- [116] Esteban Maestre, Merlijn Blaauw, Jordi Bonada, Enric Guaus, and Alfonso Pérez, “Statistical Modeling of Bowing Control Applied to Violin Sound Synthesis”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 855–871, 2010. 45
- [117] Esteban Maestre, Jordi Bonada, Merlijn Blaauw, and Alfonso Pérez, “Acquisition of Violin Instrumental Gestures using a Commercial EMF Tracking Device”, in *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, 2007. 45

- [118] Akira Maezawa, Katsutoshi Itoyama, Toru Takahashi, Tetsuya Ogata, and Hiroshi G. Okuno, “Bowed String Sequence Estimation of a Violin Based on Adaptive Audio Signal Classification and Context-Dependent Error Correction”, in *Proceedings of the 11th IEEE International Symposium on Multimedia*, pp. 9–16, San Diego, California, USA, 2009. Ieee. 45, 47, 49, 50
- [119] Cory McKay, *Automatic Music Classification with jMIR*, PhD thesis, McGill University, Montreal, Canada, 2010. 88
- [120] Cory Mckay and Ichiro Fujinaga, “jSymbolic”, <http://jmir.sourceforge.net/jSymbolic.html> (Accessed: 29.11.2013). 107
- [121] Cory McKay and Ichiro Fujinaga, “Automatic Genre Classification using Large High-level Musical Feature Sets”, in *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, pp. 525–530, Barcelona, Spain, 2004. 87, 89, 90, 91
- [122] Cory Mckay and Ichiro Fujinaga, “jSymbolic : A Feature Extractor for MIDI Files Designing High-Level Features”, in *Proceedings of the International Computer Music Conference (ICMC)*, pp. 302–305, Copenhagen, Denmark, 2006. 107, 108
- [123] Keith McMillen, David Wessel, and Matthew Wright, “The ZIPI music parameter description language”, *Computer Music Journal*, vol. 18, no. 5, pp. 52–73, 1994. 127
- [124] Todd K. Moon, “The Expectatio Maximization Algorithm”, *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996. 58
- [125] Daniel Müllensiefen and Klaus Frieler, “Optimizing Measures of Melodic Similarity for the Exploration of a Large Folk Song Database”, in *Proceedings of the 5th International Conference in Music Information Retrieval (ISMIR)*, pp. 274–280, Barcelona, Spain, 2004. 89
- [126] Meinard Müller, *Information Retrieval for Music and Motion*, Springer-Verlag New York, Inc., 2007. 39
- [127] Meinard Müller, Daniel P. W. Ellis, Anssi Klapuri, and Gaël Richard, “Signal Processing for Music Analysis”, *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1088–1110, 2011. 88
- [128] Jonathan Carey Norton, *Motion Capture to Build a Foundation for a Computer-controlled Instrument by Study of Classical Guitar Performance*, PhD thesis, Stanford University, 2008. 43
- [129] Paul D. O’Grady and Scott T. Rickard, “Automatic Hexaphonic Guitar Transcription Using Non-Negative Constraints”, in *Proceedings of the IET Irish Signals and Systems Conference (ISSC)*, Dublin, Ireland, 2009. 48, 49
- [130] Nobutaka Ono, Kenichi Miyamoto, Hirokazu Kameoka, and Shigeki Sagayama, “A Real-time Equalizer of Harmonic and Percussive Componets in Music Signals”, in *Proceedings*

- of the 9th International Conference on Music Information Retrieval (ISMIR)*, pp. 139–144, Philadelphia, PA, USA, 2008. 37
- [131] Nicola Orio, “The timbre space of the classical guitar and its relationship with the plucking techniques”, in *Proceedings of the International Computer Music Conference (ICMC)*, Beijing, China, 1999. 44
- [132] Nicola Orio, Antonio Rodà, and Antonio Rod, “A Measure of Melodic Similarity based on a Graph Representation of the Music Structure”, in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 543–548, Kobe, Japan, 2009. 90
- [133] Tan Hakan Özslan and Josep Lluis Arcos, “Legato and Glissando Identification in Classical Guitar”, in *Proceedings of Sound and Music Computing Conference (SMC), Barcelona, Spain*, pp. 457–463, 2010. 44, 45
- [134] Tan Hakan Özslan, Enric Guaus, Eric Palacios, and Josep Lluis Arcos, “Attack Based Articulation Analysis of Nylon String Guitar”, in *Proceedings of the 7th International Symposium on Computer Music Modeling and Retrieval (CMMR)*, pp. 285–298, Málaga, Spain, 2010. 44, 45
- [135] Jyri Pakarinen, “Physical Modeling of Flageolet Tones in String Instruments”, in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 4–8, Antalya, Turkey, 2005. 125, 132
- [136] Marco Paleari, Benoit Huet, Antony Schutz, and Dirk Slock, “A Multimodal Approach to Music Transcription”, in *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP)*, pp. 93–96, San Diego, California, USA, 2008. 48, 49
- [137] Geoffroy Peeters and Xavier Rodet, “Hierarchical gaussian tree with inertia ratio maximization for the classification of large musical instruments databases”, in *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx)*, pp. 1–6, London, UK, 2003. 27
- [138] Henri Penttinen, Jaakko Siiskonen, and Vesa Välimäki, “Acoustic Guitar Plucking Point Estimation in Real Time”, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 209–212, Philadelphia, PA, USA, 2005. 47
- [139] Henri Penttinen and Vesa Välimäki, “A time-domain approach to estimating the plucking point of guitar tones obtain with an under-saddle pickup”, *Applied Acoustics*, vol. 65, no. 12, pp. 1207–1220, Dec. 2004. 47
- [140] Carlos Pérez-Sancho, Pedro J. Ponce de León, José Manuel Iñesta, and Pedro J. Ponce de León, “A Comparison of Statistical Approaches to Symbolic Genre Recognition”, in *Proceedings of the International Computer Music Conference (ICMC)*, pp. 545–550, New Orleans, USA, 2006. 87, 88, 89, 90, 91
- [141] Graham E. Poliner, Daniel P. W. Ellis, Andreas F. Ehmann, Emilia Gómez, Sebastian Streich, and Beesuan Ong, “Melody Transcription From Music Audio - Approaches and

- Evaluation”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1247–1256, 2007. 40
- [142] Heiko Purnhagen and Nikolaus Meine, “HILN - The MPEG-4 Parametric Audio Coding Tools”, in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 201–204, Geneva, Switzerland, 2000. 127
- [143] Lawrence R. Rabiner, Michael J. Cheng, Aaron E. Rosenberg, and Carol A. McGonegal, “A Comparative Performance Study of Several Pitch Detection Algorithms”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 399–418, 1976. 35
- [144] Stanislaw A. Raczyński, Nobutaka Ono, and Shigeki Sagayama, “Multipitch analysis with harmonic nonnegative matrix approximation”, in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, 2007. 39
- [145] Aleksander Radisljević and Peter Driessens, “Path difference learning for guitar fingering problem”, in *Proceedings of the International Computer Music Conference (ICMC)*, Miami, USA, 2004. 47
- [146] Erhard Rank and Gernot Kubin, “A waveguide model for slapbass synthesis”, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. 443–446, Munich, Germany, 1997. 125
- [147] Jukka Rauhala, Heidi-Maria Lehtonen, and Vesa Välimäki, “Fast automatic inharmonicity estimation algorithm”, *The Journal of the Acoustical Society of America*, vol. 121, no. 5, pp. 184–189, 2007. 23
- [148] Jukka Rauhala and Vesa Välimäki, “Dispersion modeling in waveguide piano synthesis using tunable allpass filters”, in *Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-06)*, pp. 71–76, Montréal, Canada, 2006. 134
- [149] 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)*, pp. 1–4, Ann Arbor, Michigan, USA, 2012. 43, 44
- [150] H.-J. Reznicek, *I'm Walking - Jazz Bass*, AMA, 2001. 47, 105
- [151] Partners In Rhyhme, “Royalty Free Music & Sound Effects”, <http://www.partnersinrhyme.com/midi/index.shtml> (Accessed: 29.11.2013). 136
- [152] Matti Ryynänen and Anssi Klapuri, “Automatic Bass Line Transcription from Streaming Polyphonic Audio”, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP')*, pp. 1437–1440, Honolulu, Hawaii, USA, 2007. 37, 38, 39, 41, 76
- [153] Matti P. Ryynänen and Anssi Klapuri, “Automatic transcription of melody, bass line, and chords in polyphonic music”, *Computer Music Journal*, vol. 32, pp. 72–86, Apr. 2008. 38, 39, 41, 75, 77

- [154] Pasi Saari, Tuomas Eerola, and Olivier Lartillot, “Generalizability and Simplicity as Criteria in Feature Selection: Application to Mood Classification in Music”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1802–1812, Aug. 2011. 27
- [155] Justin Salamon and Emilia Gómez, “A Chroma-Based Salience Function for Melody and Bass Line Estimation From Music Audio Signals”, in *Proceedings of the 6th Sound and Music Computing Conference (SMC)*, pp. 23–25, Porto, Portugal, 2009. 37, 38, 39, 41, 75, 77
- [156] Justin Salamon, Emilia Gómez, Daniel P.W. Ellis, and Gael Richard, “Melody Extraction from Polyphonic Music Signals : Approaches, Applications and Challenges”, *IEEE Signal Processing Magazine*, 2013. 2, 76
- [157] Umut Simsekli, “Automatic Music Genre Classification Using Bass Lines”, in *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, pp. 4137–4140, Istanbul, Turkey, Aug. 2010. 87, 88, 90, 91
- [158] Julius O. Smith, “Physical Modeling Using Digital Waveguides”, *Computer Music Journal*, vol. 16, no. 4, pp. 74, 1992. 124
- [159] Spectrasonics, “Trillian”, <http://www.spectrasonics.net/products/trillian.php> (Accessed: 29.11.2013). 123
- [160] Michael Stein, “Automatic Detection of Multiple, Cascaded Audio Effects in Guitar Recordings”, in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx)*, pp. 4–7, Graz, Austria, 2010. 13
- [161] Michael Stein, Jakob Abeßer, Christian Dittmar, and Gerald Schuller, “Automatic Detection of Audio Effects in Guitar and Bass Recordings”, in *Proceedings of the 128th Audio Engineering Society (AES) Convention*, pp. 522–533, London, UK, 2010. 13, 72
- [162] Mark Sterling, Xiaoxiao Dong, and Mark Bocko, “Pitch bends and tonguing articulation in clarinet physical modeling synthesis”, in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 289–292, Taipei, Taiwan, 2009. 124
- [163] Bob L. Sturm, “A Survey of Evaluation in Music Genre Recognition”, in *Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval*, pp. 1–21, Copenhagen, Denmark, 2012. 87, 88, 89, 115
- [164] Tero Tolonen, Vesa Välimäki, and Matti Karjalainen, “Evaluation of Modern Sound Synthesis Methods”, Tech. Rep. March, Helsinki University of Technology, 1998. 123
- [165] Heber Manuel Pérez Torres, *An Analysis to the Guitar Lab’s gesture acquisition prototype with the aim of improving it.*, Master thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2011. 43
-

- [166] Caroline Traube and Philippe Depalle, “Extraction of the Excitation Point Location on a String using Weighted Least-Square Estimation of a Comb Filter Delay”, in *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx)*, vol. 5, pp. 6–9, London, UK, 2003. 44, 47
- [167] Caroline Traube and Julius O. Smith, “Extracting the Fingering and the Plucking Points on a Guitar String from a Recording”, in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 2–5, New Paltz, NY, US, 2001. 19, 48, 49
- [168] Lutz Trautmann and Rudolf Rabenstein, “Stable Systems for Nonlinear Discrete Sound Synthesis with the Functional Transformation Method”, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, pp. 1861–1864, Orlando, Florida, USA, 2002. 125
- [169] Yusuke Tsuchihashi, Tetsuro Kitahara, and Haruhiro Katayose, “Using Bass-Line Features for Content-Based MIR”, in *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pp. 620–625, Philadelphia, Pennsylvania USA, 2008. 87, 88, 89, 91
- [170] Emiru Tsunoo, Nobutaka Ono, and Shigeki Sagayama, “Musical Bass-Line Pattern Clustering and its Application to Audio Genre Classification”, in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 219–224, Kobe, Japan, 2009. 37, 87, 89, 90, 91
- [171] Emiru Tsunoo, George Tzanetakis, Nobutaka Ono, and Shigeki Sagayama, “Beyond Timbral Statistics: Improving Music Classification Using Percussive Patterns and Bass Lines”, *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 1003–1014, 2011. 38, 115, 117
- [172] Rainer Typke, Panos Giannopoulos, Remco C. Veltkamp, Frans Wiering, and René van Oostrum, “Using Transportation Distances for Measuring Melodic Similarity”, in *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, pp. 107–114, Baltimore, Maryland, USA, 2002. 89
- [173] George Tzanetakis and Perry Cook, “Musical genre classification of audio signals”, *IEEE Transactions on speech and audio processing*, vol. 10, no. 5, pp. 293–302, 2002. 91
- [174] George Tzanetakis, Andrey Ermolinsky, and Perry Cook, “Pitch Histograms in Audio and Symbolic Music Information Retrieval”, in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR)*, Paris, France, 2002. 87, 88, 90, 91
- [175] Yasunori Uchida and Shigeo Wada, “Melody and bass line estimation method using audio feature database”, in *Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Shaanxi, China, 2011. 37, 38, 41

- [176] Vesa Välimäki, Jyri Huopaniemi, Matti Karjalainen, and Zoltán Jánosy, “Physical Modeling of Plucked String Instruments with Application to Real-Time Sound Synthesis”, *Journal of the Audio Engineering Society*, vol. 44, no. 5, pp. 331–335, 1996. 125
- [177] Vesa Välimäki, Jyri Pakarinen, Cumhur Erkut, and Matti Karjalainen, “Discrete-time modelling of musical instruments”, *Reports on Progress in Physics*, vol. 69, no. 1, pp. 1–78, Jan. 2006. 125
- [178] Vesa Välimäki, Henri Penttinen, Jonte Knif, Mikael Laurson, and Cumhur Erkut, “Sound Synthesis of the Harpsichord Using a Computationally Efficient Physical Model”, *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 7, pp. 934–948, 2004. 125
- [179] Maarten van Walstijn, “Parametric FIR Design of Propagation Loss Filters in Digital Waveguide String Models”, *IEEE Signal Processing Letters*, vol. 17, no. 9, pp. 795–798, 2010. 129, 133
- [180] Vladimir N. Vapnik, *Statistical learning theory*, Wiley New York, 1st edition, 1998. 28
- [181] Emmanuel Vincent, Nancy Bertin, and Roland Badeau, “Harmonic and Inharmonic Non-negative Matrix Factorization for Polyphonic Pitch Transcription”, in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 109–112, Las Vegas, USA, 2008. 23, 39
- [182] Paul Westwood, *Bass Bible*, Ama Verlag, 2000. 33, 36, 47, 66, 105
- [183] Jun Yin, Ye Wang, and David Hsu, “Digital Violin Tutor: An Integrated System for Beginning Violin Learners”, in *Proceedings of the ACM Multimedia*, pp. 976–985, Singapore, Singapore, 2005. 35, 45
- [184] Bingjun Zhang, Jia Zhu, Ye Wang, and Wee Kheng Leow, “Visual analysis of fingering for pedagogical violin transcription”, in *Proceedings of the 15th ACM Conference on Multimedia*, pp. 521–524, Augsburg, Germany, 2007. ACM Press. 49, 50
- [185] Harry Zhang, “The Optimality of Naive Bayes.”, in *Proceedings of the FLAIRS Conference*, pp. 562–567, Miami, USA, 2004. 28
- [186] Kai-Erik Ziegenrücker, Wieland Ziegenrücker, and Peter Wicke, *Handbuch der populären Musik: Geschichte, Stile, Praxis, Industrie*, Schott, Mainz, 2007. 107