Interactive Signal Processing Tools for Analyzing Multitrack Singing Voice Recordings

Interaktive Signalverarbeitungswerkzeuge zur Analyse mehrspuriger Gesangsaufnahmen

Dissertation

Der Technischen Fakultät

der Friedrich-Alexander-Universität Erlangen-Nürnberg

zur

Erlangung des Doktorgrades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von

Sebastian Rosenzweig

aus

Erlangen

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:	29. Juni 2022
1. Gutachter:	Prof. Dr. rer. nat. Meinard Müller
2. Gutachterin:	Prof. Dr. Anja Volk

Abstract

Polyphonic vocal music is an integral part of music cultures around the world. For studying performance aspects and cultural differences, the analysis of recorded audio material has become of increasing importance. This thesis contributes several computational tools for processing, analyzing, and exploring singing voice recordings using methods from signal processing, computer science, and music information retrieval (MIR). First, we develop an approach for applying time-varying pitch shifts to audio signals based on non-linear time-scale modification (TSM) and resampling techniques. We show that our method can be used to adjust intonation (fine-tuning of pitch) in vocal recordings, e.g., in postproduction contexts. Computational analysis of polyphonic vocal music typically requires annotations of the singers' fundamental frequency (F0) trajectories, which are labor-intensive to generate and may not be available for a particular recording collection. As a second contribution, we present an approach to assess the reliability of automatically extracted F0-estimates by fusing the outputs of several F0-estimation algorithms. In this way, our approach enables the analysis and exploration of large unlabeled audio collections. One major challenge for computational analysis of polyphonic singing constitute stylistic elements such as pitch slides and pitch drifts, which can introduce blurring in analysis results. As a third contribution of this thesis, we present computational tools for handling such peculiarities. In particular, we develop musically motivated filtering techniques to detect stable regions in F0-trajectories and compensate for pitch drifts. Furthermore, our tools offer interactive feedback mechanisms that allow domain experts to incorporate musical knowledge. Development and evaluation of computational tools for analyzing polyphonic singing typically require suitable multitrack recordings with one or several tracks per voice, e.g., obtained from close-up microphones attached to a singer's head and neck. However, such recordings are challenging to produce and thus of limited availability. As an additional contribution of this thesis, we introduce carefully organized and annotated multitrack research corpora of Western choral music and traditional Georgian vocal music, which are publicly accessible through interactive interfaces. Furthermore, considering these two culturally different forms of vocal music as concrete application scenarios, we evaluate our interactive computational tools and demonstrate their potential for corpus-driven research in the field of computational ethnomusicology.

Zusammenfassung

Mehrstimmige Vokalmusik ist ein wesentlicher Bestandteil von Musikkulturen auf der ganzen Welt. Bei der Untersuchung von aufführungs- und kulturspezifischen Aspekten spielt die Analyse von aufgenommenem Audiomaterial eine zunehmend wichtige Rolle. Diese Dissertation befasst sich mit der Entwicklung von computergestützten Werkzeugen zur Verarbeitung, Analyse und Erforschung von Gesangsaufnahmen mittels Techniken der Signalverarbeitung und des Music Information Retrieval (MIR). Als ersten Beitrag wird ein Ansatz zur zeitabhängigen Tonhöhenkorrektur basierend auf Zeitdehnungs- und Resampling-Verfahren vorgestellt. Zudem wird gezeigt, wie die Techniken zur Anpassung der Intonation (Feinabstimmung der Tonhöhe) in Vokalaufnahmen, z. B. in der Postproduktion, verwendet werden können. Die computergestützte Analyse mehrstimmiger Gesänge benötigt in der Regel Annotationen der Fundamentalfrequenz-Trajektorien (oder F0-Trajektorien) aller Stimmen. Meist stehen solche Annotationen nicht oder nur begrenzt zur Verfügung, da ihre Erstellung sehr arbeits- und zeitintensiv ist. Als zweiten Beitrag wird ein Ansatz zur Bewertung der Zuverlässigkeit von F0-Schätzungen präsentiert, der die Ausgaben verschiedener automatischer F0-Schätzalgorithmen kombiniert. Der vorgestellte Ansatz ermöglicht es, große, nicht-annotierte Audiosammlungen auf Basis von verlässlichen F0-Schätzungen zu analysieren. Eine besondere Herausforderung für die computergestützte Analyse mehrstimmiger Gesänge stellen kontinuierliche Tonhöhenänderungen (sog. Glissandi oder Portamenti) und sinkende oder fallende Intonation über die Dauer eines Stücks dar. Diese musikalischen Phänomene können zu Unschärfe in den Analyseergebnissen führen. Als dritten Beitrag dieser Dissertation werden Filtertechniken für stabile Regionen und bestimmte harmonische Intervalle in F0-Trajektorien entwickelt, um diesen Phänomenen gerecht zu werden. Darüber hinaus bieten die entwickelten Werkzeuge interaktive Feedback-Mechanismen, die es Domänenexperten ermöglichen, musikalisches Fachwissen in die Analyse einfließen zu lassen. Entwicklung und Evaluation solcher Werkzeuge erfordern in der Regel geeignete Mehrspuraufnahmen mit einer oder mehreren Spuren pro Stimme. Diese können z. B. durch Kehlkopf-Mikrofone erzeugt werden, die am Hals der Sängerinnen und Sänger befestigt sind. Als zusätzlichen Beitrag dieser Arbeit werden sorgfältig organisierte und annotierte Mehrspur-Datensätze westlicher Chormusik und traditioneller georgischer Vokalmusik präsentiert, die über interaktive Schnittstellen zu Forschungszwecken öffentlich zugänglich gemacht sind. Die zwei kulturell verschiedenen Arten von Vokalmusik dienen als konkrete Anwendungsszenarien und zur Evaluation der vorgestellten Techniken. Zudem wird das Potenzial der entwickelten Werkzeuge für die audiobasierte Forschung im Bereich der computergestützten Musikethnologie aufgezeigt.

Contents

1 Introduction

The human voice can produce a remarkable range of sounds, including speech, laughing, crying, humming, or singing. Voice sounds can be “as rich and complex as those of conventional musical instruments” [200, p. 95]. One of the oldest forms of human music-making and a central part of music cultures worldwide is group singing [215]. Singing in different parts or melodic lines is typically referred to as *polyphonic vocal music* [94]. Yet, many facets of this centuries-old cultural asset, e.g., performance practices and cultural peculiarities, are not fully explored and understood. Musicological research on polyphonic vocal music is often conducted based on notated musical scores, which are obtained, e.g., by manually transcribing vocal performances. Such transcription approaches are limited since they are susceptible to subjectivity and reproducibility issues. Furthermore, important tonal cues and performance aspects may get lost in the transcription process [125, 203]. Therefore, when researching polyphonic vocal music, the analysis of audio recordings seems inevitable. This thesis presents computational tools for analyzing audio recordings of polyphonic vocal music using techniques from signal processing, computer science, and music information retrieval (MIR). To bridge the gap between engineering and musicology [73, 205, 212], our tools go beyond “black box” algorithms: firstly, by providing accessible and musically interpretable parameters, which allow a user to include musical domain knowledge, and secondly, by offering interactive feedback mechanisms (e.g., in the form of visualizations) to understand and guide the algorithms intuitively.

Central to the computational analysis of polyphonic vocal recordings is determining the fundamental frequencies (F0s) over time that correspond to the sung melodic lines. The obtained sequences of F0-values are also referred to as F0-trajectories. Research on F0-estimation for monophonic recordings (with only one singing voice present) is advanced. For instance, there exist many conceptually different estimation algorithms [27, 48, 68, 97, 109, 169] as well as semi-automatic annotation tools that allow a user to correct automatically estimated F0-trajectories [111, 126]. In the context of polyphonic singing, recording singers separately typically requires extensive recording infrastructure and is also not desirable for musical reasons. One option is to jointly estimate the F0-trajectories of all singers from a polyphonic mixture directly, also referred to as multiple-F0 or multipitch estimation. This task is particularly challenging for polyphonic vocal music and is still subject to ongoing research [9, 39, 42, 115]. To circumvent the challenges of multipitch estimation, following up on previous research on polyphonic vocal music [86, 173], we use multitrack recordings obtained from close-up microphones, which are attached to a singer’s head and neck.

In particular, we exploit so-called larynx or throat microphones that capture the vibrations of the human throat and can produce recordings with few cross-talk of individual singers in polyphonic performances.

In this thesis, we consider two musical application scenarios that motivate the development of computational tools. In the first scenario, we deal with Western choral music. Choir singers need to have an exact control over their voice and intonation (fine-tuning of the pitch). In particular, singers have to constantly adjust the intonation to stay in tune relative to the other singers. In the case of amateur *a cappella singing* (singing without instrumental accompaniment), one often observes intonation deficiencies, e.g., in the form of intonation drifts throughout a performance. To analyze such musical aspects using MIR techniques, we recorded several amateur a cappella ensembles and created an annotated multitrack research corpus called *Dagstuhl ChoirSet*. Furthermore, we present a technique for applying adaptive, time-varying pitch shifts to audio signals using time-scale modification (TSM) and resampling techniques, which can support sound engineers with editing choir recordings in postproduction settings.

In the second musical scenario, we consider the analysis of traditional three-part Georgian singing, which is listed as "Intangible Cultural Heritage of Humanity" by the UNESCO[1]. The tonal organization of traditional Georgian music has been at the center of attention of ethnomusicologists for many decades (cf. [93, p. 101 ff.]) and is still a matter of intense discussion. Parts of this thesis have been conducted within an interdisciplinary research project of computer scientists and ethnomusicologists, aiming to advance research on Georgia's cultural treasure using computational methods. Being an orally transmitted culture, most sources are available as field recordings. A fundamental, though often underrated task in MIR and computational musicology is the preparation and annotation of reusable research corpora [188]. Such annotations often need to be created in labor-intensive annotation processes conducted by domain experts. As part of this thesis, we present a corpus based on historic tape recordings of the former Georgian master chanter Artem Erkomaishvili, which includes carefully crafted F0-, structure, and note onset annotations for all 101 recordings. One major challenge for analyzing melodic and harmonic properties of the Erkomaishvili corpus and traditional Georgian vocal music in general constitute pitch slides, which introduce blurring in analysis results. As one contribution, we present two conceptually different approaches for detecting stable regions in F0-trajectories based on morphological filters and binary time–frequency masks that alleviate such issues. For extensive audio collections (such as a collection of 216 multitrack field recordings from Georgia, also referred to as the GVM collection [180]) expert annotations may not be available for all recordings. To enable analysis and exploration of unlabeled audio collections, we introduce an approach to assess the reliability of automatically extracted F0-trajectories by fusing the outputs of several F0-estimation algorithms. Finally, in a case study on recordings of traditional Georgian funeral songs, we present computational tools for determining stable, note-like objects and correcting pitch drifts in F0-trajectories using filtering techniques for musically important harmonic intervals. By applying our tools for computing pitch inventories, we show their potential for ethnomusicological research.

[1] https://ich.unesco.org/en/RL/georgian-polyphonic-singing-00008

In summary, rather than presenting end-to-end solutions, this thesis introduces interactive computational tools that support domain experts in processing and analyzing polyphonic vocal music recordings. Furthermore, through explicit mathematical modeling of algorithms and approaches as well as the visualization of processing steps and analysis results, we support interdisciplinary exchange between computer science and musicology. Finally, to ensure sustainability of our research, we make our corpora and tools publicly available for future studies.

1.1 Structure and Main Contributions of this Thesis

This thesis is organized as follows. In Chapter 2, we summarize some fundamentals of signal processing, including audio representations, multitrack recording techniques, and approaches for fundamental frequency estimation.

The main body of this thesis is subdivided into two parts covering the two musical scenarios considered. Part I starts with Chapter 3, where we introduce a novel multitrack dataset of a cappella choral singing named Dagstuhl ChoirSet (DCS). We created DCS “from scratch” by recording amateur ensembles, organizing the recorded data, and generating F0-annotations, beat annotations, and time-aligned score representations. Furthermore, we developed interactive web-based interfaces to access the dataset. We show the potential of DCS for MIR research in two case studies on measuring choral intonation and multiple-F0 estimation.

In Chapter 4, we present an approach for applying adaptive (time-varying) pitch shifts to audio recordings. Our method is based on time-scale modification (TSM) and resampling techniques. Besides providing a mathematical description of our approach, we present an open-source toolbox that includes Python implementations of our approach and several TSM algorithms. Furthermore, we show the potential of our method for adjusting intonation in a cappella recordings using our DCS dataset.

Part II begins with Chapter 5, where we introduce a carefully curated corpus of traditional Georgian vocal music named Erkomaishvili dataset. To create this dataset, we collated and organized historic tape recordings of the former master chanter Artem Erkomaishvili, digitized existing transcriptions, and annotated F0-trajectories and note onsets. In the context of case studies on harmonic intervals and pitch inventories, we demonstrate the potential of the Erkomaishvili dataset for research on traditional Georgian vocal music.

In Chapter 6, we introduce two approaches for removing pitch slides and other frequency fluctuations from F0-trajectories: the first algorithm uses morphological operations inspired by image processing, and the second one is based on suitably defined binary time–frequency masks. To avoid undesired distortions in subsequent analysis steps, both approaches keep the original F0-values unmodified while only removing F0-values in unstable trajectory regions. We evaluate both approaches against manually annotated stable regions and discuss their potential in the context of interval analysis for traditional Georgian vocal music.

In Chapter 7, we present an approach for assessing the reliability of automatically computed F0-estimates. To this end, we propose three reliability indicators that fuse the outputs of several algorithms. Besides providing a mathematical description of the indicators, we analyze the indicators' behavior using a set of annotated vocal F0-trajectories. Furthermore, we show the potential of the proposed indicators for exploring unlabeled audio collections.

In Chapter 8, we present an annotated multitrack corpus based on recordings of Georgian funeral songs (also called Zär) and interactive tools for explorative corpus-driven research. Following up on Chapter 6, we present a tool to determine stable, note-like events in F0-trajectories. Furthermore, we introduce a method for determining and compensating pitch drifts in F0-trajectories based on musically informed interval filtering techniques. By conducting a case study on pitch inventories (pitch-class histograms) of our Zär corpus, we demonstrate the potential of our tools for computational ethnomusicology.

Finally, we conclude this thesis in Chapter 9 with a summary and directions of future work.

1.2 Publications Related to Ph.D. Thesis

Major parts of this thesis have previously been published in peer-reviewed journals and conference proceedings in the field of audio signal processing and music information retrieval ([158, 159, 161, 162, 163].), except for [164], which has been submitted and is currently under review. All publications related to this thesis are listed in the following:

[158] Sebastian Rosenzweig, Frank Scherbaum, and Meinard Müller. Detecting stable regions in frequency trajectories for tonal analysis of traditional Georgian vocal music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 352–359, Delft, The Netherlands, 2019. doi: 10.5281/zenodo.3527816

[159] Sebastian Rosenzweig, Helena Cuesta, Christof Weiß, Frank Scherbaum, Emilia Gómez, and Meinard Müller. Dagstuhl ChoirSet: A multitrack dataset for MIR research on choral singing. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 3(1):98–110, 2020. doi: 10.5334/tismir.48

[161] Sebastian Rosenzweig, Frank Scherbaum, David Shugliashvili, Vlora Arifi-Müller, and Meinard Müller. Erkomaishvili Dataset: A curated corpus of traditional Georgian vocal music for computational musicology. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 3(1):31–41, 2020. doi: 10.5334/tismir.44

[162] Sebastian Rosenzweig, Frank Scherbaum, and Meinard Müller. Reliability assessment of singing voice F0-estimates using multiple algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 261–265, Toronto, Canada, 2021. doi: 10.1109/ICASSP39728.2021.9413372

[163] Sebastian Rosenzweig, Simon Schwär, Jonathan Driedger, and Meinard Müller. Adaptive pitch-shifting with applications to intonation adjustment in a cappella recordings. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 121–128, Vienna, Austria, 2021

[164] Sebastian Rosenzweig, Frank Scherbaum, and Meinard Müller. Computer-assisted analysis of field recordings: A case study of Georgian funeral songs. *ACM Journal on Computing and Cultural Heritage (JOCCH)*, 2022. to appear

1.3 Additional Publications

Along with the main publications, Sebastian Rosenzweig contributed to the development of several web-based interfaces ([160, 179, 221]) and studies on intonation analysis of choral singing ([185, 214]). As part of the interdisciplinary collaboration with ethnomusicologists, he also contributed to several papers with a musicological focus ([126, 176, 177, 180, 181]). Finally, he helped to develop educational material for a preparation course on Python programming ([121]). These additional publications are listed in the following.

[160] Sebastian Rosenzweig, Lukas Dietz, Johannes Graulich, and Meinard Müller. TuneIn: A web-based interface for practicing choral parts. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Montreal, Canada, 2020

[179] Frank Scherbaum, Sebastian Rosenzweig, Meinard Müller, Daniel Vollmer, and Nana Mzhavanadze. Throat microphones for vocal music analysis. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018

[221] Frank Zalkow, Sebastian Rosenzweig, Johannes Graulich, Lukas Dietz, El Mehdi Lemnaouar, and Meinard Müller. A web-based interface for score following and track switching in choral music. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018

[185] Simon Schwär, Sebastian Rosenzweig, and Meinard Müller. A differentiable cost measure for intonation processing in polyphonic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 626–633, Online, 2021. doi: 10.5281/zenodo.5624601

[214] Christof Weiß, Sebastian J. Schlecht, Sebastian Rosenzweig, and Meinard Müller. Towards measuring intonation quality of choir recordings: A case study on Bruckner's Locus Iste. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 276–283, Delft, The Netherlands, 2019. doi: 10.5281/zenodo.3527798

[126] Meinard Müller, Sebastian Rosenzweig, Jonathan Driedger, and Frank Scherbaum. Interactive fundamental frequency estimation with applications to ethnomusicological research. In *Proceedings of the AES International Conference on Semantic Audio*, pages 186–193, Erlangen, Germany, 2017

[176] Frank Scherbaum, Meinard Müller, and Sebastian Rosenzweig. Analysis of the Tbilisi State Conservatory recordings of Artem Erkomaishvili in 1966. In *Proceedings of the International Workshop on Folk Music Analysis (FMA)*, pages 29–36, Málaga, Spain, 2017

[177] Frank Scherbaum, Meinard Müller, and Sebastian Rosenzweig. Rechnergestützte Musikethnologie am Beispiel historischer Aufnahmen mehrstimmiger georgischer Vokalmusik. In *Proceedings of the Jahrestagung der Gesellschaft für Informatik (GI)*, pages 163–175, Chemnitz, Germany, 2017

[180] Frank Scherbaum, Nana Mzhavanadze, Sebastian Rosenzweig, and Meinard Müller. Multi-media recordings of traditional Georgian vocal music for computational analysis. In *Proceedings of the International Workshop on Folk Music Analysis (FMA)*, pages 1–6, Birmingham, UK, 2019

[181] Frank Scherbaum, Nana Mzhavanadze, Simha Arom, Sebastian Rosenzweig, and Meinard Müller. *Tonal Organization of the Erkomaishvili Dataset: Pitches, Scales, Melodies and Harmonies*. Universitätsverlag Potsdam, 2020. doi: 10.25932/publishup-47614

[121] Meinard Müller and Sebastian Rosenzweig. PCP notebooks: A preparation course for Python with a focus on signal processing. *Journal of Open Source Education (JOSE)*, 5(47):148:1–5, 2022. doi: 10.21105/jose.00148

1.4 Acknowledgments

Almost 4.5 years and a pandemic later, it is time to say thanks to all people that accompanied me on this journey. First and foremost, I would like to express my deep gratitude to my supervisor and mentor Meinard Müller. He was the one who first brought me to the idea of doing a Ph.D.—something that I had not considered for a long, long time. For all these years, he supported me overarchingly, and helped me grow scientifically and personally. Thank you, Meinard, for your time, extensive feedback, and valuable scientific advice.

Furthermore, I would like to thank Frank Scherbaum for the great collaboration throughout all these years and for sharing his fascination with traditional Georgian singing. Many thanks, Frank, you are a neverending volcano of ideas, curiosity, and passion, which is genuinely inspiring.

A special thanks goes to Anja Volk, who took the time to review this thesis.

Over the years, I had the pleasure of working with many talented and friendly people, which made this Ph.D. an enjoyable and enriching experience. Most importantly, I would like to thank the members of the "GroupMM" (in alphabetical order): Vlora Arifi-Müller, Stefan Balke, Christian Dittmar, Jonathan Driedger, Michael Krause, Patricio López-Serrano, Peter Meier, Yigitcan Özer, Thomas Prätzlich, Hendrik Schreiber, Christof Weiß, Frank Zalkow, and Julia Zalkow. I am very happy to have been part of this great and mutually supporting team. Second, I thank my students María Acuña, Judith Bauer, Benjamin Brunner, Satyaki Chatterjee, Moumita Chakraborty, Dipanjan Datta Roy, Lukas Dietz, Michael Fast, Miguel Gonzalez, Kathrin Hedrich, El Mehdi Lemnaouar, Vojtěch Pešek, Adithyan Ramadoss, Florian Schuberth, Simon Schwär, Juan Serrano, Shahmeer Shahnawaz, Edgar Suarez, and Sri Harsha Vadlamudi. I hope you learned as much from me as I from supervising you.

I would like to give a special mention to the people, who invested innumerable hours creating annotations for the research corpora presented in this thesis, in particular, Vlora Arifi-Müller, Frank Scherbaum, Nana Mzhavanadze, Simon Schwär, Moritz Berendes, Stefanie Kämmerle, and Selina Sonntag. Your excellent work is the solid foundation of this thesis—thank you very much!

Also, I want to thank my colleagues at the International Audio Laboratories Erlangen and the Fraunhofer-Institut für Integrierte Schaltungen IIS for making the AudioLabs such an ideal place to do research (in alphabetical order): Alexander Adami, Ahmad Aloradi, Carlotta Anemüller, Youssef El Baba, Stefan Bayer, Christian Borß, Martina Böhme, Sebastian Braun, Tom Bäckström, Soumitro Chakrabarty, Pablo Delgado, Sascha Dick, Bernd Edler, Mohamed Elminshawi, Esther Feichtner, Johannes Fischer, Richard Füg, Mandy Garcia, Philipp Götz, Ning Guo, Emanuël Habets, Eva Hasenberger, Tracy Harris, Jürgen

Herre, Adrian Herzog, Thorsten Kastner, Kishor Kayyar, Ankit Kumar, Anna Leschanowsky, María Luis Valero, Wolfgang Mack, Goran Markovic, Thomas Mayenfels, Daniele Mirabilii, Nils Peters, Day-See Riechmann, Thomas Robotham, Sebastian Schlecht, Konstantin Schmidt, Lorenz Schmidt, Martin Strauß, Fabian-Robert Stöter, Armin Taghipour, Maja Taseska, Stefan Turowski, Frank Wefers, Elke Weiland, Nils Werner, and Niklas Winter. An extra shout-out goes to the colleagues at Fraunhofer-Institut für Digitale Medientechnologie IDMT in Ilmenau, Jakob Abeßer, Hanna Lukashevich, Stylianos Mimilakis, Christon Ragavan, and Michael Taenzer for the great hackdays.

Thanks to the German Research Foundation (DFG) for financing my research through the GVM[2]-project (MU 2686/13-1, SCHE 280/20-1) and the AnChor[3]-project (MU 2686/12-1). In this context, I would like to thank the GVM-project collaborators Daniel Vollmer, Nana Mzhavanadze, David Shugliashvili, and Marius Kriegerowski for making this project a success. In March 2018, I had the opportunity to attend a choir recording session of the Carus publishing house, where I could conduct some recordings using throat microphones. Big thanks to Irmgard Bauer, Michael Čulo, Johannes Graulich, Götz Payer, Ester Petri, Wilfried Wenzl, Dietmar Wolf, and the ensemble singers of Sing Sang Song III for this valuable experience. Furthermore, I like to thank Emilia Gómez, Xavier Serra, Helena Cuesta, Pritish Chandna, Juan Gómez-Cañón, Marius Miron, Lorenzo Porcaro, and Furkan Yesiler from the Music Technology Group of Universitat Pompeu Fabra in Barcelona for their hospitality and vital discussions during my research stay in Summer 2019.

Thanks to the participants of the Dagstuhl Seminar on “Computational Methods for Melody and Voice Processing in Music Recordings” for the most scientifically inspiring week of my Ph.D.: Rachel Bittner, Estefanía Cano, Michèle Castellengo, Johanna Devaney, Simon Dixon, Zhiyao Duan, Masataka Goto, Frank Kurth, Cynthia Liem, Antoine Liutkus, Tomoyasu Nakano, Juhan Nam, Ryo Nishikimi, Geoffroy Peeters, Polina Proutskova, Preeti Rao, Justin Salamon, Li Su, Tomoki Toda, Julián Urbano, Ye Wang, and Yi-Hsuan Yang. In this regard, I thank the ISMIR community for being such a welcoming and inclusive group of researchers.

Finally, I am incredibly grateful for being surrounded by people who enrich my life in so many different ways. Thanks to my friends for always cheering me up and keeping me grounded at the same time.

I sincerely thank my family for always being there for me and being home.

Moltes gràcies per tot, Helena. I cannot wait to spend more time with you.

To music and to live concerts. I miss them.

And to vaccines.

[2] Computational Analysis of Traditional Georgian Vocal Music

[3] Automated Methods and Tools for Analyzing and Structuring Choral Music

2 Fundamentals of Signal Processing

Section 2.2 of this chapter is based on [179]. Frank Scherbaum developed the recording setup and conducted the field recordings. Daniel Vollmer prepared the microphones. Sebastian Rosenzweig and Meinard Müller wrote the late-breaking demo abstract. El Mehdi Lemnaouar developed the web-based interface under the supervision of Sebastian Rosenzweig.

In this chapter, we explain the technical background for analyzing and processing audio recordings. First, we formalize the most important audio representations (Section 2.1). Second, we elaborate on multitrack recording techniques for polyphonic vocal music (Section 2.2). Finally, we review approaches for F0-estimation (Section 2.3).

2.1 Audio Representations

In the following, using the notion of Müller [120], we introduce waveform representations (Section 2.1.1), the discrete Fourier transform (Section 2.1.2), the short-time Fourier transform (Section 2.1.3), and log-frequency spectrograms (Section 2.1.4).

2.1.1 Waveform

From a physical perspective, sound is a variation of air pressure over time. Sound waves between roughly 20 Hz and 20 000 Hz are typically perceivable by a young and healthy human ear. Through the use of suitable sound transducers, e.g., microphones, sensors, or pickups, a sound wave can be converted into an electrical signal, or, in other words, a continuously changing level of electrical voltage. Following [120, Section 2.2.1], we model such a continuous-time (CT) signal as a function $f : \mathbb{R} \to \mathbb{R}$ that assigns an amplitude value $f(t) \in \mathbb{R}$ to each point in time $t \in \mathbb{R}$.

Historically, audio signals were recorded using analog storage media such as wax cylinders, phonograph discs, or magnetic tapes. Today, one typically records, stores, and processes audio signals in digital form.

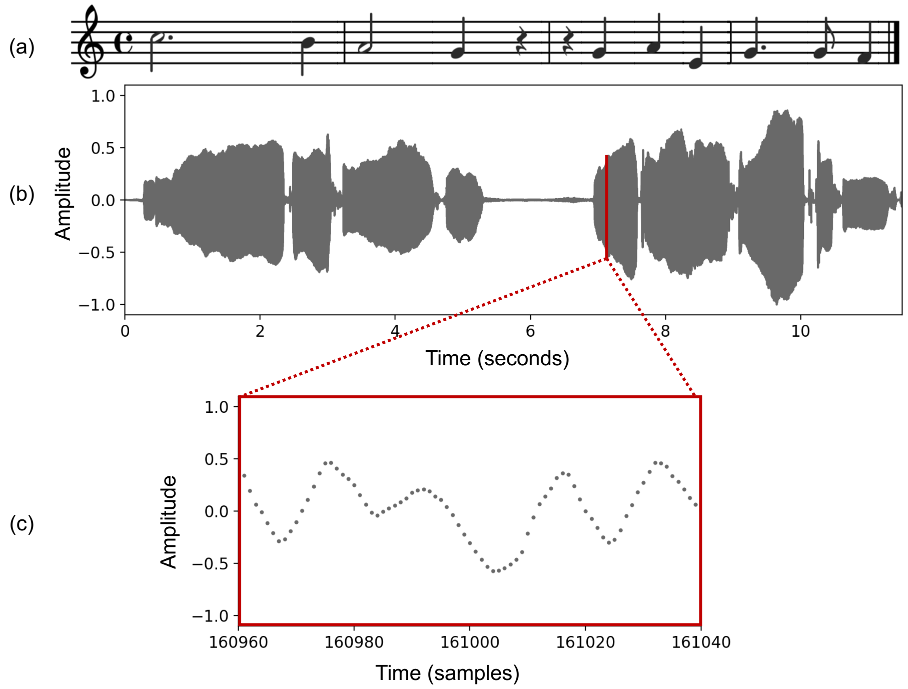


Figure 2.1: Visualization of a singing voice recording. **(a)** Sheet music. **(b)** DT-Signal (waveform). **(c)** Enlarged version of the region marked in red.

To this end, one uses analogue-to-digital (A/D) converters, as included in audio interfaces or sound cards. A/D conversion involves two steps: *sampling* and *quantization*.

One of the most widespread types of sampling is called *equidistant sampling* or *T-sampling*. Given a CT-signal $f : \mathbb{R} \to \mathbb{R}$ and a sampling period $T \in \mathbb{R}_{>0}$, one defines a function $x : \mathbb{Z} \to \mathbb{R}$ by setting

$$x(r) := f(r \cdot T), \tag{2.1}$$

for a sample index $r \in \mathbb{Z}$. The signal x is referred to as discrete-time (DT) signal since it is only defined on a discrete set of points in time. The value $x(r)$ is called a sample taken at time $t = r \cdot T$ of the analog signal f. Note that by the above definition, we assume the DT-signal may be of infinite length with a discrete time axis $\mathbb{Z}$. In practice, to avoid boundary conditions when dealing with signals of finite length, we assume that all samples outside the signal's finite range are zero.

Given the sampling period T, the inverse

$$F_\mathrm{s} = 1/T, \tag{2.2}$$

is referred to as the DT-signal's *sampling rate* $F_\mathrm{s} \in \mathbb{R}_{>0}$ measured in Hz.

In general, sampling leads to a loss of information. The Whittaker–Nyquist–Shannon theorem states that an original analog signal f can be reconstructed perfectly from its DT-version x, if f does not contain any frequencies higher than a frequency commonly referred to as the Nyquist frequency $\Omega \in \mathbb{R}_{>0}$ defined by

$$\Omega := F_{\mathrm{s}}/2\ \mathrm{Hz}. \tag{2.3}$$

Thus, theoretically, sampling rates of more than 40 000 Hz are sufficient for representing all sounds that are audible by the human ear. In practice, one uses higher sampling rates to have an additional safety margin for aliasing artifacts. For instance, CD recordings use a sampling rate of 44 100 Hz, whereas professional productions typically use sampling rates of 48 000 Hz, 96 000 Hz, and beyond. For audio analysis purposes, it is often sufficient to use sampling 22 050 Hz or even 16 000 Hz, since most musically relevant information is contained in the lower frequency regions.

After discretizing the time axis through sampling, the second step in A/D conversion is discretizing the amplitude of the signal. This process is called quantization. Following [120, Section 2.2.2.2], a *uniform quantizer* can be modeled as a function $Q : \mathbb{R} \to \Lambda$ that maps an amplitude value $a \in \mathbb{R}$ to a value from a discrete set of quantized values $\Lambda \subset \mathbb{R}$ by

$$Q(a) := \mathrm{sgn}(a) \cdot \delta \cdot \left\lfloor \frac{|a|}{\delta} + \frac{1}{2} \right\rfloor, \tag{2.4}$$

where $\mathrm{sgn}(\cdot)$ is the signum function that yields the sign of a real number and $\delta \in \mathbb{R}_{>0}$ the quantization stepsize. The quantization stepsize is dependent on the bit depth, which describes the number of bits available to store an amplitude value. For instance, CD recordings typically have a bit depth of 16 bit. Thus, an amplitude value can be quantized to $2^{16} = 65536$ different discrete values. Consequently, the quantization stepsize is $\delta = 1/2^{16}$.

Let us assume we have recorded a singer performing the melody as given by the sheet music in Figure 2.1a. The DT-audio signal of the monophonic recording with $F_{\mathrm{s}} = 22050$ Hz and 11.5 s duration is visualized in Figure 2.1b and will serve as a running example in this chapter. From this visualization, the individual samples are not visible. However, when we zoom into a small region with a duration in the order of a few milliseconds (see Figure 2.1c), we can see a sampled wave. In general, audio waveforms provide a rough idea of when a sound event is occurring (indicated through amplitude changes), but no information about its spectral properties, e.g., the sung notes. In the next section, we will discuss the Fourier transform, which provides more insights into an audio signal's frequency content.

2.1.2 Discrete Fourier Transform

The *Fourier transform* (FT), named after the French mathematician Jean-Baptiste Joseph Fourier, is one of the most fundamental tools in signal processing. The main idea of the FT is to decompose a signal into its constituent frequencies. There exist different definitions and variants of the FT, which are comprehensively

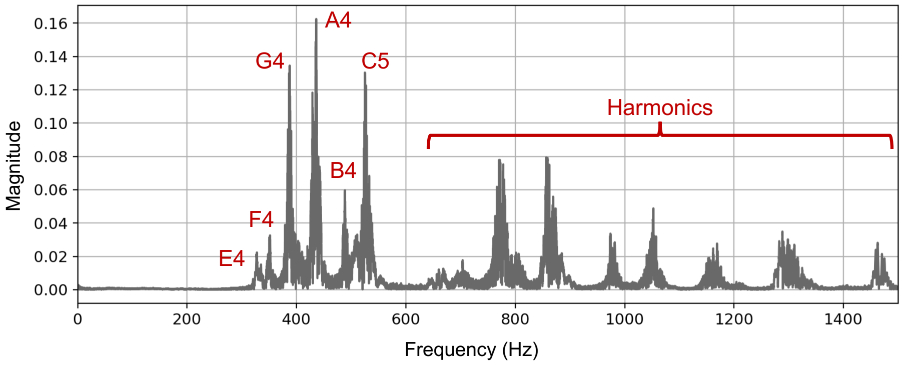


Figure 2.2: Magnitude spectrum of the singing voice recording from Figure 2.1.

discussed in [120, Chapter 2]. Since we are dealing with digital audio signals in this thesis, we focus on formalizing discrete variants of the FT in the following.

To compute the *discrete Fourier transform* (DFT), we consider a finite range $[0 : N - 1] := \{0, \ldots, N - 1\}$ of length $N \in \mathbb{N}$. Then, the DFT is defined as

$$X(k) = \sum_{r=0}^{N-1} x(r) \exp(-2\pi i k r / N), \tag{2.5}$$

for a frequency index $k \in [0 : N - 1]$.

The complex-valued *Fourier coefficient* $X(k)$ describes the magnitude and phase of sinusoidals with physical frequencies $k \cdot F_\mathrm{s}/N$. Since the obtained spectrum is symmetric, one typically considers the frequency range between 0 Hz and the Nyquist frequency Ω. Furthermore, one often uses an efficient algorithm to compute the DFT, known as the *fast Fourier transform* (FFT). For an extensive description of the DFT and the FFT we refer to [120, Chapter 2]. The magnitude spectrum $|X|$ for our singing voice recording is depicted in Figure 2.2. As one can see, the spectrum exhibits several peaks. The marked peaks correspond to the F0s of the sung notes (E4, F4, G4, A4, B4, C5). The peaks from roughly 660 Hz onwards correspond to integer multiples of the F0s, also called harmonics or overtones. In summary, the DFT reveals the frequency content of the recording, whereas information on *when* a frequency occurs (or when a note is sung) is hidden in the phase ot the complex Fourier representation.

2.1.3 Short-Time Fourier Transform

To uncover both the time and frequency information of a digital audio recording, one often uses the discrete *Short-Time Fourier Transform* (STFT) [64]. The basic idea behind the STFT is to divide the audio signal into short frames of length $N \in \mathbb{N}$ using a suitable window function $w : [0 : N - 1] \to \mathbb{R}$ and calculate the DFT for each of the frames. Following [120, Section 2.5], the STFT is defined as

$$\mathcal{X}(m, k) = \sum_{r=0}^{N-1} x(r + mH)w(r) \exp(-2\pi i k r / N), \tag{2.6}$$

where $m \in \mathbb{Z}$ is the *frame index* and $k \in [0 : \lfloor N/2 \rfloor]$ is the *frequency index*. The hopsize $H \in \mathbb{N}$ defines the number of samples between two consecutive frames. The resulting complex time- and frequency-dependent coefficients are associated with the physical time position

$$T_{\mathrm{coef}}(m) = \frac{m \cdot H}{F_{\mathrm{s}}} \tag{2.7}$$

given in seconds and the physical frequency

$$F_{\mathrm{coef}}(k) = \frac{k \cdot F_s}{N} \tag{2.8}$$

given in Hz. The magnitude of the STFT is referred to as *magnitude spectrogram* and defined by

$$\mathcal{Y}(m, k) := |\mathcal{X}(m, k)|. \tag{2.9}$$

In order to visually enhance regions in the spectrogram with low magnitude, we apply logarithmic compression to the magnitudes by setting

$$\Upsilon_{\upsilon}(\mathcal{Y}) := \log(1 + \upsilon \cdot \mathcal{Y}), \tag{2.10}$$

where the compression factor $\upsilon \in \mathbb{R}_{>0}$ determines the degree of compression.

Let us revisit our running example. Figure 2.3a shows a magnitude spectrogram of the singing voice recording with $N = 8192$ (using a Hann window of length 4096 and suitable zero-padding), $H = 256$, and $\upsilon = 0.1$. The higher the magnitude of a time–frequency coefficient, the darker it is shown in the spectrogram representation. From this visualization, the melody of the singing voice is clearly recognizable. For visual support, the F0s that correspond to the sung notes (see the top of the plot) are highlighted in red. Furthermore, the magnitude spectrogram representation shows the harmonic structure of the singing voice.

2.1.4 Log-Frequency Spectrogram

The spectrogram representation introduced in Section 2.1.3 possesses a linearly sampled frequency axis with equidistantly spaced center frequencies of neighboring frequency bands. To account for the logarithmic frequency perception of the human ear, it is desirable to have a logarithmically spaced frequency axis. In the following, we introduce the so-called *log-frequency spectrogram*, which can be

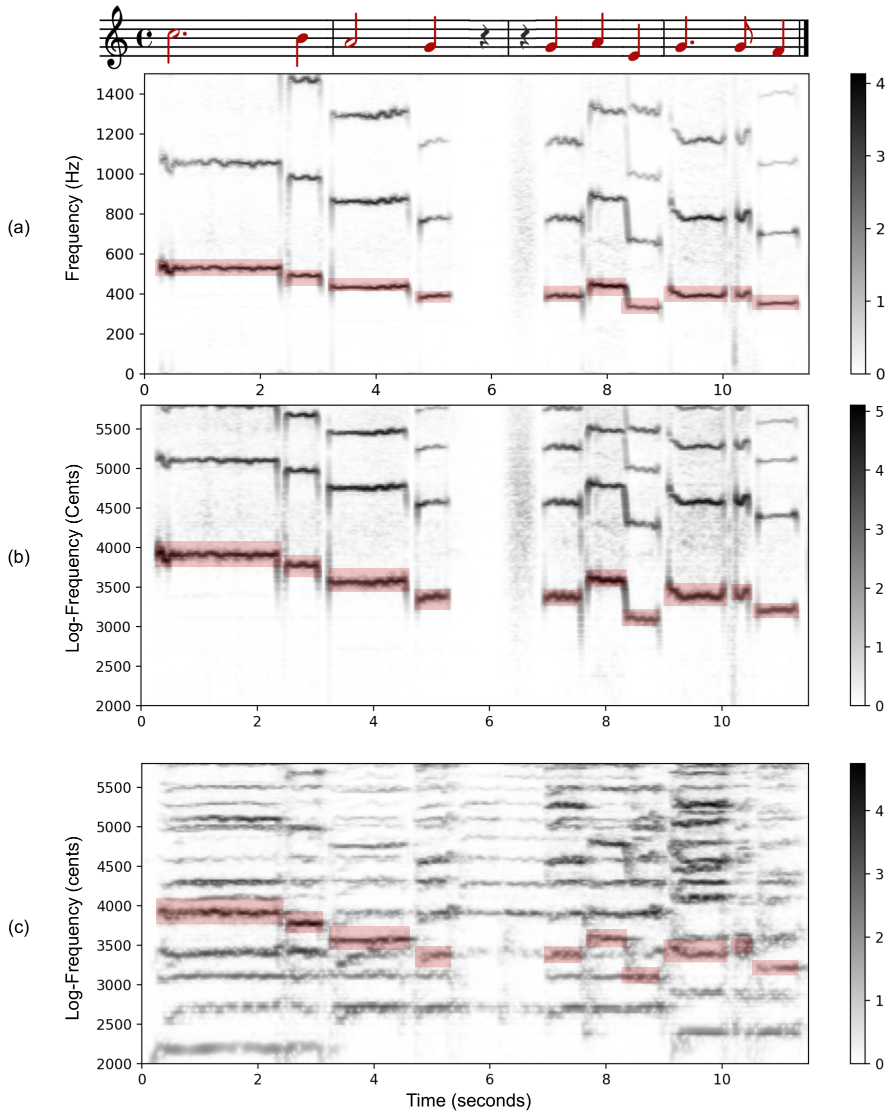


Figure 2.3: Time–frequency representations. **(a)** Magnitude spectrogram of singing voice recording from Figure 2.1. **(b)** Log-frequency spectrogram of singing voice recording from Figure 2.1. **(c)** Log-frequency spectrogram of four-voice performance including the singing voice from (a)/(b).

computed in different ways (see [120, Chapter 3, 8] for further information). In the following, we focus on a basic binning technique.

Let ω_{ref} be a reference frequency (given in Hz). Then, an arbitrary frequency value ω is converted into the logarithmic domain by defining

$$F_{\mathrm{cents}}(\omega) := 1200 \cdot \log_2\left(\frac{\omega}{\omega_{\mathrm{ref}}}\right), \tag{2.11}$$

which measures the distance between ω and ω_{ref} in the unit cents.

Following [120, Chapter 8.2.2.1], given a magnitude spectrogram $\mathcal{Y}$ of an audio signal, the main idea of the binning technique is to map the coefficients $\mathcal{Y}(m, k)$ to a logarithmic frequency axis. To this end, let $R \in \mathbb{R}$ (given in cents) be the desired resolution of the logarithmic frequency axis. Then, a frequency value given in cents is assigned to a bin index $\mathrm{Bin}(\omega)$ with $\mathrm{Bin} : \mathbb{R} \rightarrow \mathbb{Z}$ by setting

$$\mathrm{Bin}(\omega) = \left\lfloor \frac{F_{\mathrm{cents}}(\omega)}{R} + 1.5 \right\rfloor . \tag{2.12}$$

For instance, $R = 100$ results in a logarithmic frequency axis where the bins are spaced one semitone apart. Note that although this definition allows us to bin the coefficients with an arbitrary resolution, the overall frequency resolution of the log-frequency spectrogram is still limited by the frequency grid introduced by the STFT. Let $B \in \mathbb{N}$ be the number of bins to be considered and $b \in [1 : B]$ the bin index. Then, we define the set

$$P(b) = \{k : \mathrm{Bin}(F_{\mathrm{coef}}(k)) = b\}, \tag{2.13}$$

which includes all coefficients that are mapped to a specific bin b. Finally, we define the log-frequency spectrogram by setting

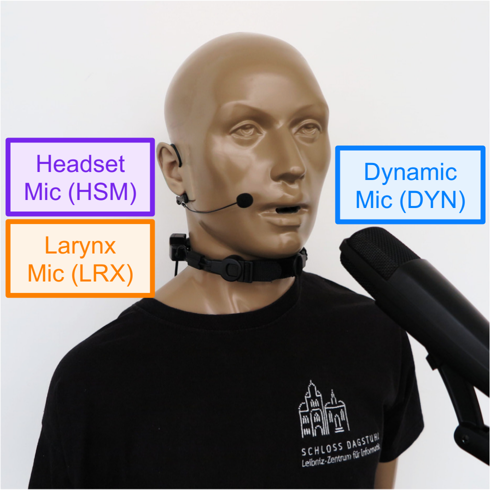
$$\mathcal{Y}_{\mathrm{LF}}(m, b) = \sum_{k \in P(b)} \mathcal{Y}(m, k). \tag{2.14}$$

A log-frequency spectrogram with a resolution of $R = 25$ cents (a quarter semitone) and a reference frequency $\omega_{\mathrm{ref}} = 55$ Hz for our running example is shown in Figure 2.3b. Again we applied logarithmic compression with $\upsilon = 0.1$. As one can see, the logarithmic frequency binning blurs the spectral lines towards the lower frequency range. Furthermore, while the harmonics of the singing voice were spaced linearly in Figure 2.3a, they are spaced logarithmically in the log-frequency spectrogram in Figure 2.3b.

2.2 Multitrack Recording Techniques

When several people sing together, their voices blend into a polyphonic sound mixture. Vocal ensembles are typically recorded using a stereo microphone placed a few meters in front of the ensemble [137]. The goal of such recording setups is to capture the ensemble as a cohesive whole in the best possible acoustic quality. Figure 2.3c shows a logarithmically compressed ($\upsilon = 0.1$) log-frequency spectrogram of a vocal performance with four singers recorded with a microphone placed in front of the singers. The performance includes the singing voice from Figure 2.3a. As one can see, the harmonic structure of the polyphonic sound mixture is much more complex than the one of the monophonic recording. On closer inspection, one can also see that the F0s and the harmonics of the singers' voices partially overlap with each other. Our example shows that computational analysis, e.g., F0-estimation, is generally far more

Figure 2.4: Close-up microphone recording setup for one singer.



challenging for polyphonic recordings (Figure 2.3c) compared to monophonic recordings (Figure 2.3b). Furthermore, some MIR tasks such as source separation (decomposition of a polyphonic recording into its monophonic sources) require multitrack recordings with one or several monophonic tracks per voice as training or evaluation data. Despite the recent advances with decomposing pop music [84, 195], techniques for decomposing recordings of polyphonic singing still face several issues, not least due to the limited availability multitrack vocal recordings [77, 138, 171].

To obtain multitrack recordings of polyphonic vocal music, we make use of close-up microphones attached to the singer's neck and head. The microphones considered in this thesis are depicted in Figure 2.4. The most widespread types are hand-held dynamic (DYN) microphones and headset (HSM or HDS)[4] microphones. Such microphones typically capture signals of high acoustic quality. However, recorded signals may suffer from cross-talk of other singers in proximity (which is likely to happen in vocal ensembles). Thus, DYN and HSM/HDS recordings may not be entirely monophonic.

As a third microphone type, we consider throat or larynx (LRX) microphones, which exploit the peculiarities of human voice production. Singing originates from a complex interplay between the different parts of the vocal apparatus. The lungs and the oscillating vocal folds within the larynx mainly control the pitch and loudness of a sound, whereas resonances and modulations in the vocal tract influence the timbre of a sound. During talking or singing, vibrations of the larynx can be recorded by LRX microphones attached to the skin of the throat. Such microphones typically use electret or piezo pick-ups to sense vibrations through contact with solid objects. Capturing the human voice directly from the throat skin is advantageous since the recorded signals are not interfered by other sounds carried by the air (e.g., the voices of other singers). For this reason, LRX microphones are also used for communication in high-noise environments (e.g., by military and security agencies) and for speech health monitoring. Furthermore, because of their simple usage and robustness, the microphones are ideal for mobile and outdoor use. There

[4] Both abbreviations are used in different parts of this thesis to stay consistent with the file name conventions in the respective datasets.

are some disadvantages of throat microphones as well. Due to the missing contributions of the vocal tract, the recorded signals sound unnatural and muffled. In addition, the signal quality can be affected by tissue characteristics and facial hair on the user's throat. Some singers also complain about unpleasant pressure on their throats during singing. Despite such disadvantages, LRX microphones have shown great potential for analyzing vocal music [86, 173]. In particular, due to the predominant pitch of the recorded voice, the task of F0-estimation is much easier for LRX signals, as we will see in the next section.

2.3 Fundamental Frequency Estimation

In the following, we formalize the notion of an F0-trajectory (Section 2.3.1) and elaborate on the F0-estimation algorithms YIN (Section 2.3.2), pYIN (Section 2.3.3), Melodia (Section 2.3.4), and CREPE (Section 2.3.5), which are used in the experiments of this thesis.

2.3.1 Notion of F0-Trajectory

Throughout this thesis, we use a consistent notion of an F0-trajectory. We model an F0-trajectory as a function

$$\eta : \mathbb{Z} \to \mathbb{R} \cup \{*\} \tag{2.15}$$

that assigns to a given time index $n \in \mathbb{Z}$ either a real-valued frequency value $\eta(n) \in \mathbb{R}$ (given in cents) or the symbol $\eta(n) = *$ (when the frequency value is left to be unspecified). For brevity, we use the notion

$$\eta(a : b) := \{\eta(a), \eta(a+1), ..., \eta(b)\} \tag{2.16}$$

for integers $a, b \in \mathbb{N}$.

2.3.2 YIN

One of the most well-known algorithms for F0-estimation is YIN[5], which was first introduced by Cheveigné and Kawahara [48]. YIN is a time domain algorithm, which produces one F0-estimate for each time frame following three main steps. In the first step, one computes a function referred to as *cumulative mean normalized difference function* (CMNDF). The CMNDF is depicted for one frame of our running example in Figure 2.5a. As one can see, the CMNDF has local minima at integer multiples of the period of the signal. In the second step, one sets an absolute threshold and determines the smallest value of τ for which CMNDF has a local minimum deeper than that threshold. For our example frame, given a threshold as indicated by the red dotted line in Figure 2.5a, we obtain $\tau = 42$ samples, corresponding to a frequency of $F_s/\tau = 22050$ Hz$/42 = 525$ Hz. In the third step, the period estimate

[5] The name YIN stems from the Chinese philosophical concept "Yin" and "Yang".

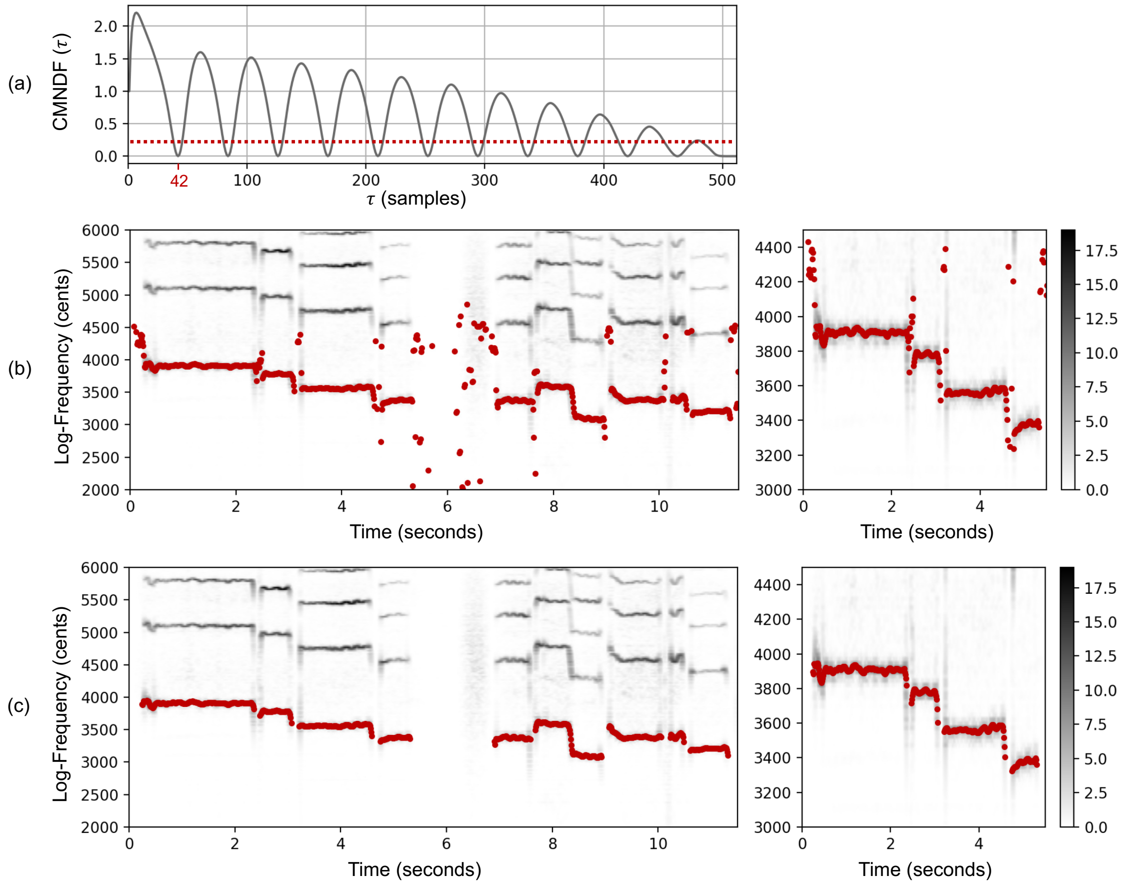


Figure 2.5: Illustration of YIN and pYIN F0-estimation for running example. **(a)** Cumulative mean normalized difference function for one frame. **(b)** Log-frequency spectrogram superimposed with F0-trajectory estimated by YIN (red line) and zoom into a section (right). **(c)** Log-frequency spectrogram superimposed with F0-trajectory estimated by pYIN.

is refined using parabolic interpolation. Figure 2.5b depicts the F0-trajectory estimated by YIN for our running example superimposed with a log-frequency spectrogram. Since the algorithm does not enforce continuity of the estimated F0-trajectories, one often obtains highly fluctuating F0-estimates (e.g., see Figure 2.5b at around 6 seconds). In particular, YIN suffers from confusions of the F0 with higher harmonics (especially the octave). As one of its main benefits, YIN is an algorithm of low computational complexity. Implemented efficiently, the YIN algorithm can be used to estimate F0-trajectories in real-time applications. YIN implementations are, for instance, available in librosa [112] (Python), the aubio library[6] (C), Essentia [20, 38] (C, JavaScript), and as Vamp-Plugin[7]. For further information on the YIN algorithm, we refer to [48].

[6] https://aubio.org/

[7] https://vamp-plugins.org/

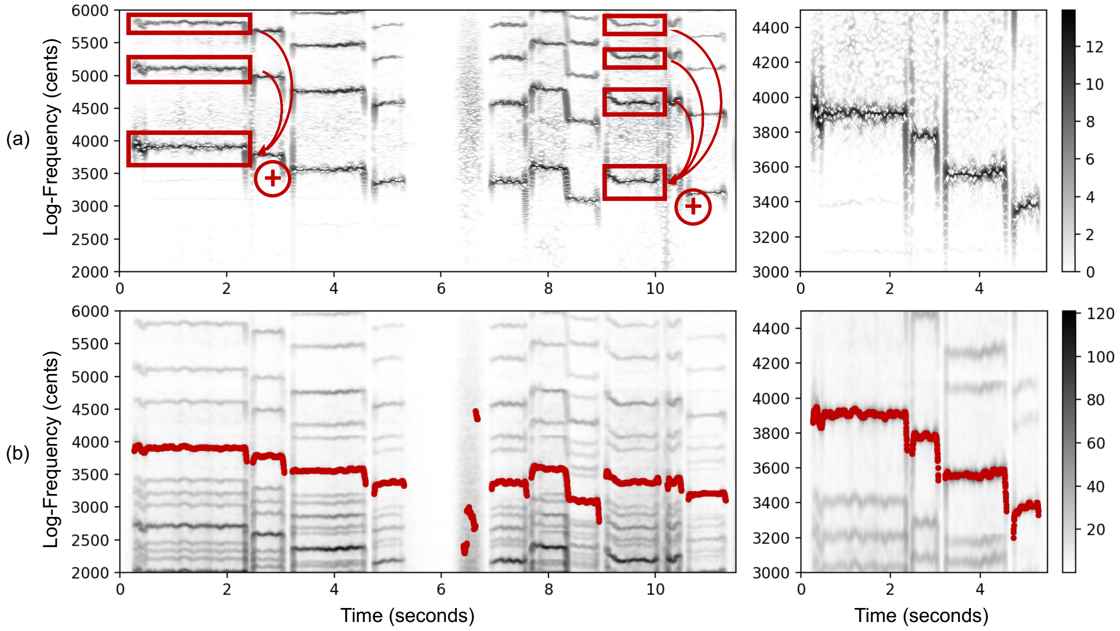


Figure 2.6: Illustration of Melodia F0-estimation for running example. **(a)** Harmonic summation on log-frequency spectrogram with a refined frequency resolution and zoom into a section (right). **(b)** Salience representation superimposed with F0-trajectory estimated by Melodia (red line).

2.3.3 pYIN

Probabilistic YIN, or pYIN, introduced by Mauch and Dixon [109], is a modification of the previously described YIN algorithm. To increase the robustness and alleviate the continuity problems of the YIN algorithm, the authors propose two main strategies. First, one applies YIN multiple times with different thresholds taken from a given threshold distribution. In this way, one obtains multiple F0-candidates per frame. Second, the authors introduce an additional temporal smoothing step. Using a hidden Markov model (HMM) and Viterbi decoding, the algorithm determines a smooth trajectory of F0-values from the F0-candidates. Furthermore, the HMM smoothing includes frame-wise decision whether a frame is voiced or unvoiced (commonly referred to as *voicing detection*). Figure 2.5c shows the estimated pYIN-trajectory for our running example. As one can see, the outliers of the YIN-trajectory have been removed and the estimated F0-trajectory is smooth. As a downside of pYIN, the algorithm is computationally more complex than YIN. Furthermore, because of the additional HMM smoothing, pYIN is not real-time capable. pYIN implementations are, for instance, available in librosa [112] (Python), Essentia [20, 38] (C, JavaScript), and as Vamp-Plugin. For further information on the pYIN algorithm, we refer to [109].

2.3.4 Melodia

Melodia is a frequency domain algorithm introduced by Salamon and Gómez [169] that is primarily designed for the task of melody extraction. Given a polyphonic recording, the task of melody extraction involves automatically extracting the F0-trajectory of the main (predominant) melodic line. In the close-up microphone signals considered in this thesis, the vocal F0 is predominant during singing. Melodia relies on an enhanced time–frequency representation, also called *salience representation*, of the audio signal, which can be computed in four main steps. First, an STFT is computed. Second, by making use of the phase of the complex Fourier-coefficients, the frequency resolution is refined using a technique referred to as *instantaneous frequency* (IF) estimation (see [169] and [120, Section 8.2.1] for details). Third, the IF-estimates are binned onto a logarithmic frequency axis using a conceptually similar technique as explained in Section 2.1.4. Fourth, one applies a technique called *harmonic summation*, which exploits the harmonicity of sounds by accumulating the harmonics of a tone over frequency. The refined log-frequency spectrogram for our running example and the working principle of harmonic summation are visualized in Figure 2.6a. The resulting salience representation is visualized in Figure 2.6b. As one can see, harmonic summation leads to replications of spectral patterns ("ghost components") appearing particularly in the lower frequency regions. Subsequently, the F0-trajectory is computed using a peak streaming approach based on heuristics inspired by auditory streaming cues. Furthermore, Melodia includes a voicing detection step. Figure 2.6b shows the F0-trajectory estimated by Melodia for our running example. The robustness of Melodia to the presence of other, non-predominant sound sources in the analyzed signal comes at the cost of an increased computational complexity of the algorithm. Melodia implementations are, for instance, available in Essentia [20, 38] (C, JavaScript) and as Vamp-Plugin. For further information on the Melodia algorithm, we refer to [169].

2.3.5 CREPE

CREPE (Convolutional Representation for Pitch Estimation) is a deep learning-based algorithm introduced by Kim, Salamon, and Bello [97]. CREPE takes a waveform as input and outputs frame-wise F0-estimates. The network architecture has over 22 million parameters and consists of six convolutional blocks and a final fully connected layer. Each convolutional block consists of a 1D-convolution layer, a batch normalization layer [89], a max-pool layer, and a dropout layer [194]. Figure 2.7a shows the concatenated magnitude Fourier spectra of the 1024 learned filters in the first convolutional layer. The spectra are sorted according to the index of their maximal value. As one can see, the learned filters exhibit a bandpass-like characteristic. Furthermore, the filters' center frequencies are non-linearly distributed across the frequency range, which resembles the non-linear resolution of the log-frequency spectrogram (see Section 2.1.4). The final fully connected layer has 360 nodes that are associated to frequency values on a logarithmic frequency axis with a 20 cents quantization in the range between the pitches C1 and B7. Each node outputs an activation value in the range [0, 1], which indicates the likelihood of the input signal to have an F0 that falls within the associated frequency bin. The output F0-estimates are obtained through averaging of associated frequencies weighted with their activations. Optionally, the obtained F0-trajectory can be smoothed using the Viterbi algorithm. Figure 2.7b shows the smoothed F0-trajectory estimated by CREPE for our running example. Note that CREPE outputs one F0-value per time frame. However, the voicing can be inferred

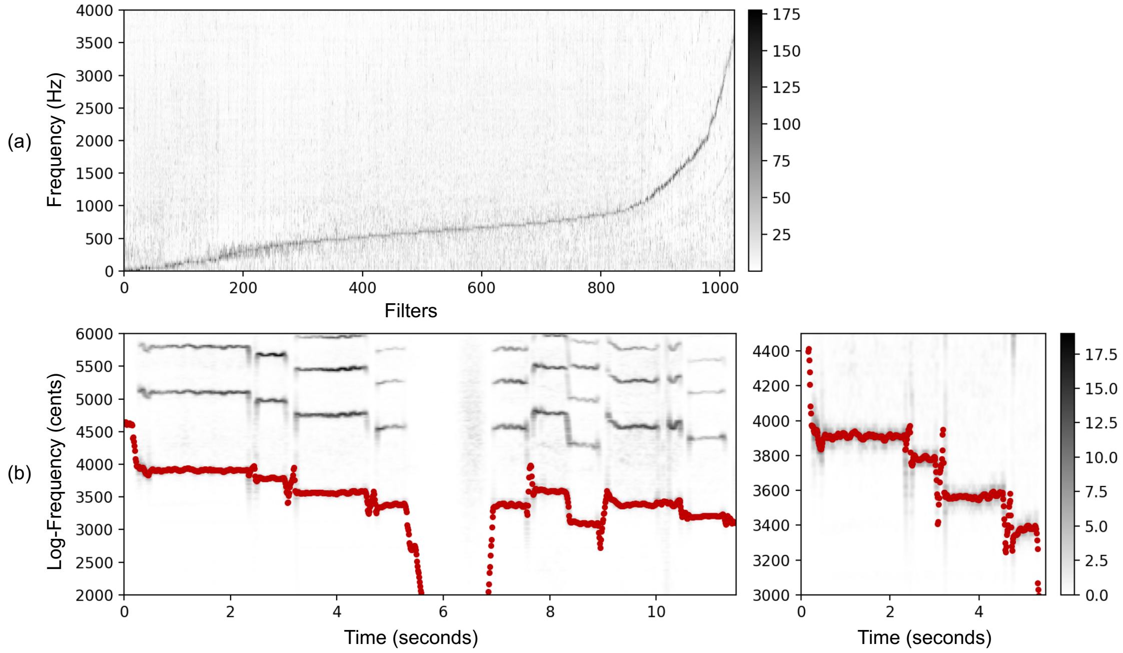


Figure 2.7: Illustration of CREPE F0-estimation for running example. **(a)** Concatenated and sorted magnitude Fourier spectra of the learned filters in the first convolutional layer. **(b)** Log-frequency spectrogram superimposed with F0-trajectory estimated by CREPE (red line) and zoom into a section (right).

from the output layer activations. A stripped-down version of CREPE with roughly 500 000 parameters can run in real-time in the browser.[8] The original CREPE implementation is available on GitHub[9] and PyPi[10]. For further information on CREPE, we refer to [97]. A detailed analysis of the CREPE network can be found in [5].

[8] https://marl.github.io/crepe/

[9] https://github.com/marl/crepe

[10] https://pypi.org/project/crepe/

Part I

Analysis of Choral Music

3 Dagstuhl ChoirSet: Creation of a Corpus for Analyzing Choral Singing

This chapter is based on [159]. Sebastian Rosenzweig and Helena Cuesta substantially contributed to the recording process, dataset creation and writing of the article. Christof Weiß took over the musical coordination as well as the conducting during rehearsals and recordings. Frank Scherbaum contributed to the technical coordination of the recording session and provided his microphones. Emilia Gómez and Meinard Müller supervised this work and contributed to dataset creation and writing of the article.

MIR research on choral singing benefits from multitrack recordings of the individual singing voices. However, there exist only few publicly available multitrack datasets on polyphonic singing. In this chapter, we present Dagstuhl ChoirSet (DCS), a multitrack dataset of a cappella choral music designed to support MIR research on choral singing. The dataset includes recordings of an amateur vocal ensemble performing two choir pieces in full choir and quartet settings. The audio data was recorded during an MIR seminar at Schloss Dagstuhl using different close-up microphones to capture the individual singers' voices. In this chapter, we give detailed insights into all stages of creating DCS: recording process, data preparation, generation of annotations as well as development of suitable interfaces for publicly accessing and reusing the data. Furthermore, we demonstrate the potential of the dataset for MIR research by discussing case studies on choral intonation assessment and multiple-F0 estimation.

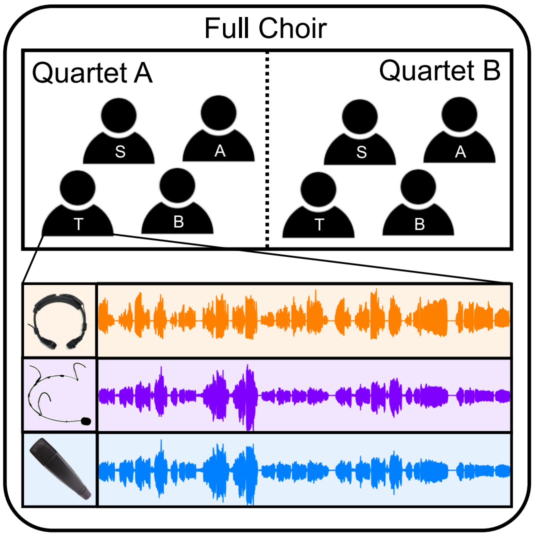
3.1 Introduction

Choral singing is one of the most widespread types of polyphonic singing [197]. For instance, the European Choral Association[11] reports over 37 million amateur and professional choir singers on the European continent, while Chorus America[12] reports 54 million active singers in the U.S. The great interest in choral singing motivates the need for MIR technologies to support singers and conductors in

[11] https://europeanchoralassociation.org

[12] https://www.chorusamerica.org

Figure 3.1: Dagstuhl ChoirSet—an overview.



their rehearsal practices[13] via mobile applications[14,15] and web-based interfaces[16]. Over the last years, there has been an increasing number of MIR techniques developed for analyzing polyphonic vocal music [40, 46, 50, 51, 86, 87, 110, 214] as well as for synthesizing expressive singing [16, 32]. Essential to the development of such techniques is the availability of suitable datasets and processing tools. In particular, multitrack recordings are of great value for evaluation purposes. However, due to high demands on recording equipment and infrastructure, there exist only few publicly available multitrack datasets on polyphonic vocal music.

The lack of suitable research data was one of the driving motivations to create *Dagstuhl ChoirSet* (DCS), a publicly available multitrack dataset of a cappella choral music for MIR research. The audio data was recorded during a one-week research seminar on "Computational Methods for Melody and Voice Processing in Music Recordings" [128] at Schloss Dagstuhl[17]. For the recordings, we assembled a vocal ensemble of mostly amateur singers (all were participants of the Dagstuhl seminar) covering different SATB (Soprano, Alto, Tenor, and Bass) voice sections. After several rehearsals with a conductor, we recorded multiple takes of two choir pieces in a full choir setting and two quartet settings (Quartet A and Quartet B). Furthermore, we recorded some systematic exercises for practicing choral intonation. As one main feature of the dataset, individual singers were recorded using multiple close-up microphones, including larynx, headset, and dynamic microphones (see Figure 3.1). Subsequent to recording and curating the recorded multitrack data, we annotated beat positions and generated time-aligned score representations for each of the music recordings. Furthermore, we automatically extracted F0-trajectories for all close-up microphone signals. The publicly available dataset is archived on Zenodo[18] and is

[13] https://cordis.europa.eu/project/id/770376
[14] https://www.carus-verlag.com/en/digital-media/carus-music-the-choir-app
[15] https://www.singerhood.com
[16] https://trompamusic.eu/choir-singers
[17] https://www.dagstuhl.de/19052
[18] https://doi.org/10.5281/zenodo.3897182

accessible via an interactive web-based interface with score-following and playback functionality[19]. In order to facilitate reproducibility and further research using this dataset, we have created an open source Python toolbox with helper functions to load, parse, and process dataset files[20].

In summary, our annotated dataset has different musical and acoustical dimensions that open up a variety of research scenarios. Besides being a good basis for studying amateur choral singing, DCS constitutes a challenging scenario for various fundamental tasks in MIR such as automatic music transcription [7], score-to-audio alignment [199], and beat tracking [18, 223]. Moreover, the close-up microphone signals as well as the available F0-trajectories and scores can serve as a baseline to research on (informed) source separation techniques [30, 31]. Furthermore, it allows for comparisons between multiple choir/quartet performances, choir settings, and microphone types.

The remainder of this chapter is structured as follows. In Section 3.2, we give an overview on datasets related to our work. In Section 3.3, we describe DCS by providing details on the choir settings, selected pieces, technical setup of the recordings, and generated annotations. In Section 3.4, we explain the different interfaces to access and use the dataset. In Section 3.5, we demonstrate the relevance of this dataset for MIR research by conducting two case studies on choral intonation assessment and multiple-F0 estimation using state-of-the-art algorithms. Finally, in Section 3.6, we summarize our contributions and provide further notes.

3.2 Prior Work

There is an urgent need for datasets in the field of MIR: annotated data are crucial for training data-driven systems or evaluating methods developed to solve specific tasks. Over the last years, the availability of suitable datasets has triggered research on tasks such as melody extraction (e.g., MedleyDB [11]), music style identification (e.g., Ballroom dataset [76]), and automatic chord recognition (e.g., Beatles dataset [82]).

The datasets closely related to DCS are presented in Table 3.1. Su et al. [196] created a small dataset for research on choral music. It consists of five short excerpts of Western choral music, ranging from 18 to 40 seconds in length. The dataset contains stereo audio recordings and note event annotations, annotated by a professional pianist. Although small in size, this dataset is relevant for multiple-F0 estimation in complex scenarios where sources are similar, (e.g., voices of a choir), and where several sources produce the same notes (i.e., unisons).

Over the last years, there has been an increasing interest of the MIR community in analyzing world music [134, 188], including traditional singing [208, 211]. A conceptually similar dataset to DCS in terms of recording methodology and utilized microphones is a set of multitrack field recordings of three-voice

[19] https://www.audiolabs-erlangen.de/resources/MIR/2020-DagstuhlChoirSet

[20] https://github.com/helenacuesta/ChoirSet-Toolbox

Name/Author	Multitrack	Annotations	Publicly Available	# Recordings	Duration (hh:mm:ss)
[196]	No	MIDI	On Request	5 excerpts	00:02:11
Barbershop Quartets[21]	Yes	MIDI	No	22 songs	00:42:10
Bach Chorales[22]	Yes	MIDI	No	26 songs	00:58:20
[180] (see Section 7.2)	Yes	-	On Request	216 songs	06:08:51
Erkomaishvili Dataset [161] (see Chapter 5)	No	Structure, F0, Score, Onsets	Yes	101 songs	07:05:00
Choral Singing Dataset (CSD) [40]	Yes	MIDI, F0, Notes	Yes	3 songs	00:07:14
Dagstuhl ChoirSet (DCS)	Yes	MIDI, F0, Beats	Yes	2 songs, exercises	00:55:30

Table 3.1: Comparison of polyphonic singing datasets described in Section 3.2. The reported durations refer to the total recording duration (not counting multiple tracks per recording if available).

Georgian vocal music [180] (see Section 7.2 for a description). Furthermore, the Erkomaishvili Dataset is a publicly available corpus based on historic tape recordings of three-voice traditional Georgian songs performed by the former master chanter Artem Erkomaishvili [161] (see Chapter 5 for a description).

In the context of Western polyphonic vocal music, we find very few multitrack datasets. Two examples are datasets from a commercial application that have been used by [115, 183, 184]: the Barbershop Quartets[21] and the Bach Chorales[22]. Both datasets contain separate tracks for each of the four SATB singers and an additional track with a stereo mix. The Barbershop recordings comprise 22 songs with a total length of 42 minutes, whereas the Bach Chorales contain 26 recordings with a total length of 58 minutes. The audio recordings and the accompanying synchronized MIDI files are not freely available.

The Choral Singing Dataset (CSD) [40] is a publicly available dataset of Western polyphonic vocal music[23]. The CSD consists of multitrack recordings of three SATB choral pieces: *Locus Iste* by Anton Bruckner, *Niño Dios d'Amor Herido* by Francisco Guerrero, and *El Rossinyol*, a popular Catalan song, performed by a small choir of 16 singers. The four singers of each choir section were recorded simultaneously in the same room with individual handheld dynamic microphones. However, the different sections were recorded separately where a MIDI track served as reference. The recording length of the three songs is around seven minutes. Furthermore, the CSD includes synchronized MIDI files, note annotations per choir section, and F0-annotations. In summary, the CSD is most similar to our dataset in terms of musical aspects. Further similarities and differences of the CSD to our dataset are discussed in Section 3.3.3.

3.3 Dagstuhl ChoirSet

In this section, we describe all components of DCS. In Section 3.3.1, we give details on the choir settings as well as the recorded pieces and exercises. Then, we explain the recording setup of the multitrack

[21] https://www.pgmusic.com/barbershopquartet.htm
[22] https://www.pgmusic.com/bachchorales.htm
[23] https://zenodo.org/record/2649950

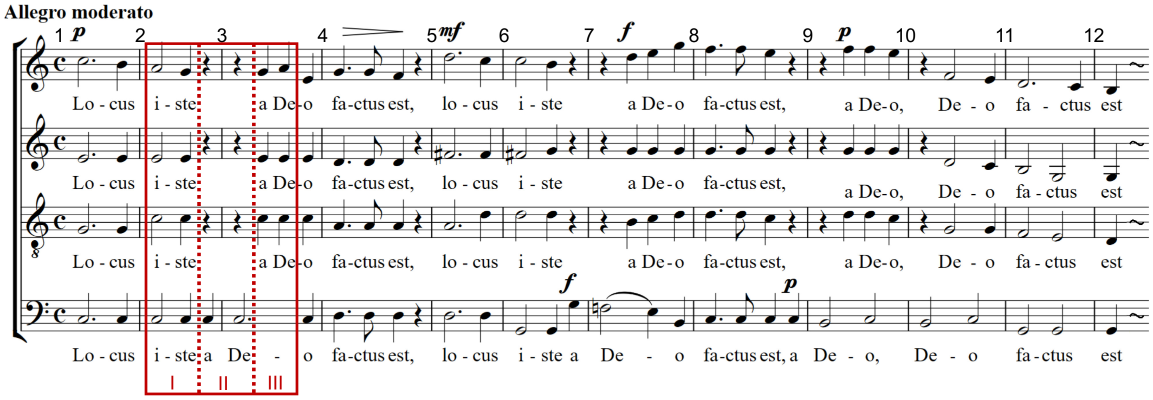

Allegro moderato
1 2 3 4 5 6 7 8 9 10 11 12
p mf f p f p
Lo - cus i - ste a De - o fa - ctus est, lo - cus i - ste a De - o fa - ctus est, a De - o, De - o fa - ctus est
Lo - cus i - ste a De - o fa - ctus est, lo - cus i - ste a De - o fa - ctus est, a De - o, De - o fa - ctus est
Lo - cus i - ste a De - o fa - ctus est, lo - cus i - ste a De - o fa - ctus est, a De - o, De - o fa - ctus est
Lo - cus i - ste a De - o fa - ctus est, lo - cus i - ste a De - o fa - ctus est, a De - o, De - o fa - ctus est
I II III

Figure 3.2: Anton Bruckner, *Locus Iste* WAB 23 (measures 1 to 11). The score was obtained from CPDL and edited by Brian Marble[24].

recordings in Section 3.3.2 and discuss the different dimensions of DCS in Section 3.3.3. Subsequently, we elaborate on the manually created beat annotations in Section 3.3.4. Furthermore, we provide details on the time-aligned score representations in Section 3.3.5. Finally, we describe the automatically extracted F0-trajectories in Section 3.3.6.

3.3.1 Choir Settings and Musical Content

In total, 13 singers (Dagstuhl seminar participants) took part in the recording session. All singers have provided their consent to publish the recorded material for research purposes under a Creative Commons license. The Full Choir consisted of two sopranos, two altos, four tenors, and five basses. From the Full Choir, we selected two soloistic SATB quartets (Quartet A and Quartet B) with four different singers each. The singers had diverse musical backgrounds (from hobby musicians to such holding a music degree) as well as varying levels of experience in (choir) singing within different musical genres. These experiences ranged from singers who had never sung in a choir before to a professional singer with many years of training. Considering that the singers had not sung in this constellation before the Dagstuhl seminar and had only few rehearsals together (3 sessions of roughly 1 hour length), the recorded choir and quartets may be representative of an amateur choir level, with individual skills partly exceeding that level. Rehearsals and recorded performances were also conducted by a Dagstuhl seminar participant, who is a professional composer with solid experience in conducting semi-professional choirs, orchestras, and big bands. We recorded two pieces as well as several intonation exercises with the full choir and the two quartets. The central piece of DCS is Anton Bruckner's *Locus Iste* (WAB 23) in Latin language. Figure 3.2 displays the first eleven measures of the piece's score obtained from the Choral Public Domain Library (CPDL)[24]. This small choir piece of approximately three minutes' duration is musically interesting, containing several melodic and harmonic challenges such as chromatic parts and covering a large part of each voice's tessitura (S: B3-G5, A: G3-B4, T: C3-E4, B: F2-C4). Beyond that, the piece is part of the CSD [40]

[24] `https://www.cpdl.org/wiki/images/9/94/Locus_Iste_rev.pdf`

Table 3.2: Overview of the audio recordings in DCS. The third column indicates the number of takes available for each piece and the last column refers to the total duration of all takes together.

Piece	Setting	# Takes	Duration (mm:ss)
Locus Iste	Full Choir	3	07:22
	Quartet A	7	16:26
	Quartet B	6	14:02
Tebe Poem	Full Choir	5	05:27
	Quartet A	2	02:30
Exercises	Full Choir	33	06:00
	Quartet A	25	03:43
Total		**81**	**55:30**

(see Section 3.2), thus allowing for interesting comparative studies across datasets. Furthermore, we selected the piece *Tebe Poem* by the Bulgarian composer Dobri Hristov[25]. Both pieces are written for SATB choirs in four parts. In addition to these two pieces, the dataset contains a set of vocal exercises of different difficulties and forms taken from the book *Choral Intonation* [1]. The exercises include scales, long and stable notes, chords, cadences, and a variety of intonation exercises. The additional recordings are potentially interesting to study aspects of ensemble singing such as interval intonation, F0-agreement in unison singing, and intonation drift in a cappella performances.

3.3.2 Multitrack Recordings

During the recording session, which took place in a Dagstuhl seminar room, we recorded multiple takes of the different pieces and settings. An overview of the recorded material in DCS is presented in Table 3.2. The reported durations refer to the accumulated durations of all takes for a specific piece and setting (not counting multiple tracks per take). The different choir settings were recorded using multiple microphones. In order to record the overall performance, we used an ORTF stereo microphone (Schoeps MSTC 64 U) spaced roughly 3 m away from the singers. The recorded stereo microphone signal is referred to as STM signal in the following. Furthermore, we used dynamic (Sennheiser MD421 II), headset (DPA 4066F), and throat (Albrecht AE 38 S2a) microphones to record individual singers as illustrated in Figure 2.4. In the following, we abbreviate the three microphone types as DYN, HSM, and LRX, respectively.

To illustrate the microphone differences, magnitude spectrograms of LRX and DYN microphone signals for a tenor singer in a quartet setting are shown in Figure 3.3a. The shown excerpts correspond to the marked *Locus Iste* passage in Figure 3.2. It can be observed that the LRX signal is cleaner than the DYN signal. This becomes evident especially in Part II (middle part of the marked passage), where the solo bass voice leaks more strongly into the DYN signal than into the LRX signal of the tenor.

For our recordings, we had four DYN, three HSM and eight LRX microphones available. The complete setup as shown in Figure 2.4 could only be used for three singers—other singers were equipped with two,

[25] `http://www3.cpdl.org/wiki/index.php/Tebe_Poem_(Dobri_Hristov)`

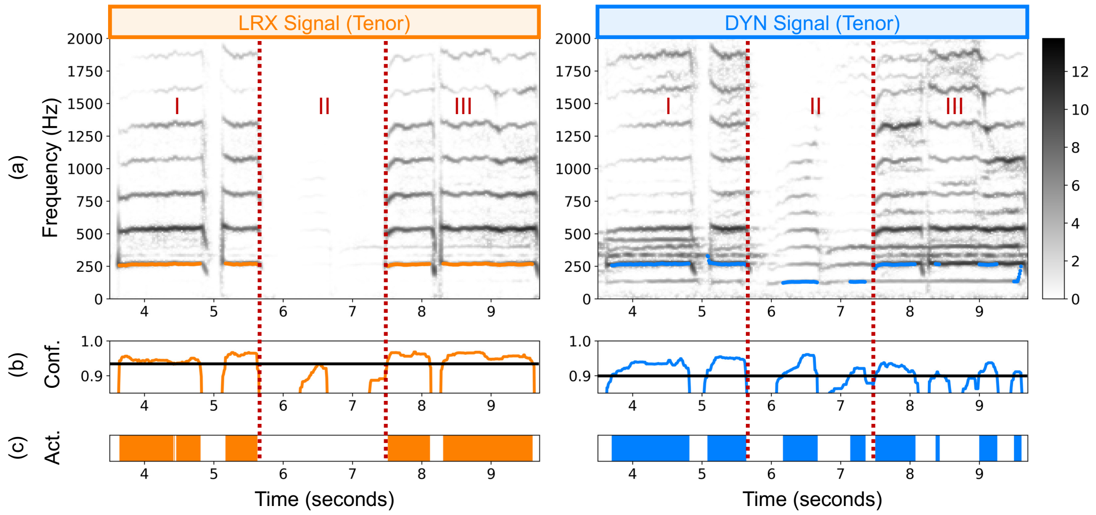


Figure 3.3: Comparison of LRX and DYN signals from a tenor singer. Excerpts correspond to the marked *Locus Iste* passage in Figure 3.2. **(a)** Magnitude spectrograms. CREPE F0-trajectories are plotted on top in the respective colors. **(b)** Smoothed CREPE confidence. **(c)** Binarized trajectory activations obtained by thresholding smoothed confidence (LRX threshold: 0.935, DYN threshold: 0.9).

one, or no individual microphone(s). Note that we distributed the microphones such that at least one singer of each part was captured with one LRX and one DYN microphone. The microphone signals were recorded using one RME Fireface UFX audio interface, two 8-channel RME Micstasy A/D converters, and the Digital Audio Workstation (DAW) Logic Pro X running on an Apple MacBook Pro (see Figure 3.4). Furthermore, we created an additional reverb version of the stereo microphone signal using the *ChromaVerb* plug-in in Logic Pro X with a decay time of 2 seconds. After recording, all tracks were exported from the DAW and subsequently cut according to manually set cut points using the tool PySox [12]. PySox is an open source library that provides a Python interface to SoX (Sound exchange)[26], a command line tool for sound processing. The cut tracks are available in DCS as monophonic WAV files with a sampling rate of 22 050 Hz.

3.3.3 Dataset Dimensions

DCS offers different musical and acoustical dimensions, which are summarized in Table 3.3. We refer to the dimensions as Song, Setting, Take, Voice, and Microphone. The Song dimension consists of the two choral pieces *Locus Iste* and *Tebe Poem* as well as the systematic exercises. The Setting dimension includes the three choir settings: Full Choir, Quartet A, and Quartet B. The Take dimension indicates the number of takes. The Voice dimension is defined by the singers present in the signal—either one of the SATB sections or the mixture of all sections recorded by the STM microphone. Finally, the Microphone dimension refers to the microphone types used to record the singers.

[26] `http://sox.sourceforge.net/`

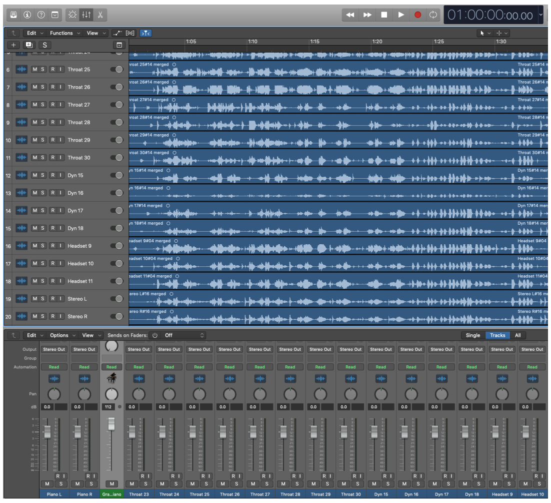


Figure 3.4: Screenshot (detail) of digital audio workstation (Logic Pro X) with multiple tracks.

Table 3.3: DCS dimensions.

Dimension	Shortcut	Meaning
Song	`LI`	*Locus Iste*
	`TP`	*Tebe Poem*
	`SE`	Systematic Exercises
Setting	`FullChoir`	Full Choir Setting
	`QuartetA`	Quartet A Setting
	`QuartetB`	Quartet B Setting
Take	`Take`	Take Number
Voice	`S`	Soprano
	`A`	Alto
	`T`	Tenor
	`B`	Bass
	`Stereo`	Stereo Mic
	`StereoReverb`	Stereo Mic Reverb
Microphone	`LRX`	Larynx Mic
	`DYN`	Dynamic Mic
	`HSM`	Headset Mic
	`STR`	Stereo Mic R
	`STL`	Stereo Mic L
	`STM`	Stereo Mic L+R

The multiple dimensions of DCS make it unique when compared to related datasets such as the CSD [40]. The main differences between the CSD and DCS lie in the Setting, Take, and Microphone dimensions. The CSD includes one singer setting, a single take per song and one microphone type. Furthermore, the CSD choir sections were recorded separately, while all singers were captured at the same time in DCS. The different recording setup in DCS enables studies on interactions between sections. However, as opposed to the Full Choir setting in DCS, the recorded choir in the CSD is larger and balanced in the number of singers per section. Therefore, CSD allows for more detailed studies on singer interaction within choir sections.

In order to account for the variety of different dimensions, we developed a filename convention for all audio and annotation files included in DCS. The general format of the filenames is the following (cf. Table 3.3): `DCS_{Song}_{Setting}_Take{#}_{Voice}{#}_{Microphone}.{Suffix}`. For example, `DCS_LI_FullChoir_Take02_T2_LRX.wav` refers to the audio signal from the larynx microphone (LRX) of the second tenor (T2) in the Full Choir setting (FullChoir) during the second take (Take02) of *Locus Iste* (LI). Note that the files with microphone shortcut STM contain a mono mix of the left and right channel of the stereo microphone.

3.3.4 Manual Beat Annotations

The beat is a key unit of the temporal structure of music [75]. As stated by [155], when beat annotations are manually generated by tapping along to an audio signal, they reflect the ability of the annotator to *produce* the beats rather than their *perception*. In such cases, the *produced* beat annotations can be subsequently refined by iteratively listening and modifying them according to perceptual cues. Following this premise, we generated beat annotations for all STM signals of *Locus Iste* and *Tebe Poem* in a two-stage process: in the first stage, annotations were manually created by an annotator with some musical background. The *annotation by tapping* feature in Sonic Visualiser [29] was used for this task. Sonic Visualiser is an open source software for generating manual annotations of various kinds. In the second stage, annotations were reviewed and refined by a second, experienced annotator using the same software.

These beat annotations are provided as comma-separated value (CSV) files with two columns. The first column contains timestamps in seconds, whereas the second column contains beat and measure information provided as floating point numbers to three decimal places. The part in front of the decimal point encodes the measure number. The part after the decimal point indicates the beat position inside the measure. For example, in 4/4 time, each beat is represented as an increment of 1/4 = 0.250, and therefore the beat positions are given as 1.000, 1.250, 1.500, 1.750, 2.000, 2.250, 2.500, and so on.

3.3.5 Time-Aligned Score Representations

In order to obtain a musical reference for the different performances of *Locus Iste* and *Tebe Poem*, we aligned MIDI representations of the pieces to the STM signals using the beat annotations from Section 3.3.4. The MIDI files were obtained from the CPDL (see Section 3.3.1). For synchronization, we used the dynamic time warping pipeline from Ewert et al. [61] and Müller et al. [123] that uses the beat annotations as anchor points for the alignment. In order to facilitate data parsing and processing, we converted the aligned MIDI files to CSV files using `pretty_midi` [144], a Python library for processing and converting MIDI files. For each STM signal, DCS contains one separate CSV file per section (as opposed to MIDI files that include all sections). Each CSV file contains three columns, which represent note onset in

Table 3.4: Evaluation results for pYIN trajectories averaged over two quartet recordings.

Mic	VR	VFA	RPA	RCA	OA
LRX	**0.99 (0.00)**	**0.11 (0.06)**	**0.95 (0.02)**	**0.95 (0.01)**	**0.93 (0.03)**
HSM	0.98 (0.01)	0.33 (0.09)	0.81 (0.10)	0.91 (0.04)	0.77 (0.08)
DYN	**0.99 (0.00)**	0.16 (0.11)	0.93 (0.04)	**0.95 (0.01)**	0.90 (0.05)

Table 3.5: Evaluation results for CREPE trajectories averaged over two quartet recordings.

Mic	VR	VFA	RPA	RCA	OA
LRX	**0.96 (0.01)**	**0.12 (0.02)**	**0.96 (0.01)**	**0.96 (0.01)**	**0.93 (0.02)**
HSM	0.92 (0.02)	0.32 (0.08)	0.91 (0.01)	0.91 (0.02)	0.84 (0.02)
DYN	0.93 (0.01)	0.18 (0.07)	0.93 (0.01)	0.93 (0.01)	0.90 (0.02)

seconds, note offset in seconds, and MIDI pitch. The number of rows is equal to the number of notes in the piece.

3.3.6 Fundamental Frequency Trajectories

One of the most important cues for computational studies on choral singing and choral intonation are the F0-trajectories of the individual singers' voices [40, 46, 47]. However, annotating F0-trajectories from polyphonic mixtures is cumbersome and requires a lot of labor-intensive work. We exploit the multitrack nature of DCS to automatically compute the F0-trajectories of each singer from the close-up microphone signals using two state-of-the-art algorithms for monophonic F0-estimation: pYIN [109] (see Section 2.3.3) and CREPE [97] (see Section 2.3.5).

The pYIN annotations were obtained using the pYIN Vamp Plug-in[27] for Sonic Annotator [28]. For pYIN, we used an FFT size of 2048 and a hop size of 221 samples, which corresponds to around 10 ms for a sampling rate of 22 050 Hz. We used the algorithm in the `smoothedpitchtrack` mode, which uses an HMM and Viterbi decoding to smooth the F0-estimates. In addition, we configured the plugin to output negative F0-values in frames that are estimated as unvoiced (`outputunvoiced=2`) as well as the probability of each frame to be voiced (`output=voicedprob`). For CREPE, we used the CREPE Python package[28] with the model capacity set to `full`, Viterbi smoothing activated, a default hop size of 10 ms, and a default input size of 1024 samples. Similar hop sizes were used with both methods for an easier comparison. The F0-trajectories are stored in CSV files with three columns. The first two columns contain the timestamps in seconds and the F0-values in Hz. In the case of pYIN, the third column contains the probabilities of the frames to be voiced. In the case of CREPE, the third column contains the confidence as provided by the algorithm. The confidence is a number between 0 and 1 that indicates the reliability of an F0-estimate.

In order to validate the automatically extracted F0-trajectories, we generated manual F0-annotations for all voices of two quartet recordings based on the LRX signals. The annotations were made by a sound

[27] `https://code.soundsoftware.ac.uk/projects/pyin`

[28] `https://github.com/marl/crepe`

engineer with over ten years' training on saxophone using the tool Tony [111] and are included in DCS as CSV files. For evaluation, we use common evaluation metrics for melody extraction as detailed by Poliner et al. [139], Salamon et al. [170]. The metrics Voicing Recall (`VR`) and Voicing False Alarm (`VFA`) measure the accuracy of the algorithm's voice activity estimation. The metrics Raw Pitch Accuracy (`RPA`) and Raw Chroma Accuracy (`RCA`) measure the proportion of frames for which the estimated F0-trajectory lies within 50 cents (half a semitone) of the reference (`RCA` ignores octave errors). Additionally, the Overall Accuracy (`OA`) is a combined metric that accounts for both voice activity and F0-accuracy. We use the open source toolbox `mir_eval` [145] to compute the evaluation metrics. In our experiments, we derive the voice activity for F0-trajectories extracted by CREPE by choosing a confidence threshold that maximizes the overall accuracy. The evaluation results averaged over the two recordings for pYIN and CREPE (8 LRX, 6 HSM, 8 DYN trajectories per algorithm) are given in Tables 3.4 and 3.5, respectively. The standard deviations are given in brackets. Both algorithms perform most accurately on the LRX signals (0.93 of overall accuracy), slightly less accurate on DYN signals and least accurate on HSM signals. This is expected, since the F0 of the voice is more dominant in LRX signals than in DYN or HSM signals (see Section 3.3.2). The overall performance of both algorithms is similar on LRX and DYN signals and deviates for HSM signals, where CREPE performs better than pYIN.

In the following, we further analyze the differences between the microphone signals. Figure 3.3 illustrates the F0-trajectories from a tenor singer extracted from LRX and DYN signals using CREPE. The CREPE confidence values are depicted in Figure 3.3b. For visualization purposes, the confidences are smoothed with a median filter of length 210 ms. Thresholding the smoothed confidence values with a threshold of 0.935 for the LRX confidence and a threshold of 0.9 for the DYN confidence leads to the binary activations depicted in Figure 3.3c and the F0-trajectories depicted in Figure 3.3a. Note that the thresholds are chosen exemplarily to show the differences between the microphones. In Part I, CREPE shows similar confidence values for both microphone signals when the tenor is singing. Part II shows significant differences between the two microphones. In this part, low confidence values are expected since the tenor is not active. Still, CREPE shows some confidence for both microphone signals due to cross-talk of the bass voice. However, one can find a suitable threshold for the LRX confidence to avoid an F0-output. Since the cross-talk is much stronger in the DYN signal, there exists no meaningful threshold that suppresses any F0-output in Part II of the DYN signal. In Part III, the F0-trajectory of the DYN microphone suffers from confusions with the bass voice even though the tenor is singing.

3.4 Interfaces

The main goal of our work is to create a freely available and easy-to-access dataset in order to support MIR research on a cappella choral music. To this end, we provide several interfaces to interact with the dataset. As the most important step, we make the dataset publicly available in order to support scientific exchange and ensure reproducibility of scientific results. We decided to host DCS on Zenodo[18] , an Open

Science platform, which supports sharing and distributing scientific data. As main features, the platform provides versioning and citeable Digital Object Identifiers (DOIs) for uploaded data.

However, Zenodo is a data repository and does not offer to play back the audio files in the browser. The interdisciplinary field of MIR benefits from interfaces that help to lower access barriers to datasets by providing direct, intuitive, and comprehensive access. This can be accomplished by means of interactive interfaces, e.g., with playback functionalities [66, 91, 167]. As one contribution, we created a publicly accessible web-based interface[19], which hosts the multitrack audio data. The entry page of the interface is subdivided into a "Music Recordings" section providing links to the *Locus Iste* and *Tebe Poem* recordings as well as a "Systematic Exercises and Additional Recordings" section. Furthermore, the interface allows for searching and sorting of specific recordings. Each multitrack recording has an individual sub-page with an open source audio player [216] with score-following functionality [221] that allows for seamless switching between the different tracks.

Along with web-based interfaces, accompanying dataset-specific processing tools simplify the usage of datasets [11, 15]. We created a Python toolbox named `DCStoolbox`[20] that accompanies the release of the dataset. The toolbox provides basic functions to parse and load data from DCS, which are demonstrated in a Jupyter notebook. Furthermore, it includes scripts to reproduce the computed F0-trajectories from Section 3.3.6 and an Anaconda[29] environment file that specifies all Python packages required to run the toolbox functions. Additionally, we provide access to DCS via *mirdata* [15], a Python package that includes functionalities for downloading, validating, and parsing MIR datasets.

3.5 Applications to MIR Research

In this section, we demonstrate the potential of DCS for MIR research by means of two case studies. In the first case study discussed in Section 3.5.1, the goal is to evaluate and compare the intonation quality of quartet performances using a recently published intonation measure [214]. In the second case study, conducted in Section 3.5.2, we consider the task of multiple-F0 estimation. More specifically, we apply a state-of-the-art approach [13] on different recordings and show the benefits of our multitrack recordings for multiple-F0 estimation in polyphonic vocal music.

3.5.1 Intonation Quality of Quartet Performances

A central challenge for a cappella singers is the adjustment of pitch in order to stay in tune relative to the fellow singers. Even if choirs achieve good local intonation, they may suffer from intonation drifts slowly evolving over time [50]. Algorithms that attempt to measure intonation quality have to account for such intonation drifts. A recently published approach measures the distance between the recording's

[29] https://www.anaconda.com/distribution/

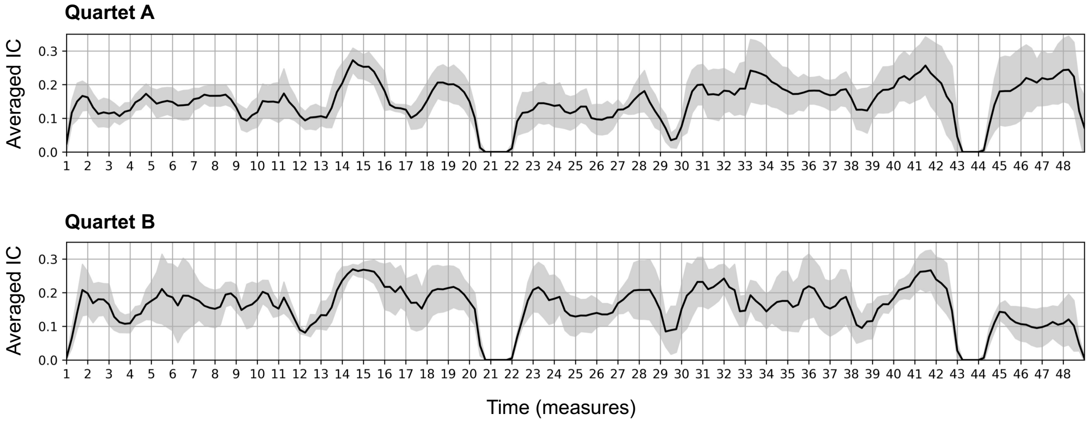


Figure 3.5: Averaged intonation cost (IC) measures for six takes of *Locus Iste* by Quartet A and five takes by Quartet B. The local standard deviations are indicated in light grey.

local salient frequency content and a shifted 12-tone equal-temperament (12-TET) grid [214]. Although choirs often aim for just intonation, the 12-TET scale has been used to approximate intonation in Western choral performances [69]. The intonation measure requires as input the F0s and harmonic partials (integer multiples of the F0) together with their respective amplitudes for the four singing voices. In a frame-wise fashion, a grid-shift parameter is computed that minimizes the distance between the F0s/partials and the shifted 12-TET grid. As output, the approach returns a frame-wise intonation cost (IC) that reflects the remaining distance from the optimally shifted 12-TET grid. The IC is bounded in the interval [0, 1], where small values indicate good local intonation, and large values indicate local intonation deviations. In the following, we use this approach to compare the performances of Quartet A and B in our DCS.

Weiß et al. [214] show that multitrack recordings of the individual voices are beneficial for estimating the frequency and amplitude information required to compute the IC. For our case study, we make use of the recorded LRX and DYN signals as follows. We obtain the frequency information from the extracted pYIN F0-trajectories of the LRX signals (see Section 3.3.6). Using the time-aligned score representations from Section 3.3.5, we restrict the trajectories to regions where the respective voices are active. We obtain the amplitude information from a magnitude spectrogram representation of the DYN signals at the locations of the extracted LRX F0-trajectories and their harmonic partials. In our experiments, we consider 16 harmonic partials. Subsequently, we compute IC measure curves for all quartet recordings of *Locus Iste* in DCS. In order to compare the different takes, we map the curves on a common time axis in measures using the measure information encoded in the beat annotations from Section 3.3.4.

The averaged IC curves for six recordings of Quartet A and five recordings of Quartet B are depicted in Figure 3.5. To remove local outliers, we post-process the IC curves using a moving median filter of length 21 frames. Note that the IC is zero for silent regions and small for monophonic passages where only one singer is active (see measures 12, 20/21, and 43). Overall, the curves exhibit a similar progression. For both curves, we observe higher IC values in the passage from measures 13 to 20. This passage is challenging to sing due to the highly chromatic voice leading and the jumps in the bass part. Furthermore, the passage from measure 40 to 42 exhibits higher ICs for both quartets—a passage which is highly chromatic. The largest differences between the quartets can be found in the last part of the piece (measures 44 to 48). For this passage, Quartet B achieves a better intonation quality on average than Quartet A, especially in the intonation of the final chord of the piece.

This short case study indicates the potential of our recordings for studying intonation in polyphonic a cappella music. Furthermore, our data can form a starting point for future studies on singer interaction in amateur choirs.

3.5.2 Multiple-F0 Estimation in A Cappella Singing

Multiple-F0 estimation is defined as the task of estimating the F0s of several concurrent sounds in a polyphonic signal [98, 99]. This task is particularly challenging for polyphonic vocal music [184, 196]. In a cappella choral singing, we find multiple singers with similar timbres singing in harmony, thus producing overlapping harmonics [41]. Furthermore, it is very common that several singers sing the same part (unison), but produce slightly different frequencies. However, MIR research on multiple-F0 estimation in polyphonic vocal music has so far been focusing on SATB quartets and there exist no suitable methods for multiple-F0 estimation in larger ensembles with multiple singers per part. The Full Choir recordings in DCS constitue a starting point for further research in this direction.

In the following, we show the potential of DCS by applying a state-of-the-art multiple-F0 estimation algorithm on different scenarios offered by the DCS quartet recordings. The first scenario consists of applying the algorithm on a mix of all DYN signals. In the second and third scenario, the algorithm is applied on the STM signal (room microphone) with and without additional reverb. In particular, we consider the recordings of *Locus Iste* from Quartet A (Take 3).

In our case study, we use the *DeepSalience* method [13], a deep convolutional neural network trained to produce a pitch salience representation (enhanced time–frequency representation) of the input signal, which contains values in the range $[0, 1]$. This salience representation is thresholded such that only time–frequency bins with a salience value above the chosen threshold remain. These remaining bins correspond to the multiple-F0 estimates. Although the model is not specifically trained for polyphonic vocal music, it was found to obtain the best performance for multiple-F0 estimation in vocal quartets [41].

For the evaluation, we exploit the multitrack nature of DCS. In particular, we take the previously extracted pYIN F0-trajectories from the LRX signals as reference (see Section 3.3.6). Note that these trajectories are the output of an algorithm. Although our evaluation reveals they are very accurate (see Table 3.4), they still contain some errors. As evaluation metrics, we use the standard multiple-F0 estimation metrics Precision, Recall, and F-Score. For a detailed description of these metrics, we refer to Bittner [9, Chapter II, Section 6.3]. The evaluation metrics were computed using the `mir_eval` library [145].

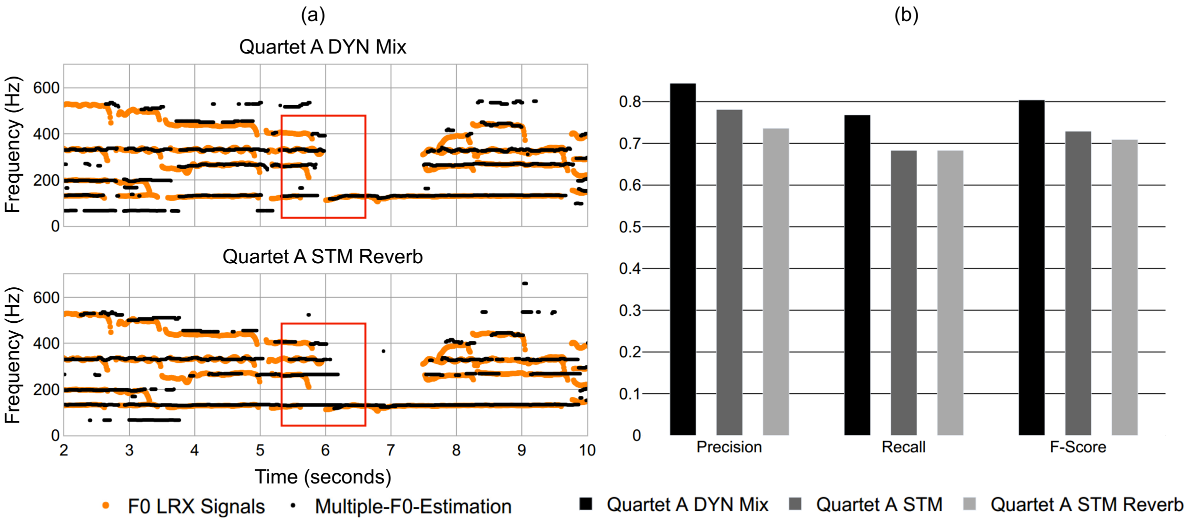


Figure 3.6: Multiple-F0 estimation using *DeepSalience* [13] with a threshold of 0.1. **(a)** Estimation results (excerpts) for the mix of DYN signals and the STM signal with reverb. **(b)** Evaluation metrics for all scenarios.

We experimented with several thresholds between 0.05 and 0.5, and found 0.1 to obtain the best results on the studied quartet recordings with respect to our evaluation metrics. However, instead of comparing absolute values (which is problematic for automatically extracted reference F0-trajectories), we want to focus on relative differences between the different scenarios. Figure 3.6a shows excerpts of the computed multiple-F0 estimates for the mix of DYN signals and the STM signal with reverb obtained by thresholding the salience representations with a threshold of 0.1. Figure 3.6b shows the evaluation results for all three scenarios. From the F-Score values, we observe that the algorithm performs best for the DYN signal mix of Quartet A. Furthermore, we observe that an increasing amount of reverb in the recordings goes along with a decreasing overall performance of the algorithm. This indicates that reverb further complicates the task of multiple-F0 estimation. The Precision and Recall measures give further insights into this observation. While Precision is lower in the scenario with reverb, Recall is not affected. In reverb conditions, sung notes become temporally smeared, leading to a temporal mismatch between the reference F0-trajectories from the LRX signals and the audio recording. For this reason, the number of false positives increases, causing Precision to decrease. This effect can be seen by comparing the red marked areas in Figure 3.6a. We leave a more detailed analysis of these effects to future studies.

In summary, this brief case study indicates that the DCS is a versatile and challenging resource to develop and test algorithms for multiple-F0 estimation in polyphonic a cappella vocal music. Furthermore, the time-aligned score representations could serve as a reference for the evaluation of note-tracking algorithms. This requires accounting for intonation drifts of the choirs, which can, e.g., be determined from the F0-annotations.

3.6 Conclusions and Further Notes

In this chapter, we presented Dagstuhl ChoirSet—a publicly accessible multitrack dataset of a cappella choral music for MIR research. This work is based on our recordings of an amateur vocal ensemble we gathered at an MIR seminar at Schloss Dagstuhl. As main feature of the dataset, the singers were recorded using different close-up microphones including dynamic, headset, and larynx microphones. As part of our work, we curated the recorded material and manually generated beat annotations as well as time-aligned sheet music representations. Furthermore, we automatically extracted F0-trajectories for all close-up microphone tracks. The dataset is released together with an interactive web-based interface and a Python toolbox to provide convenient access. In summary, the different musical and acoustical dimensions of DCS open up a variety of new and challenging scenarios for MIR research. Additionally, as part of the European research project TROMPA[30], several multitrack datasets of choral singing have been created, e.g., the Cantoría Dataset[31] and the ESMUC Choir Dataset[32]. A description of the datasets can be found in [39]. Together with Dagstuhl ChoirSet, these data sources provide a basis for applying recent data-driven methods for analyzing choral singing.

[30] https://trompamusic.eu/

[31] https://zenodo.org/record/5878677

[32] https://zenodo.org/record/5848990

4 An Adaptive Pitch-Shifting Approach for Intonation Adjustment

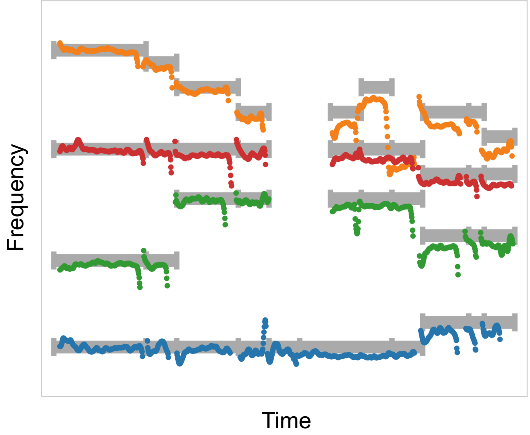
This chapter is based on [163]. The first author Sebastian Rosenzweig is the main contributor of this article. In collaboration with his supervisor Meinard Müller and Jonathan Driedger, he developed the ideas, formalized the approach, implemented the toolbox, and wrote the paper. Simon Schwär conducted some of the experiments under the supervison of Sebastian Rosenzweig.

A central challenge for a cappella singers is to adjust their intonation and to stay in tune relative to their fellow singers. During editing of a cappella recordings, one may want to adjust local intonation problems of individual singers or to account for global intonation drifts over time. This requires applying a time-varying pitch shift to the audio recording, which we refer to as adaptive pitch-shifting. In this context, existing (semi-)automatic approaches are either labor-intensive or face technical and musical limitations. In this chapter, we present automatic methods and tools for adaptive pitch-shifting with applications to intonation adjustment in a cappella recordings. Motivated by this application, we show how to incorporate time-varying information into existing pitch-shifting algorithms that are based on resampling and time-scale modification (TSM). Furthermore, we release an open-source Python toolbox, which includes a variety of TSM algorithms and an implementation of our method. Finally, we show the potential of our tools by two case studies on global and local intonation adjustment using a cappella recordings from Dagstuhl ChoirSet (see Chapter 3).

4.1 Introduction

A cappella singing is a wide-spread vocal performance practice where one or multiple singers sing together without instrumental accompaniment. Without having an instrumental reference, it becomes crucial that a cappella singers adjust their pitch relative to their fellow singers [1, 79]. Performances (in particular of amateur or semi-professional ensembles) can exhibit different kinds of intonation inaccuracies, ranging from individual, local intonation problems (e.g., singers singing a note too low or too high) to global

Figure 4.1: F0-trajectories of a four-voice a cappella performance (soprano=orange, alto=red, tenor=green, bass=blue). The score reference is indicated in grey.



intonation drifts over time [1, 50, 86, 110]. Figure 4.1 exemplifies such inaccuracies with an excerpt from an SATB quartet performance, showing F0-trajectories on top of a reference derived from a musical score (visualized in gray). The figure illustrates two phenomena: first, the performance exhibits local intonation inaccuracies such as for the tenor voice (green), which sings the beginning of the first note slightly too low. Second, the performance exhibits a global intonation drift downwards over the course of the excerpt (all four F0-trajectories lay below the gray score reference at the end of the excerpt).

During postprocessing of a cappella recordings, one may want to adjust local or global intonation deviations using pitch-shifting techniques. Pitch-shifting is the task of changing an audio recording's pitch without altering its duration. Over the last decades, several conceptually different approaches have been proposed in the literature, ranging from time-domain algorithms [21, 33, 81, 108] to frequency domain approaches [56, 182]. An overview on several pitch-shifting approaches can be found in [224]. However, for adjusting local and global intonation in a cappella recordings, it is not sufficient to apply a single fixed pitch shift to the recording, as Figure 4.1 demonstrates. Instead, it is necessary to apply a time-varying pitch shift to the audio recording, which we refer to as adaptive pitch-shifting.

A naïve approach for adaptive pitch-shifting is to apply individual pitch shifts to small sections of an audio signal, e.g., using user-guided functionalities provided by most digital audio workstations. However, besides being labor-intensive, this approach can lead to audible "clicking" artifacts at pitch shift transitions due to phase and other discontinuities. Previous research on adaptive pitch-shifting has been conducted in the context of audio restoration and "wow" reduction of gramophone and tape recordings [44, 45, 71]. Recently, a deep learning-based approach for adaptive pitch correction of singing performances with instrumental accompaniment has been proposed in [213]. State-of-the-art commercial tools such as Melodyne[33] or Antares AutoTune[34] offer semi-automatic functionalities for pitch correction according to different scales and tunings. However, due to the presence of global intonation drifts over time and a varying local intonation depending on the musical context, the assumption of a fixed (time-invariant) scale

[33] https://www.celemony.com/en/melodyne
[34] https://www.antarestech.com

or tuning is problematic for a cappella music [1]. Popular open-source music processing libraries such as librosa [112] are often limited to fixed pitch-shifting functionalities. As an exception, the C++ library Rubber Band[35] provides an interface for real-time pitch-shifting of an audio stream. Furthermore, the PyTSMod package [220] includes an adaptive pitch shift implementation designed for monophonic audio.

In this chapter, we propose automatic methods and tools for adaptive pitch-shifting with applications to intonation adjustment in a cappella recordings. We base our work on an existing pitch-shifting method, which makes use of resampling and time-scale modification (TSM) [56]. As one contribution, we propose and formalize an extension to this method, which enables time-varying pitch shifts. Furthermore, we release a Python re-implementation of a Matlab TSM toolbox [55] called `libtsm`, which we extended with an implementation of our adaptive pitch-shifting method. In order to show the potential of our method, we consider two case studies based on Dagstuhl ChoirSet (DCS, see Chapter 3 and [159]), a publicly available dataset of a cappella performances. The first study targets the adjustment of global intonation, whereas our second study targets the adjustment of local intonation.

The remainder of this chapter is structured as follows. In Section 4.2, we review pitch-shifting via resampling in combination with TSM and introduce our adaptive pitch-shifting method. In Section 4.3, we give details on our Python toolbox and in Section 4.4, we address our two case studies. Finally, we summarize our work and provide further notes on research based on our toolbox in Section 4.5.

4.2 Pitch-Shifting via Resampling and TSM

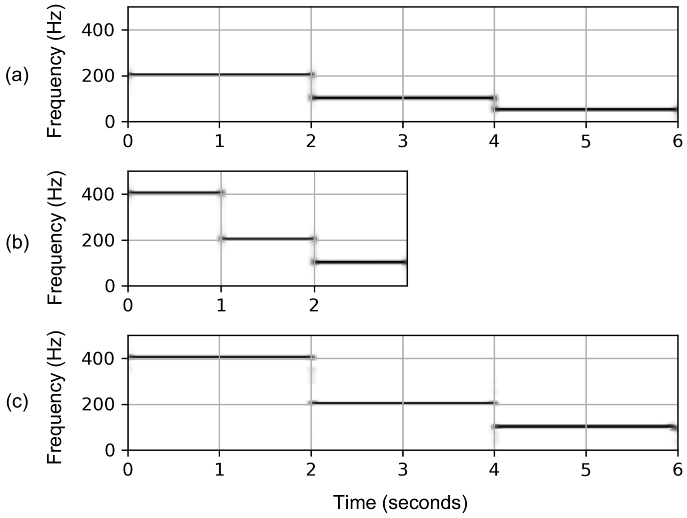
Pitch-shifting can be seen as the complementary task to TSM [56, 224]. While TSM attempts to alter the duration of an audio recording without changing its pitch, pitch-shifting attempts to alter the pitch of an audio recording without changing its duration. In the following, we summarize existing TSM algorithms (Section 4.2.1), explain the basic principle of fixed (time-invariant) pitch-shifting using resampling and TSM (Section 4.2.2), and finally introduce our adaptive pitch-shifting method (Section 4.2.3).

4.2.1 TSM Algorithms

Over the last decades, several TSM algorithms have been proposed. In general, TSM algorithms can be subdivided into time-domain and frequency-domain approaches. Time-domain approaches typically rely on variants of the overlap-add (OLA) principle. In this case, an input signal is first decomposed into overlapping frames, which are relocated on the time axis in a second step to achieve the actual time-scale modification. Examples of time-domain algorithms are SOLA (*Synchronized* OLA) [166], TD-PSOLA (*Time-Domain Pitch-Synchronized* OLA) [23, 33, 119] or WSOLA (*Waveform-Similarity* OLA) [210]. A well-known frequency-domain approach is based on the phase vocoder technique [63, 140]. In order

[35] `https://breakfastquay.com/rubberband/`

Figure 4.2: Pitch-shifting via resampling and TSM illustrated using power spectrograms. **(a)** Input signal. **(b)** Resampled signal. **(c)** Pitch-shifted signal after TSM application.



to obtain a time-scaled version of the input signal, the method relocates the frames of the input signal's STFT (see Section 2.1.3) and applies a frequency-dependent phase correction. Recent works on TSM propose modifications of the phase vocoder technique [141] or use the phase vocoder in combination with non-negative matrix factorization [156]. While time-domain TSM methods are known to be well-suited for recordings with strong transient sound components, frequency-domain approaches typically perform well on recordings with strong harmonic sound components. This observation has been exploited by the approach in [57], which first conducts harmonic–percussive separation (HPS) [62] and then applies OLA on the percussive component and the technique based on the phase vocoder on the harmonic component. A more detailed review of several TSM methods can be found in [56].

4.2.2 Fixed Pitch-Shifting

Resampling a given audio signal and playing it back at the original sampling rate changes its duration and pitch at the same time. In other words, resampling can be interpreted as a TSM procedure that additionally modifies the pitch of an audio signal. Pitch-preserving TSM algorithms, such as the ones mentioned in Section 4.2.1, can be used to compensate for the change in duration after resampling. Note that pitch-shifting can also be achieved by processing in reverse order (first performing TSM and then resampling) [224].

The processing steps for fixed (time-invariant) pitch-shifting via resampling and subsequent TSM are illustrated in Figure 4.2. For illustrative purposes, we use a synthetic signal as input signal, which contains three sequentially played sinusoidal tones. Figure 4.2a shows a power spectrogram of our input signal. Let us assume our input signal is equidistantly sampled at a rate F_s^in and we are given a fixed pitch shift $p \in \mathbb{R}$ in cents. In a first step, we resample the given signal to have a new sampling rate $F_\mathrm{s}^\mathrm{out}$ defined by

$$F_\mathrm{s}^\mathrm{out} := F_\mathrm{s}^\mathrm{in} \cdot 2^{-p/1200}. \tag{4.1}$$

When playing back the resampled signal at the original sampling rate F_s^in, one can observe two effects. First, the signal's duration is scaled by a factor $\alpha_\mathrm{RS} \in \mathbb{R}_{>0}$ defined as

$$\alpha_\mathrm{RS} := \frac{F_\mathrm{s}^\mathrm{out}}{F_\mathrm{s}^\mathrm{in}} = 2^{-p/1200}. \tag{4.2}$$

Second, the signal is pitch-shifted by p cents. These two effects can be seen in Figure 4.2b for a pitch shift of $p = 1200$ cents, which is equivalent to an octave in musical terms or a doubling of frequency in physical terms.

To compensate for the undesired time-scale modification, we then use a suitable pitch-preserving TSM algorithm to scale the signal to it's original duration. To this end, we stretch the signal with the factor $\alpha_\mathrm{TSM} \in \mathbb{R}_{>0}$ defined by

$$\alpha_\mathrm{TSM} := \alpha_\mathrm{RS}^{-1} = 2^{p/1200}. \tag{4.3}$$

For a pitch shift of $p = 1200$ cents we obtain $\alpha_\mathrm{TSM} = 2$. The resulting pitch-shifted signal is depicted in Figure 4.2c.

4.2.3 Adaptive Pitch-Shifting

Adaptive pitch-shifting is the task of applying a time-varying pitch shift to an audio signal. To this end, we extend the method for fixed pitch-shifting from Section 4.2.2. More specifically, we combine non-linear resampling with a technique referred to as non-linear TSM [56]. In the following, we explain our approach along with the example depicted in Figure 4.3.

Let us assume we are given an audio signal, which is equidistantly sampled at a sampling rate of F_s^in. As illustrative example, we again consider an input signal with three sequential sinusoidal tones, as visualized in Figure 4.3a. For the task of adaptive pitch-shifting, we model the pitch shift p as a continuous time-varying function $p : \mathbb{R} \to \mathbb{R}$, which maps a time instance $t \in \mathbb{R}$ in seconds to a musical interval given in cents. Figure 4.3b shows p in our example, which consists of three parts: in the first part (0 s to 2 s), the input signal should be left unshifted, in the second part (2 s to 4 s), the signal should be frequency modulated, and in the third part (4 s to 6 s), a frequency sweep should be applied.

In a first processing step, we perform non-linear resampling of our input signal. As explained earlier, resampling can be interpreted as a kind of pitch-modifying TSM. In this light, we first define a scaling factor function $\alpha_\mathrm{RS} : \mathbb{R} \to \mathbb{R}_{>0}$ that maps a time instance t to a scaling factor by

$$\alpha_\mathrm{RS}(t) := 2^{-p(t)/1200}. \tag{4.4}$$

The resulting $\alpha_\mathrm{RS}(t)$ for our example is depicted in Figure 4.3c. Subsequently, we introduce a non-linear and strictly monotonously increasing time-stretch function $\tau_\mathrm{RS} : \mathbb{R} \to \mathbb{R}$, which defines a mapping between

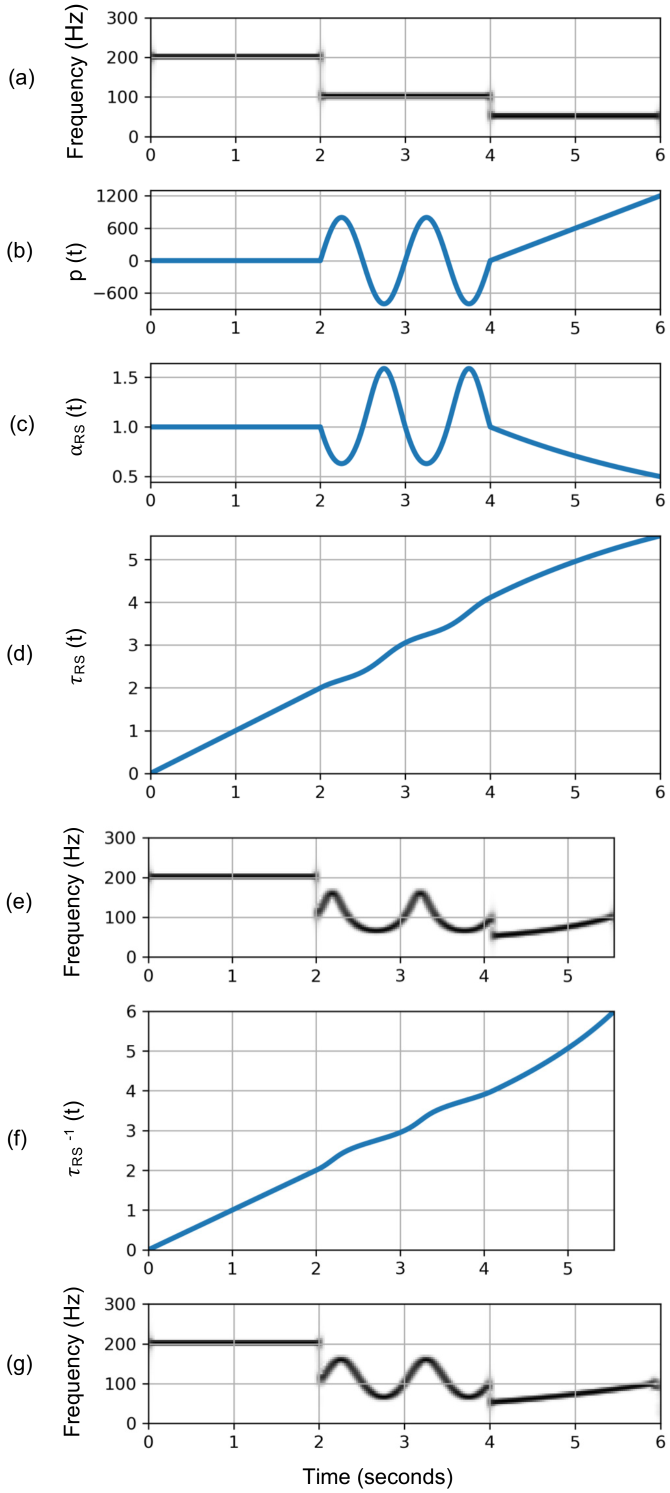


Figure 4.3: Adaptive pitch-shifting via non-linear resampling and non-linear TSM. **(a)** Power spectrogram of input signal. **(b)** pitch shift function. **(c)** Scaling factor function. **(d)** Time-stretch function. **(e)** Power spectrogram of resampled signal. **(f)** Inverse time-stretch function. **(g)** Power spectrogram of pitch-shifted signal.

time instances of an input and an output signal, by

$$\tau_{\mathrm{RS}}(t) := \int_0^t \alpha_{\mathrm{RS}}(t)\,dt. \tag{4.5}$$

The function τ_{RS} for our example is depicted in Figure 4.3d. As one can see, the first part of the function has a slope equal to one. As a consequence, this part of our example signal is mapped to the output signal without modification. The overall slope of the function's second part is slightly larger than one, leading to an expansion of this region in the output signal. The overall slope of the function's third part is slightly smaller than one, leading to a compression of this region in the output signal. By performing non-linear resampling according to the mapping defined by the function τ_{RS}, we obtain the signal depicted in Figure 4.3e.

Note that in practice, non-linear resampling can be done in many different ways [192, 207]. A comparison of resampling implementations in digital audio workstations can be found online[36]. Advanced resampling methods such as multirate filterbanks include lowpass filtering to avoid aliasing artefacts, but also require a windowing of the time-stretch function τ_{RS}. However, our goal is to adjust intonation with pitch shifts in the order of a few cents up to roughly a semitone, where aliasing artefacts are less problematic. For the sake of simplicity, we therefore use cubic interpolation to non-linearly resample the input signal.

In a second processing step, we perform non-linear TSM on the resampled audio signal to retain the signal's original duration. To this end, we use a pitch-preserving TSM algorithm to non-linearly stretch the signal with respect to τ_{RS}^{-1}, which is depicted in Figure 4.3f. Further details on non-linear TSM can be found in [56, Section 7.1]. The resulting pitch-shifted audio signal is depicted in Figure 4.3g. As one can see, the adaptive pitch shift p has been applied to our input signal.

4.3 Python Toolbox

The release of open-source implementations along with scientific publications has become increasingly important in the field of music signal processing [113, 224]. Besides allowing for reproducing experimental results, publicly available implementations stimulate and support further research activities. In this spirit, we ported an exisiting Matlab TSM toolbox [55] to Python and expanded its functionality with our adaptive pitch-shifting method. Python is currently considered as the most used programming language in data science and machine learning. Our Python TSM toolbox is released under an open source license[37].

[36] https://src.infinitewave.ca/

[37] https://github.com/meinardmueller/libtsm

Algorithm	Matlab Function	Python Function
WSOLA/OLA [210]	`wsolaTSM()`	`wsola_tsm()`
Phase Vocoder TSM [63, 140]	`pvTSM()`	`pv_tsm()`
Harmonic–Percussive Separation TSM [57]	`hpTSM()`	`hps_tsm()`
Fixed pitch-shifting	`pitchShiftViaTSM()`	`pitch_shift()`
Adaptive pitch-shifting	-	`pitch_shift()`

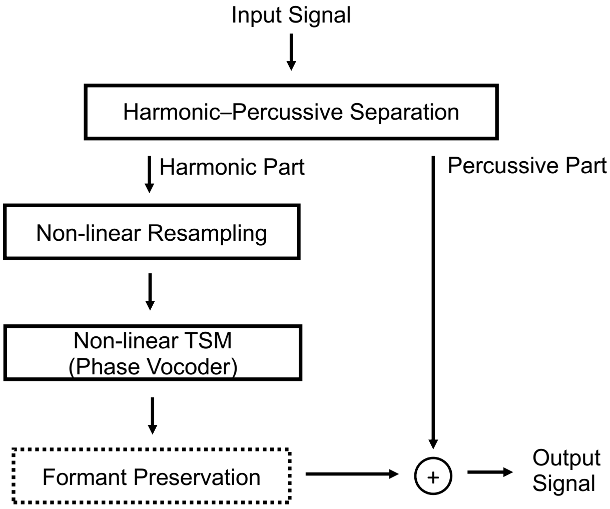
Table 4.1: Main algorithms and implementations of the Matlab TSM toolbox and `libtsm`.

```
# Load packages
import libtsm
import librosa
import numpy as np

# Load Audio File
fn_in = 'data/three_sinusoidals.wav'
x, Fs = librosa.load(fn_in, sr=22050)

# TSM Algorithms
alpha = 1.8  # scaling factor

y_wsola = libtsm.wsola_tsm(x, alpha)
y_pv = libtsm.pv_tsm(x, alpha)
y_hps = libtsm.hps_tsm(x, alpha)

# Fixed pitch-shifting (Figure 2)
p = 1200  # cents
y_psf = libtsm.pitch_shift(x, p)

# Adaptive pitch-shifting (Figure 3)
t = np.arange(0, len(x)/Fs, 1/Fs)  # sec
N = len(t)
t_1 = t[0:N//3]
t_2 = t[N//3:2*N//3]
t_3 = t[2*N//3:]

p = np.concatenate((np.zeros(len(t_1)),
 ↪  800*np.sin(2*np.pi*1*t_2),
 ↪  np.linspace(0,1200,len(t_3))))  # cents
y_psa = libtsm.pitch_shift(x, p, t)
```

Listing 1: Code example using functions of `libtsm`.

In our re-implementation of the toolbox, we ensured that the naming conventions and usage of our Python implementation are basically the same as in the Matlab version. Table 4.1 provides an overview on the main algorithms, functions, and naming conventions of the Matlab and the Python toolbox. Furthermore, we tested all Python functions with respect to numerical identity to the Matlab implementations. In the following, we demonstrate the main functions of the Python toolbox using the code example in Listing 1.

Figure 4.4: Overview on our intonation adjustment setup.



As one can see in line 2, the TSM toolbox can be imported as a Python package `libtsm`. The toolbox includes short demo audio files, including our synthetic audio example from Section 4.2.2 and Section 4.2.3, which is loaded in lines 7–8. Lines 11–15 demonstrate the main TSM functions of the toolbox called with default settings. Note that each of the functions provides various other input arguments to tune the parameters of the algorithms. The input arguments are documented in the functions' docstrings.

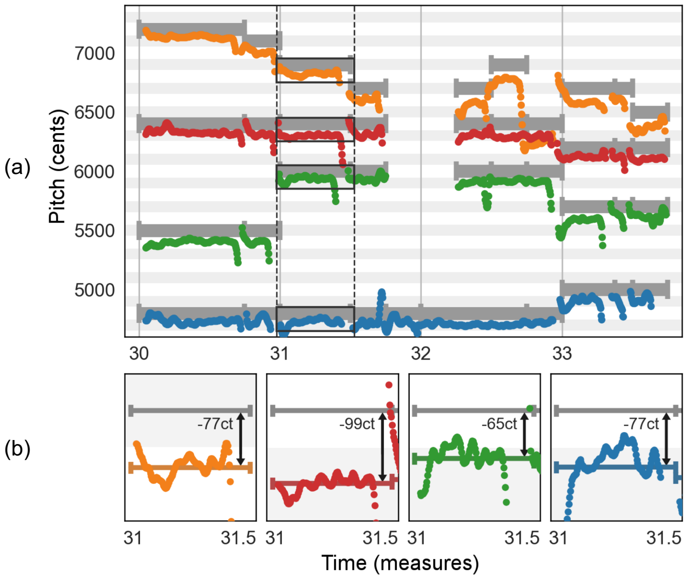
Along with the TSM implementations, we added a function `pitch_shift()` to the toolbox, which implements our fixed and adaptive pitch-shifting algorithm. Lines 18–19 replicate the fixed pitch shift by 1200 cents, as visualized in Figure 4.2. Adaptive pitch-shifting can be achieved using the same function by handing over two arrays of equal length, as shown in lines 22–28. The first array contains the pitch shift values in cents, whereas the second array contains the time axis in seconds. Our example replicates the adaptive pitch shift shown in Figure 4.3. A more detailed demonstration of all toolbox functions can be found in the Jupyter notebook `demo_libtsm.ipynb`, which is part of our toolbox.

4.4 Application: Intonation Adjustment in A Cappella Recordings

In the previous sections, we have presented a method for adaptive pitch-shifting (Section 4.2.3) as well as a Python toolbox with implementations of our method and a variety of TSM algorithms (Section 4.3). In this section, we show the potential of our method and our tools for adjusting global and local intonation in a cappella recordings.

As indicated in Section 4.2, the technical realization of our adaptive pitch-shifting method, in particular, the choice of a suitable resampling and TSM algorithm, depends on the acoustic properties of the input signal. In our application scenario, the versatility of the human voice imposes additional challenges on our pitch-shifting setup. Especially, an appropriate handling of fricatives, plosives, and formants is required to aviod a degradation of the audio quality. In the following, we present an extension to our adaptive

Figure 4.5: **(a)** Excerpt of F0-trajectories and score reference for a performance of *Locus Iste* (DCS, Quartet B, Take 3, measures 30–34). **(b)** Detailed view of the notes on the first beat in measure 31. Horizontal lines represent 12-TET pitch of the note (dark grey) and the median of the respective F0-trajectories (S=orange, A=red, T=green, B=blue).



pitch-shifting method that accounts for these challenges. Our setup is depicted in Figure 4.4. Similar to the approach in [57], we first apply HPS on the input signal. In a vocal recording, the percussive component typically includes fricatives, plosives, and other non-tonal background noise, whereas the harmonic component contains tonal elements. In our setup, we apply adaptive pitch-shifting only on the harmonic component. We use cubic interpolation for non-uniform resampling and the technique based on the phase vocoder for TSM. In order to avoid unnatural sounding pitch-shifted voices (sometimes referred to as the “chipmunk effect”), we include a formant preservation step [23, 119, 224] in our setup for monophonic input signals (recordings where only one voice is present). The formant preservation step first involves estimating the spectral envelopes of the original and the pitch-shifted signal from smoothed spectrogram representations. Subsequently, using the approach outlined in [54], the envelope of the pitch-shifted signal is corrected.

Note that this technical setup is only one possible way to realize adaptive pitch-shifting for our application scenario. A comparison of different pitch-shifting setups as well as a detailed evaluation of the musical quality is beyond the scope of this study and is left for future work. For an evaluation of the perceptual audio quality of the HPS-TSM approach, we refer to [57].

Given this technical setup, we show in two case studies how suitable pitch shift functions p can be computed to achieve global intonation adjustments (Section 4.4.1) and local intonation adjustments (Section 4.4.2). Our studies are based on recordings from Dagstuhl ChoirSet (see Chapter 3). Figure 4.5a shows an excerpt of an SATB quartet performance of *Locus Iste* with a global intonation drift and several local intonation issues, which serves as a running example in our case studies. Figure 4.5b provides a detailed view on local intonation deviations and pitch fluctuations. Accompanying audio examples for our case studies are available online[38].

[38] https://www.audiolabs-erlangen.de/resources/MIR/2021-DAFX-AdaptivePitchShifting

4.4.1 Case Study 1: Global Intonation Adjustment

In this case study, the task is to compensate a global intonation drift over the course of a performance. To this end, we first measure the intonation drift over time and then input the inverted drift as a function p to our adaptive pitch-shifting algorithm. One way to measure intonation drift is to compute the deviations of the singers' F0-trajectories from the time-aligned musical score. The singers in our recordings tuned to tones played on a piano right before the performance (cf. Chapter 3). Therefore, we compute the deviations to the notes' MIDI frequencies using 440 Hz as reference frequency for the note A4.

However, computing the deviations on a fine temporal level leads to highly fluctuating drift curves, which result in an unnatural "wobbling" in the pitch-shifted recording. Therefore, we introduce a temporal quantization of the measured intonation drifts. More precisely, we first compute the note-wise F0-median (see horizontal colored lines in Figure 4.5b), and then average the note-wise deviations on a measure-level. After inverting the measured intonation drift curve, we obtain the pitch shift function p, as depicted in Figure 4.6a for our excerpt. As one can see, p increases from roughly 77 cents to roughly 110 cents over the course of the excerpt, since the quartet drifts downwards.

The intonation adjustment can now be conducted either by applying the adaptive pitch shift defined by the function p on each individual singer's microphone signal or on the polyphonic room microphone signal. The drift-corrected F0-trajectories for our example are shown in Figure 4.6b and a detailed view is provided in Figure 4.6c. As one can see, the drift is adjusted over the course of the four bars, whereas the local intonation is still fluctuating around the score reference. Furthermore, all note-internal pitch fluctuations are preserved.

4.4.2 Case Study 2: Local Intonation Adjustment

In our second case study, we show how to use adaptive pitch-shifting to adjust local intonation. As opposed to Section 4.4.1, we now compute an individual pitch shift function p for each singer in the performance. To this end, we again compute the note-wise F0-median and its deviation from the aligned score reference. This time, the temporal quantization of our measured deviations remains on a note-level. By inverting the measured deviations for the individual voices, we obtain the pitch shift functions depicted in Figure 4.7a for our example. Note that adjusting local intonation to MIDI pitches in 12-TET is musically problematic in the context of Western choral music [1]. In general, the task of measuring intonation in a cappella music using computational tools is subject to ongoing scientific discussions [40, 50, 214]. Therefore, the above described strategy mainly serves illustrative purposes.

The locally adjusted F0-trajectories are depicted in Figure 4.7b, while a detailed view is provided in Figure 4.7c. In contrast to the global intonation adjustment in Figure 4.6c, we can see that after pitch-shifting, the note-wise F0-median now corresponds exactly to the 12-TET reference. Pitch variations within notes (e. g. vibrati and portamenti at the beginning of notes) are again preserved. In order to adjust

Figure 4.6: Global intonation adjustment in performance of *Locus Iste* (DCS, Quartet B, Take 3, measures 30–34). **(a)** Adaptive pitch-shifting function. **(b)** Globally adjusted F0-trajectories. **(c)** Detailed view of the notes on the first beat in measure 31.

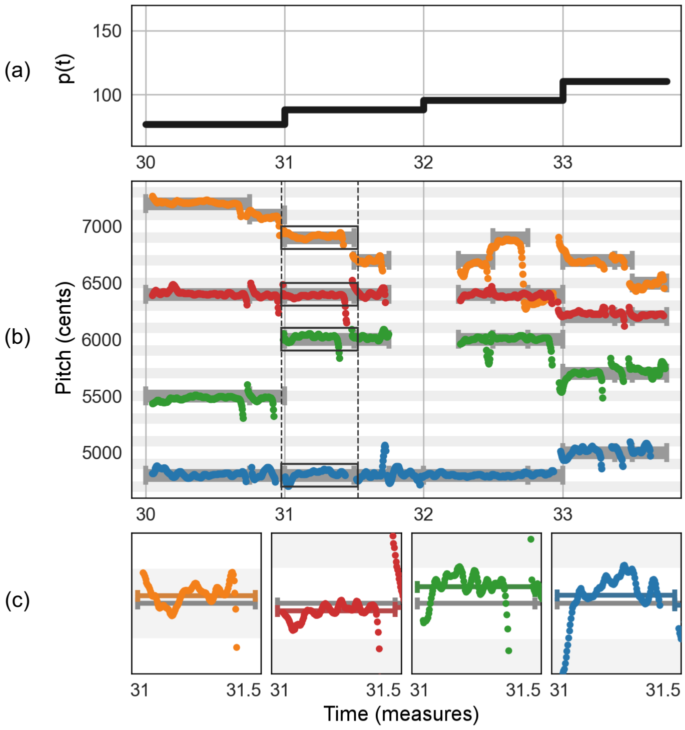
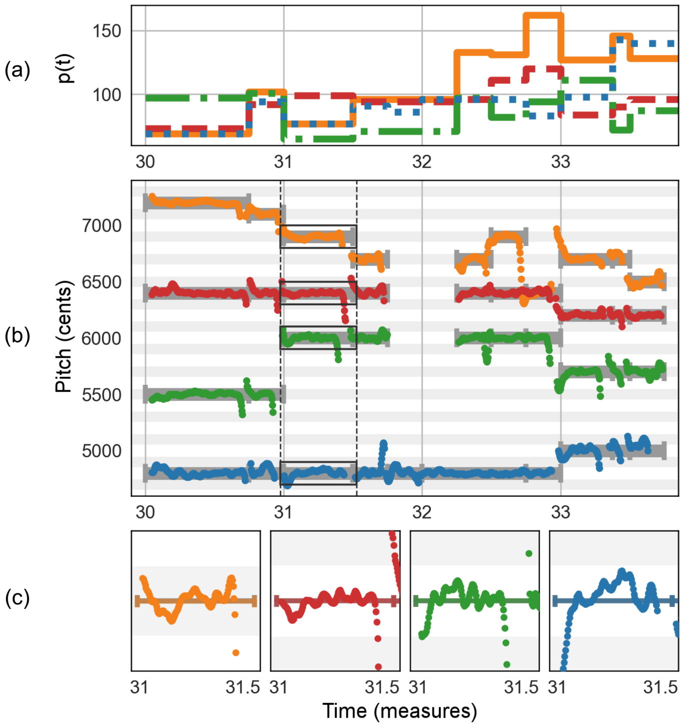


Figure 4.7: Local intonation adjustment in performance of *Locus Iste* (DCS, Quartet B, Take 3, measures 30–34). **(a)** Adaptive pitch-shifting functions for each voice. **(b)** Locally adjusted F0-trajectories. **(c)** Detailed view of the notes on the first beat in measure 31.



these fluctuations, one would have to apply pitch adjustments on a finer temporal level at the cost of an increasing unnaturalness of the pitch-shifted recordings.

4.5 Conclusions and Further Notes

In this chapter, we presented an automatic method for adaptive pitch-shifting of audio recordings based on non-linear resampling and TSM. Furthermore, we created an open source toolbox that includes implementations of various TSM algorithms and our proposed method. Finally, we showed the potential of our tools for adjusting global and local intonation in a cappella music. In our study, we measured (tonal) intonation deficiencies with respect to the 12-TET grid. While 12-TET can only be seen as a rough approximation of intonation in choral singing [110, 214], in practice, choral intonation is much more complex [1, 47, 79, 83, 86]. For instance, choir singers tend to aim for just intonation [52].

In the context of a further study using our toolbox [185], we have developed a differentiable intonation cost measure consisting of a tonal and a harmonic component, which accounts for such challenging intonation scenarios. The tonal component measures intonation deviations with respect to a fixed musical grid (e.g., the 12-TET tuning). The harmonic component uses a model of perceptual dissonance [8, 189] to capture the deviations to just intonation. A variable weight between the components allows for flexibly tuning our cost measure anywhere between 12-TET and just intonation. For a mathematical formulation of the cost measure, we refer to [185]. In combination with our adaptive pitch-shifting approach, the cost measure can be used as a flexible tool for intonation adaptation in multitrack choral music recordings[39]. Furthermore, our cost measure is differentiable and thus suitable for deep learning applications.

In future research, one may evaluate the perceptual quality of different intonation adjustment setups and investigate further cost measures for intonation processing. Furthermore, our adaptive pitch-shifting approach and the differentiable cost measure may be used to provide feedback to choir singers in rehearsal situations. One suitable platform for this scenario are interactive web-based interfaces, which record a singer's voice, analyze the intonation, and provide feedback (e.g., through interactive visualizations, sonifications, or a pitch-adjusted version of the recorded performance). A first prototype that includes a basic feedback mechanism using an interactive score representation is demonstrated in Appendix A.

[39] Audio examples: `https://www.audiolabs-erlangen.de/resources/MIR/2021-ISMIR-IntonationCostMeasure`

Part II

Analysis of Georgian Vocal Music

5 Erkomaishvili Dataset: Curation of a Corpus for Analyzing Traditional Georgian Singing

This chapter is based on [161]. The first author Sebastian Rosenzweig is the main contributor to this article. Together with his supervisor Meinard Müller and Frank Scherbaum, he developed the ideas, coordinated the annotation process, curated the dataset, conducted the experiments, and wrote the article. David Shugliashvili kindly provided his transcriptions. Vlora-Arifi Müller substantially contributed to the annotations of the dataset.

In this chapter, we present a curated dataset of traditional Georgian vocal music for computational musicology. The corpus is based on historic tape recordings of three-voice Georgian songs performed by the former master chanter Artem Erkomaishvili. In particular, we give a detailed overview of the audio material, transcriptions, and annotations contained in the dataset. Beyond its importance for ethnomusicological research, this carefully organized and annotated corpus constitutes a challenging scenario for MIR tasks such as F0-estimation, onset detection, and score-to-audio alignment. The corpus is publicly available and accessible through web-based score-following interfaces.

5.1 Introduction

The analysis of recorded audio material using computational methods has become increasingly important in musicological research [65, 124, 187]. With the goal to contribute to the preservation of the Georgian cultural heritage and to support research on Georgian vocal music, we have created a manually annotated dataset of traditional three-voice Georgian songs. The corpus is based on recordings of the former Georgian master chanter Artem Erkomaishvili (* October 26, 1887, † February 2, 1967), which were recorded in 1966 with tape recorders by the ethnomusicologist Kakhi Rosebashvili. The original recordings are preserved in the archive of the Georgian Folk Music Department of the Tbilisi State Conservatoire. 101 recordings are publicly available.[40] Due to a lack of fellow singers, Artem Erkomaishvili sung all three voices on his own, which was made possible through a three-stage overdubbing recording process. Beyond

[40] http://www.alazani.ge/old-archives-Artem-Erkomaishvilis-Sagaloblebi-folk-songs-ans59.html

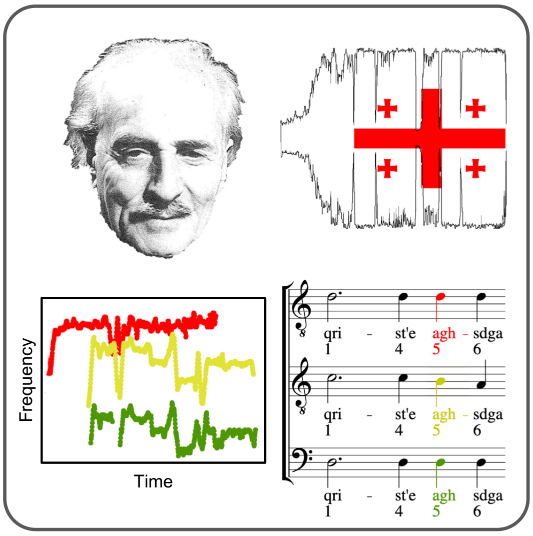


Figure 5.1: Erkomaishvili dataset with annotations. Picture of Artem Erkomaishvili from [190].

the historic recordings, there exist transcriptions of all songs in Western staff notation created by the Georgian ethnomusicologist David Shugliashvili [190]. Furthermore, Müller et al. [126] annotated the three-part recording structure and F0-trajectories for the three voices in all recordings.

Our main contributions to the Erkomaishvili dataset are threefold. First, we have collated existing audio data and annotations and introduced a uniform filename convention. Second, based on the existing transcriptions, the sheet music was converted into the digital, machine-readable MusicXML-format. Subsequently, we manually annotated note onsets of the first voice in each of the recordings. This step has been carried out by an experienced annotator with the advice of domain experts. Third, in order to provide a direct and convenient access to the dataset, we developed an interactive web-based interface with score-following audio players that make use of the annotated data. Complementing the publicly available audio material, we release all F0-annotations, recording structure annotations, and note onset annotations. Additionally, we make the MusicXML files of the symbolic transcripts publicly available.

Due to the importance of Artem Erkomaishvili's recordings for ethnomusicological research, the presented corpus is a vital source for studying tonal organization, intonation, and harmonic and melodic thinking in traditional Georgian vocal music. Furthermore, the dataset can be used for developing and testing algorithms for MIR tasks such as F0-estimation, onset detection, or score-to-audio alignment.

The remainder of this chapter is organized as follows. First, we highlight related corpora and open-source tools for computational ethnomusicology (Section 5.2). Then, we give an introduction to traditional Georgian vocal music and explain the importance of Artem Erkomaishvili's recordings (Section 5.3). Subsequently, we provide detailed descriptions of the Erkomaishvili recordings, available transcriptions, and annotations (Section 5.4). Furthermore, we give an overview of the interactive web-based interface for accessing the dataset (Section 5.5) and show possible applications of the dataset for musicology and MIR research (Section 5.6). Finally, we conclude our findings and outline further musicological research on the Erkomaishvili dataset (Section 5.7).

5.2 Related Datasets and Tools

The goal of increasing the reproducibility and transparency of scientific results has led to the release of various open datasets and open-source software for computational musicology. In the following, we give a short summary on related datasets and tools that are fundamental to our work on the Erkomaishvili dataset. One of the most extensive databases for computational ethnomusicology has been collected within the CompMusic research project [188]. The collection comprises recordings of Indian Art music (Carnatic and Hindustani music) [193], Turkish-Makam [58, 186, 206], Jingju [74, 151], and Andalusian music [153]. The individual corpora, which include annotations of lyrics, scores, and editorial metadata, are hosted on the web-platform Dunya[41]. Kroher et al. [104] released the corpus COFLA, a dataset for the computational study of Flamenco music. The Dutch song database[42] hosts a research collection referred to as the "The Meertens Tune Collections", which is based on field recordings of Dutch folk songs [80]. The collection is accompanied with syllabified lyrics, key annotations, phrase annotations, and transcriptions [208, 211]. Furthermore, the *Polyphony Project*[43] hosts a collection of Ukrainian folk music recordings which is accessible via a web-based interface with multitrack audio and video players. A detailed overview of corpora for computational ethnomusicology can be found in [134].

As the number of public datasets increases, so does the number of open-source toolboxes for computational analysis of music recordings. Prominent examples are librosa [112], Essentia [20], MIR-Toolbox [106] and Marsyas [202]. Furthermore, tools such as Praat [19], Sonic Visualiser [29] and Tarsos [191] offer graphical user interfaces to compute and display analysis results. Recently, a collection of implementations, mathematical descriptions and explanations of music processing algorithms with emphasis on didactic aspects was released [122].

5.3 Traditional Georgian Vocal Music

Despite its small size, Georgia is home to diverse singing traditions, which form an essential part of its cultural identity. The disparity of polyphonic Georgian vocal music in comparison to Western music is—among other aspects—based on the abundant use of "dissonances" and on the fact that the music is not tuned to the 12-tone equal-tempered scale. While musicologists agree on the not equal-tempered nature of traditional Georgian vocal music, the peculiarities of the traditional Georgian tuning system are an ongoing topic of intense and controversial discussions [60, 173, 201]. A related aspect, which by some musicologists has been considered characteristic for Georgian singing, is the importance of harmonic intervals, which often goes along with a relaxed precision of melodic intervals, e.g., as discussed by [34, 176].

[41] https://dunya.compmusic.upf.edu/

[42] http://liederenbank.nl/

[43] https://www.polyphonyproject.com

Figure 5.2: Illustration of three-stage recording process.

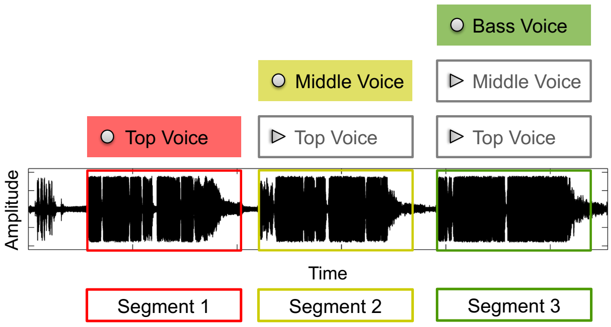


Table 5.1: Overview on Erkomaishvili's recordings.

# Songs	Total Duration (hh:mm:ss)	Mean / Min / Max Duration (mm:ss)
101	7:04:49	04:12 / 00:40 / 13:37

One key towards understanding these phenomena is the analysis of high-quality audio recordings. A recently released research corpus of traditional Georgian vocal music [180] meets all the quality criteria for computational analysis and allows for a systematic investigation of 216 performances (see Section 7.2 for a description of the dataset). However, with few exceptions, it only captures the current performance practice in Svaneti, a historic province in Georgia. Regarding historical field recordings, the known publicly available audio material is rather limited. This is true despite the fact that there have been considerable efforts to record traditional Georgian vocal music, starting with phonograph recordings more than 100 years ago. Unfortunately, many recordings from the early days of the last century have not survived the course of time. The audio data that have survived are mostly of insufficient quality for computational analysis. A notable exception are the 1966 tape recordings of Artem Erkomaishvili—one of the last Georgian master chanters—which are considered today as "original masterpieces of Georgian musical thinking" [190, p. XXVII]. A part of the recordings was manually remastered and published on CD [92]. Today, the recordings of Artem Erkomaishvili are very likely the oldest collection of Georgian chants of sufficient size and quality for computational studies.

5.4 Erkomaishvili Dataset

In this section, we describe the main components of the Erkomaishvili dataset. More specifically, we first explain the specific recording procedure and elaborate on existing transcriptions (Section 5.4.1). Then, we detail on the available manual annotations and the annotation process (Section 5.4.2). Finally, we present a semi-automatic method for the transfer of note onset annotations using alignment and interpolation techniques (Section 5.4.3).

5.4.1 Recordings and Transcriptions

In 1966, shortly before his death, Artem Erkomaishvili was asked to perform three-voice chants on his own by successively singing each of the individual voices. At the beginning of each recording, Artem Erkomaishvili announced the name of the song he was about to perform. After recording the top voice, one tape recorder was used to play back this first voice while a second tape recorder synchronously recorded the middle voice. Similarly, playing back the first and second voice, the bass voice was recorded, see Figure 5.2. In this way, Erkomaishvili accompanied his own recordings. However, due to this specific recording procedure, Artem Erkomaishvili usually began the middle and bass voices with a slight offset against the top voice. In summary, the Erkomaishvili recordings are not multitrack recordings in a strict sense (i.e., with isolated recordings for each voice). Only the top voice exists as an isolated (monophonic) recording. The resulting collection comprises 101 audio recordings with a total length of more than seven hours (see Table 5.1). Due to the distortions introduced by the tape recorders, the sound quality decreases with each recording stage. The strongest distortions typically occur in the third part, where it can sometimes be challenging to distinguish the bass voice from the other two voices. Additionally, since Artem Erkomaishvili was a bass singer, all songs are performed quite low. Considering the distortions and low-frequency content in the audio material, the recordings constitute a particularly challenging scenario for audio processing algorithms.

Transcriptions of Artem Erkomaishvili's recordings in Western staff notation have been published in the book "Georgian Church Hymns, Shemokmedi School" by David Shugliashvili [190]. The book contains 118 consecutively numbered transcriptions with song titles given in Georgian and English language, and song lyrics in Georgian and Latin letters. As opposed to Artem Erkomaishvili's performances, the transcriptions are notated in a higher register to account for a wider singer audience. During curation of the dataset, we used the score numbers in the book as unique file identifiers (Georgian Chant Hymns-IDs, abbr. GCH-IDs). As a naming convention, we included the GCH-IDs as three digit prefix consistently in all audio, sheet music, and annotation filenames. Since the publicly available audio collection comprises only 101 recordings, the Erkomaishvili dataset does not contain data for the following GCH-IDs: 021, 028, 037, 038, 039, 055, 064, 075, 082, 084, 096, 117, 118. Furthermore, recordings with the following GCH-IDs include two songs (second song in brackets): 022 (023), 043 (044), 058 (059), 102 (103)[44].

5.4.2 Manual Annotations

In this section, we explain all manual annotations contained in the Erkomaishvili dataset. From a previous study [126], we included recording structure annotations (Section 5.4.2.1) and semi-automatically annotated F0-trajectories of the three voices (Section 5.4.2.2). As one main contribution, we generated digital sheet music (Section 5.4.2.3) and onset annotations (Section 5.4.2.4) with the help of an experienced annotator. In the following, we use the song "Da Sulisatsa" (GCH-ID 087) as a running example.

[44] Further deviations are documented in the web-based interface (see Section 5.5).

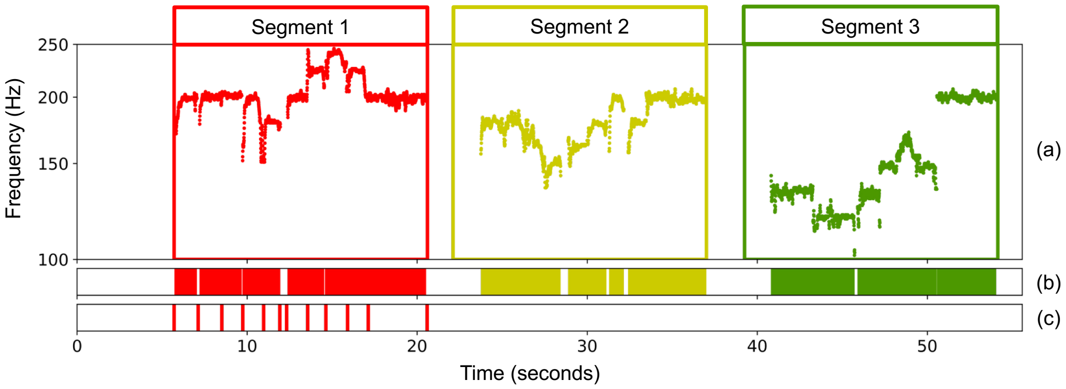


Figure 5.3: Illustration of available annotations for the song "Da Sulisatsa" (GCH-ID 087). **(a)** F0-trajectories within annotated segment boundaries plotted on a logarithmic frequency axis. **(b)** Activations of F0-trajectories. **(c)** Onset annotations including segment end.

5.4.2.1 Segment Annotations

As explained in Section 5.4.1, the first voice appears three times in every recording and marks the beginning and end of each recording stage. Due to varying tape velocities, the durations of second and third stages may slightly deviate from the duration of the first stage. However, for most of the recordings with few exceptions (GCH-ID 004, 015, 107), it is a good approximation to assume the same duration for all three stages. Following this assumption, Müller et al. [126] determined in all recordings the positions of three segments with equal duration (see Figure 5.2). Thereby, the segment start is defined by the start of each recording stage, whereas the segment duration is defined by the duration of the first stage. The segment annotations are available in CSV-format and contain six timestamps corresponding to the start and end positions of the three segments.

5.4.2.2 Fundamental Frequency Annotations

As part of the same study on Artem Erkomaishvili's recordings, Müller et al. [126] annotated F0-trajectories of the three voices for all 101 songs using a semi-automatic tool with a graphical user interface. The annotation procedure was as follows: first, the user specified temporal-spectral constraint regions in an enhanced time–frequency representation of the recording. Subsequently, F0-trajectories were automatically computed within the specified regions using an F0-estimation algorithm similar to Melodia [169]. In this way, the annotator could guide the estimation process. Additionally, the tool provides audiovisual feedback mechanisms for validation purposes and allows for correcting the computed F0-trajectories. The resulting annotated trajectories have a time resolution of 5.8 ms and a log-frequency resolution of 10 cents. The two-column annotation files in CSV-format contain equally-spaced timestamps in seconds in the first column and the F0-estimates in Hertz in the second column. The value of 0 Hz is used to indicate parts where the voice is inactive. Since the F0-trajectories of the three voices were annotated independently from the segment annotations, a few F0-values might be annotated outside the segment boundaries. The

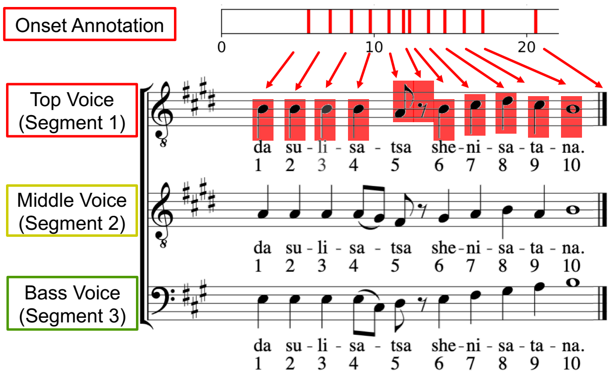


Figure 5.4: Digital score for “Da Sulisatsa” (GCH-ID 087). The annotated note and rest onsets for the top voice are highlighted in red. The QNRs are displayed underneath the lyrics of each voice.

F0-trajectories for our running example, plotted within the segment boundaries on a logarithmic frequency axis, are depicted in Figure 5.3a. The activations of the F0-trajectories are shown in Figure 5.3b.

5.4.2.3 Digital Sheet Music

Computational comparisons of the transcribed musical scores with the actual performances of Artem Erkomaishvili require digital scores in machine-readable format. One way to transfer printed sheet music to digital formats is to use Optical Music Recognition (OMR) systems [26, 149]. However, despite the advances over the last years, such systems are still error-prone and usually require labor-intensive manual corrections to obtain good quality results. Furthermore, most systems are not able to recognize characters from the Georgian writing system that are contained in the lyrics of the scores. Due to these circumstances, the transcriptions of David Shugliashvili were manually transferred to digital scores in MusicXML-format using the scorewriter programs Finale[45] and Sibelius[46]. As opposed to Western music, the traditional Georgian songs do not have a fixed musical time signature and are not organized using measures. However, a musical reference grid is beneficial for orientation within the scores. Furthermore, it helps to align the audio with the sheet music domain, as we will see in the following sections. Therefore, we introduce the concept of Quarter Note References (QNRs)—a concept of rather technical nature which has no further musical importance in traditional Georgian vocal music. A QNR is assigned to each note and indicates its position in terms of quarter notes from the beginning of the score. Following this concept, QNR 1 refers to the first note in the score, whereas QNR 2 refers to the note on the second quarter beat, which is not necessarily the second note in the score. In this way, QNRs are assigned to notes in every system of the score. In case the system contains shorter notes than quarter notes (e.g., eighth notes), QNRs can be floating point numbers indicating fractions of quarter notes. For visualization purposes, only integer QNRs have been added to the lyrics of the individual voices using the music21 Python toolkit [43]. The generated digital score with QNRs for our running example is depicted in Figure 5.4.

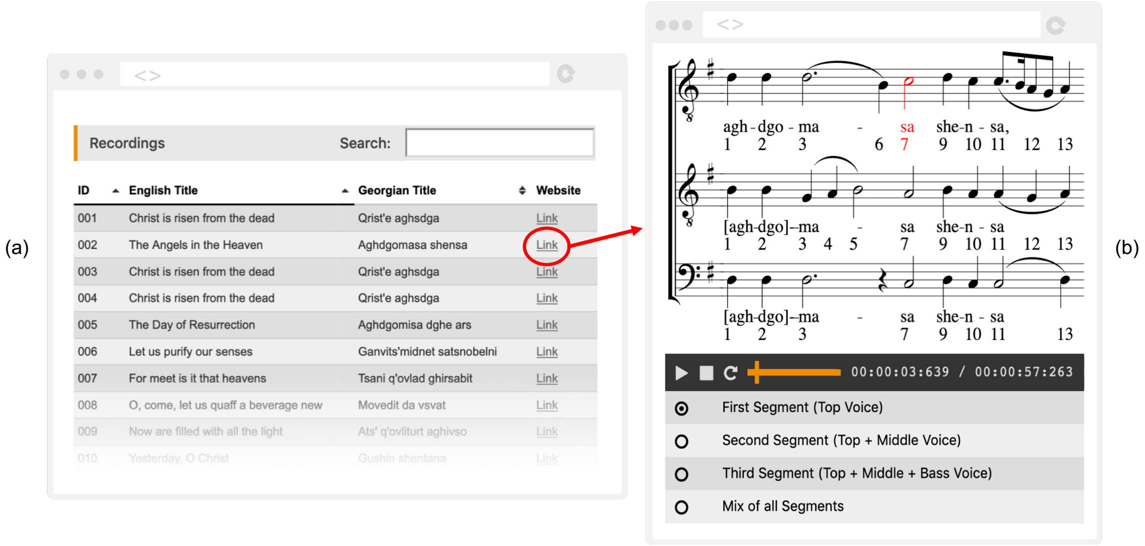
[45] https://www.klemm-music.de/makemusic/finale/
[46] https://www.avid.com/de/sibelius

5.4.2.4 Onset Annotations

In order to align the audio recordings with the digital scores, we manually annotated note and rest onset positions in the recordings using the open source software Sonic Visualiser [29]. Due to practical reasons, we only annotated note and rest onsets of the first voice. The onsets of the second and third voice were then derived from the onsets of the first voice and the segment annotations (for more details see Section 5.4.3). Figure 5.3c depicts the onset annotations of the first voice, which complement the existing manual segment and F0-annotations. Figure 5.4 shows the correspondences of the onset annotations to note events in the digital score for our running example. As a convention, the onset annotations include the end of the first voice (end of first segment) as a last timestamp.

5.4.3 Onset Computation

Generating onset annotations for the middle and bass voices in the Erkomaishvili recordings is challenging due to the polyphony and the poor audio quality in the second and third recording stages. In addition, Artem Erkomaishvili's low voice and the Georgian singing style with the abundant use of pitch slides in the beginning, end, and in between consecutively sung notes complicates this task. Therefore, instead of manually annotating the onsets of the middle and bass voices, we computed the onsets using a semi-automatic approach. As described earlier, due to the overdubbing recording process, the top voice is played back in the second and third segment and serves as reference for the other two voices. Using the segment annotations from Section 5.4.2.1, we mapped the onset annotations of the top voice to the other two segments by calculating the difference between the segment start positions and adding it to the top voice onset timestamps. In a subsequent step, we determined the onsets of the middle and bass voices using the previously introduced QNR grid. For notes of the middle and bass voices that share the same QNR as notes in the top voice (two notes that are exactly on the same score time), we assigned the mapped onset time of the top voice note. In order to obtain onsets of notes with a unique QNR (such as the notes between QNR 4 and QNR 5 in the middle and bass voices of our running example in Figure 5.4), we interpolated between the neighboring note onsets according to the QNR grid. We want to note that this approach requires the segments to be of equal duration and the tape velocity to stay constant during all recording stages in order to obtain a close approximation of the onsets for the second and third voices. Furthermore, the three voices are required to be sung in sync. These requirements can be assumed for most of the songs in the dataset. However, outliers can be found in the recordings with identifiers GCH-ID 004, 015, and 107. These recordings suffer from strong tape recorder artifacts. Therefore, it would be necessary to manually correct playback velocity and pitch prior to onset computation. In our Erkomaishvili dataset, we want to preserve the original recordings while indicating cases where outliers occur. We leave further modifications of the historic audio material to future studies. For all 101 recordings, the annotated onsets for the first segments and the computed onsets for the second and third segments are released in CSV format (one CSV file per segment). In the CSV files, each row contains information for one onset in the



Recordings Search:

ID	English Title	Georgian Title	Website
001	Christ is risen from the dead	Qrist'e aghsdga	Link
002	The Angels in the Heaven	Aghdgomasa shensa	Link
003	Christ is risen from the dead	Qrist'e aghsdga	Link
004	Christ is risen from the dead	Qrist'e aghsdga	Link
005	The Day of Resurrection	Aghdgomisa dghe ars	Link
006	Let us purify our senses	Ganvits'midnet satsnobelni	Link
007	For meet is it that heavens	Tsani q'ovlad ghirsabit	Link
008	O, come, let us quaff a beverage new	Movedit da vsvat	Link
009	Now are filled with all the light	Ats' q'ovliturt aghivso	Link
010	Yesterday, O Christ	Gushin shentana	Link

◉ First Segment (Top Voice)
○ Second Segment (Top + Middle Voice)
○ Third Segment (Top + Middle + Bass Voice)
○ Mix of all Segments

Figure 5.5: Web-based interface for accessing the Erkomaishvili dataset. **(a)** Main page with overview table. **(b)** Sub-page for the song "Aghdgomasa shensa" (GCH-ID 002) with score-following player.

following format: onset index, onset time in seconds relative to the segment start, onset "end" in seconds relative to the segment start (equivalent to the onset time of the next onset), QNR of the corresponding note or rest, QNR of the next note or rest.

5.5 Web-Based Interface

The public availability of MIR research corpora is essential for the reproducibility of scientific results, as well as for the preservation and dissemination of audio material and its annotations. Platforms such as Zenodo[47] offer to publicly share and distribute scientific data, while also providing citeable DOI. However, the interdisciplinary field of computational musicology requires platforms beyond data repositories, which support a cross-disciplinary scientific exchange by offering a direct, intuitive, and comprehensive access to the data. This can be accomplished by means of interactive interfaces that bridge the gap between the musicological and the audio domain, e.g., see [66, 91, 167].

As one main contribution, we developed a publicly accessible web-based interface[48] which hosts the full dataset. The interface provides download links to all segment, F0-, and onset annotations. Each song in the dataset has its individual sub-page, which is accessible through an interactive table with search and sorting functionalities as shown in Figure 5.5a. The central element of each sub-page is a multitrack audio player [216] with score-following functionality [221]. The displayed digital sheet music (given as an MEI file) is dynamically rendered in the web-browser with the help of Verovio [142]. The user can seamlessly switch between the three individual recording segments and a mix version of the three

[47] https://zenodo.org/

[48] https://www.audiolabs-erlangen.de/resources/MIR/2019-GeorgianMusic-Erkomaishvili

segments. In parallel, sung notes, lyrics, and QNRs are highlighted in the score according to the manually annotated onsets of the top voice and the automatically generated onset annotations for the middle and bass voices (see Figure 5.5b). In summary, beyond providing a non-technical and multimodal access to the Erkomaishvili dataset, the developed interface constitutes a first application scenario based on our annotations.

5.6 Applications for MIR and Musicology

The Erkomaishvili dataset can be used to address a wide range of research questions including technical as well as musicological ones. For example, a cappella vocal music is a challenging scenario for various MIR tasks such as F0-estimation [170], onset detection [17], and score-to-audio alignment [3, 127, 199]. In particular, the not equal-tempered nature of the Georgian songs and the characteristic pitch slides in traditional Georgian singing constitute challenging test scenarios for MIR algorithms. The Erkomaishvili dataset is one of few publicly available datasets on polyphonic a cappella singing [40, 180]. Due to the overdubbing procedure, the audio material provides a suitable scenario for studying source separation [31], audio segmentation [157], and audio restoration techniques [70].

Computational ethnomusicology is a rather young and still evolving field of research [73, 203, 205]. Its potential depends strongly on the existence of data collections which on the one hand are musically relevant, and on the other hand are of sufficient quality for the application of computational tools. The presented corpus meets both of these criteria. Its musicological relevance is undisputed. Ethnomusicologist John Graham, for example, writes: "Any theory must account for both the tuning system heard in the 1966 Erkomaishvili recordings and evidence from earlier singers and other regional chant systems seen in the transcription record" [78, p. 292]. Some musicologists even believe that only through the analysis of historical recordings (such as the Erkomaishvili collection), the Georgian musical system can be understood [60].

In the following, we illustrate the potential of our annotations in two case studies using the song "Gushin Shentana" (GCH-ID 010) as a running example. In the first case study (see Figure 5.6), we analyze the harmonic content of the Erkomaishvili recordings by computing distributions of sung harmonic intervals following the approach of Müller et al. [126]. To this end, the annotated F0-trajectories of the top, middle, and bass voices (see Figure 5.6a) are superimposed using the segment annotation (see Figure 5.6b). Then, for each time position, the intervals (given in cents) between the F0-trajectories of the top and middle voices, the top and bass voices, as well as the middle and bass voices are computed. Finally, integrating the occurrences of the different intervals over time, we obtain for each of the three cases an interval distribution (see Figure 5.6c). By computing and averaging such distributions over all 101 Erkomaishvili recordings, we obtain the distributions shown in Figure 5.6d. Besides the peak around 0 cents (unison), the accumulated distribution exhibits a prominent peak around 700 cents (fifth), which reflects the importance of the fifth interval in traditional Georgian vocal music. The peak at around 350 cents, located between the

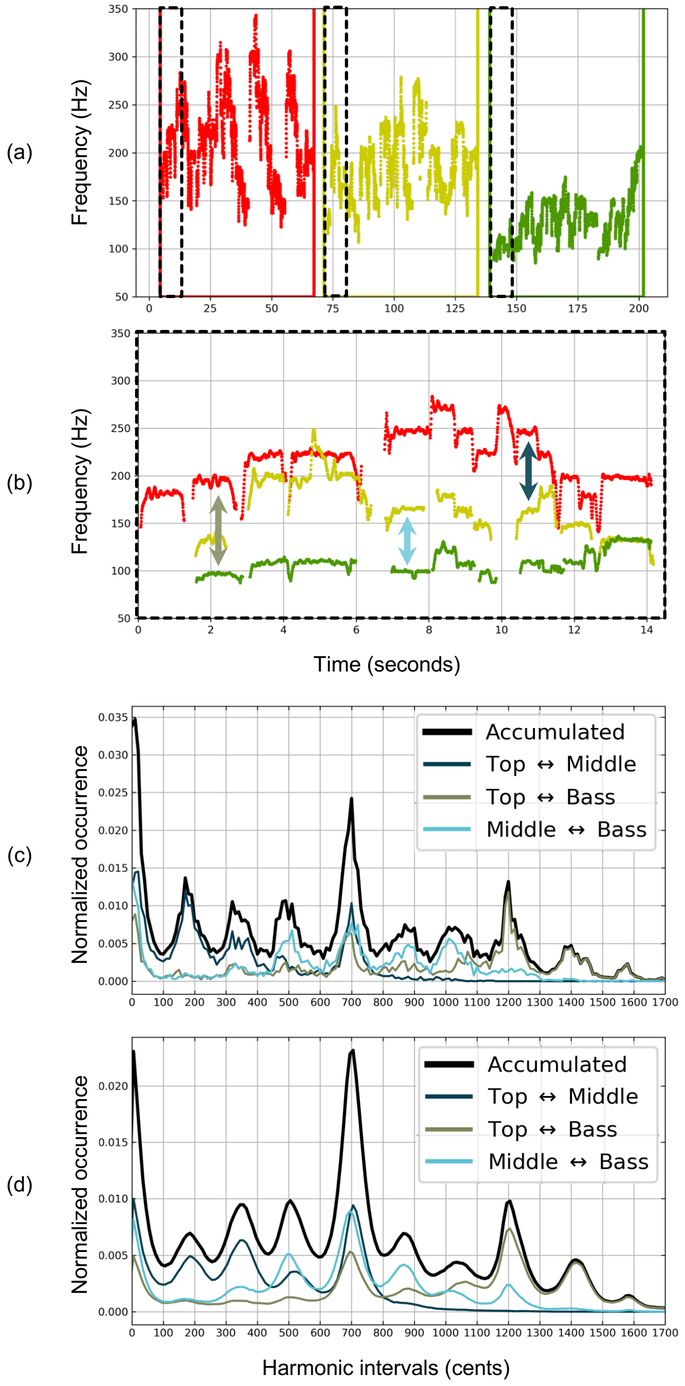


Figure 5.6: Computation of harmonic intervals. **(a)** F0-trajectories of “Gushin Shentana” (GCH-ID 010). **(b)** F0-trajectories of all three voices superimposed using segment annotation (zoom region). **(c)** Histogram of harmonic intervals for “Gushin Shentana” (GCH-ID 010). **(d)** Histogram of harmonic intervals averaged over all 101 songs of the dataset.

minor third (300 cents) and major third (400 cents), indicates the not equal-tempered nature of traditional Georgian vocal music. For a more detailed study on traditional Georgian tuning, we refer to [173].

What adds to the scientific value of the Erkomaishvili dataset is the availability of digital sheet music in Western staff notation for all songs (see Section 5.4.2.3). Although Western staff notation does not account for the not equal-tempered nature of traditional Georgian vocal music (see Section 5.3), the transcriptions can serve as a reference for more detailed studies on traditional Georgian tuning, e.g., as approximate guidance for MIR algorithms. Furthermore, qualitative comparisons with acoustical properties of Erkomaishvili's recorded performances give insights into the challenges of transcribing not equal-tempered music. This is illustrated in our second case study (see Figure 5.7).

In this study, we compare the pitch inventory as specified by the score representation with the pitch inventory as used by Artem Erkomaishvili. To this end, we proceed as follows. First, based on the digital sheet music (see Figure 5.7a), we generate a piano roll representation as shown in Figure 5.7b. Second, we extract stable regions in the F0-trajectories that roughly correspond to note events. For this task, we use an approach with morphological filters, which is described in detail in Section 6.3.2. Third, we temporally align the filtered F0-trajectories with the piano roll representation using the onset annotations. By making use of the previously introduced QNR concept (see Section 5.4.2.3), we obtain a QNR axis for both the score and the audio information. As a common frequency axis, we choose a logarithmic axis in cents (reference frequency 55 Hz). Fourth, we adapt the filtered and aligned F0-trajectories to the piano roll representation using a global pitch shift. This step is necessary since the transcriptions are notated in a higher pitch range than Artem Erkomaishvili's original performance (see Section 5.4.1). We determine the global pitch shift by computing the difference between the mean pitch of the piano roll representation and the mean pitch of the adapted trajectories (considering the trajectories of all voices jointly). In this way, we determine for our running example a pitch shift of 282 cents. The piano roll representation superimposed with the filtered and shifted trajectories is depicted in Figure 5.7c. In most of the cases, the extracted stable trajectory regions match the note events in the piano roll representation. However, a few regions in the F0-trajectories were not detected as "stable" (e.g., for the middle voice at QNR 10). In other cases (e.g., for the bass voice between QNR 11 and 13), the F0-values differ from the piano roll representation. To get an overall view on these deviations, we integrate the occurrences of pitch values of the piano roll representation and the adapted F0-trajectories over time. The two resulting distributions ("audio" and "score") are depicted in Figure 5.7d. In general, both distributions exhibit similar peak locations. However, there exist two peaks in the audio distribution that deviate substantially from the score distribution. The most significant deviation can be found between the pitches A♭3 and A3. Note that the audio distribution exhibits only one peak located between A♭3 and A3. A similar, but less salient deviation can be observed in the pitch range between D♭3 and D3.

In order to further investigate these discrepancies, we fit a Gaussian Mixture Model (GMM) with 13 Gaussians to the audio distribution from Figure 5.7d. The resulting mixture distribution is shown in Figure 5.8. The centers of the Gaussian pitch clusters are denoted with black numbers on top of the peaks, while the intervals between neighboring clusters are indicated with red numbers in between. The intervals between cluster centers from left to right are 192, 191, 152, 191, 173, 166, 155, 213, 178, 146, 179, and 196 cents. The numbers show that all intervals are all significantly larger than a semi-tone

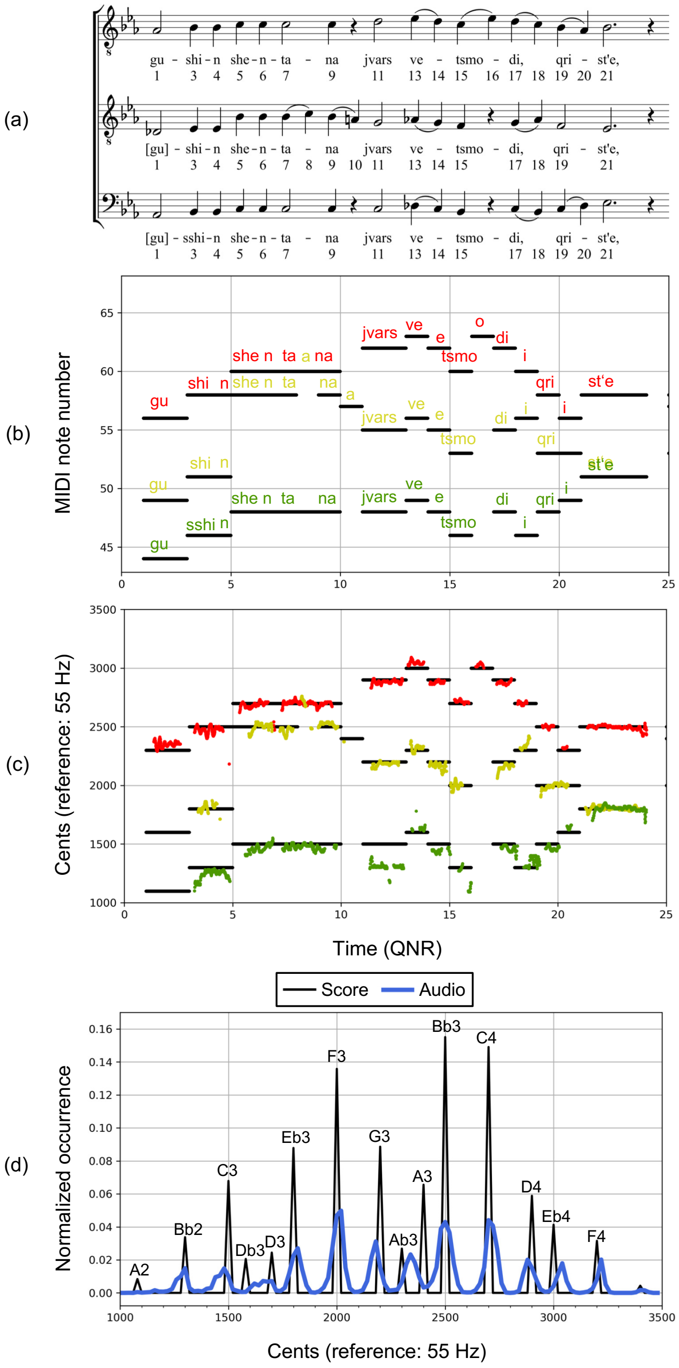


Figure 5.7: Comparison of transcribed score representation and annotated F0-trajectories for “Gushin Shentana” (GCH-ID 010). **(a)** Sheet music representation (excerpt). **(b)** Piano roll representation of score with lyrics (excerpt). **(c)** Adapted F0-trajectories for all three voices restricted to stable regions. **(d)** Pitch histograms for piano roll representation and adapted F0-annotation. The note names are given (A4 = 440 Hz).

(100 cents) and most of them are smaller than a whole tone (200 cents). From these results, we can draw two conclusions: first, the sung intervals indicate—once more—that Artem Erkomaishvili’s tuning is clearly not equal-tempered. Second, melodic steps between 100 and 200 cents can sometimes be perceived

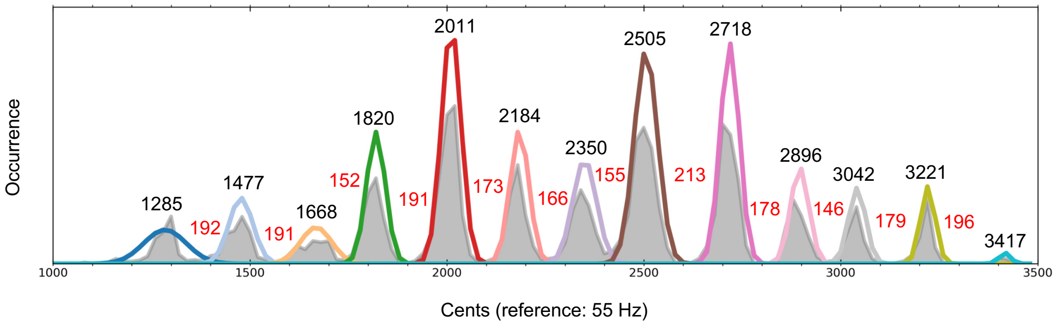


Figure 5.8: GMM with 13 Gaussians fitted to pitch histogram determined from adapted F0-trajectories of the song "Gushin Shentana" (GCH-ID 010). The black numbers on top of the peaks indicate the peak values in cents, while the red numbers indicate the intervals between neighboring peaks. The original distribution is indicated in grey color.

and transcribed as minor 2nd, sometimes as major 2nd. As a consequence, this can lead to effects in the transcription like in Figure 5.7a (QNR 10–14), where A, A♭, D, and D♭ appear closely together in time. From a Western (12-TET) perspective, this might seem counter intuitive. However, this is merely an effect of forcing a not equal-tempered tuning system into tempered Western staff notation. The task gets even more challenging for the transcriber if additional constraints by the harmonic context are imposed (e.g., the harmonic fifth between bass and middle voice at QNR 13).

In summary, these case studies show the potential of our annotations for studies on Artem Erkomaishvili's performances, as well as for analyzing tuning, pitch inventory, and musical scales underlying traditional Georgian vocal music.

5.7 Conclusions and Further Notes

In this chapter, we presented a carefully organized, manually annotated, and publicly available dataset of traditional Georgian vocal music. The corpus is based on historic recordings of the former master chanter Artem Erkomaishvili. As part of our work, we collated existing audio data and annotations. Furthermore, we generated onset annotations based on the digitized transcriptions by [190]. Finally, we developed an interactive web-based user interface with score-following audio players, which provides convenient access to the corpus data. Beside contributing to the preservation and dissemination of the rich Georgian musical heritage, this dataset is a versatile resource for MIR research and empowers musicological research on traditional Georgian vocal music.

An in-depth musicological analysis of the Erkomaishvili dataset can be found in [181]. In particular, ethnomusicologists have developed synoptic scale models of traditional Georgian singing considering harmonic and melodic aspects of the complete Erkomaishvili corpus. Furthermore, the study reveals that Artem Erkomaishvili may intentionally deviate from the melodic scale quite freely at one instance of time, while compensating for this deviation in the subsequent melodic steps. This observation suggests that

Artem Erkomaishvili actively follows a “deviation-compensation strategy,” which honors the scales but allows for melodic flexibility. In addition to the tangible results of our work, we believe that our study on the Erkomaishvili recordings has general implications for the determination of tuning models from audio data, particularly for “non-Western” music.

6 Filtering Approaches for Detecting Stable Regions in F0-Trajectories

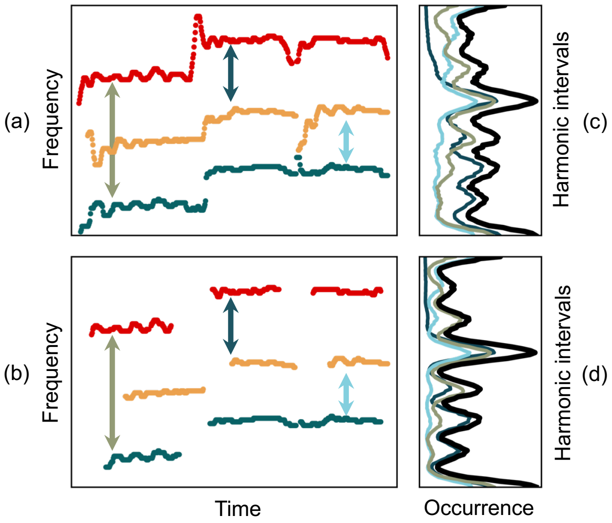
This chapter is based on [158]. The first author Sebastian Rosenzweig is the main contributor to this article. In collaboration with his supervisor Meinard Müller and Frank Scherbaum, he developed the ideas, formalized and implemented the approaches, conducted the experiments, and wrote the paper.

One major challenge in F0-based tonal analysis is introduced by unstable regions in the trajectories due to pitch slides and other frequency fluctuations. In this chapter, we describe two approaches for detecting stable regions in F0-trajectories: the first algorithm uses morphological operations inspired by image processing, and the second one is based on suitably defined binary time–frequency masks. To avoid undesired distortions in subsequent analysis steps, both approaches keep the original F0-values unmodified, while only removing F0-values in unstable trajectory regions. We evaluate both approaches against manually annotated stable regions and discuss their potential in the context of interval analysis for traditional three-part Georgian singing.

6.1 Introduction

Pitch slides are known to be part of vocal music across musical cultures [101, 103, 110, 176]. For example, as a stylistic element of traditional Georgian music, sung notes are often continuously connected, see Figure 6.1a. For tonal analysis based on extracted F0-trajectories, such stylistic elements constitute a major challenge. For example, when computing harmonic interval statistics (as illustrated by Figure 6.1c), one observes a blurring and a less salient peak structure in the resulting histograms. Thus, tonal analysis of traditional Georgian vocal music based on highly fluctuating and error-prone F0-trajectories is problematic. To alleviate such issues, contributions such as [110, 176] apply (semi-automatic) post-processing procedures to remove unstable regions in the trajectories and derive note-like events with a stable pitch. Note that for other scenarios (e. g., the tonal analysis of Hindustani Raga [165]), non-stable regions may contain musically important information.

Figure 6.1: Detection of stable regions in F0-trajectories for a three-part singing recording. **(a)** Original F0-trajectories. **(b)** F0-trajectories restricted to stable regions. **(c)** Harmonic interval histogram based on (a). **(d)** Sharpened harmonic interval histogram based on (b). The histograms in (c) and (d) were computed considering the entire Erkomaishvili corpus.



Motivated by such tonal analysis applications, we present in this chapter two automatic approaches that aim at identifying stable regions in F0-trajectories. Technically speaking, such regions correspond to horizontal structures (up to some tolerance) of trajectories. In acoustical and musical terms, such regions relate to pitched sounds where a singer has tuned into a harmonically stable pitch synchronized to other singers. In this context, our goal is to remove all frequency values in unstable regions, while keeping the original frequency values unmodified in the stable ones (see Figure 6.1b), resulting in a sharpened harmonic interval histogram (see Figure 6.1d). For accomplishing this task, we introduce two conceptually different approaches—one based on morphological operations and the other one based on binary masking. Furthermore, we evaluate both approaches against manually annotated stable regions and indicate their potential for interval analysis using the Erkomaishvili corpus (see Chapter 5) as example.

The remainder of this chapter is organized as follows. We discuss related work in Section 6.2, then give a mathematical description of our approaches in Section 6.3, and report on our experiments in Section 6.4. Finally, we summarize our findings and provide further notes in Section 6.5.

6.2 Related Work

In the following, we give an overview on work that is related to detecting stable regions in F0-trajectories. First, we want to note that stable region detection is not equivalent to F0-based transcription. In general, automated music transcription (AMT) aims at converting a music recording into some form of music notation [6, 7, 100]. In this process, many AMT systems apply temporal and spectral quantization of previously extracted F0-trajectories to derive pitches, onsets, and offsets of note events [24, 25, 53, 72, 103, 111, 115, 131, 168]. Rather than using quantized or modified F0-trajectories for our analysis, we aim at using trajectories restricted to stable regions (that may or may not correspond to note events) while leaving the original F0-values unmodified.

Detecting stable, transitional, and fluctuating patterns in F0-trajectories plays an important role for various tasks such as vibrato detection [35, 150, 219], singing style classification [135, 152], and motif detection [90, 148]. For example, in [217, 218, 219], the authors address the problem of detecting portamento (note transition) regions in Chinese string music. In [103], the authors identify stable regions as an important step towards transcribing recordings of Flamenco singing. In [118], the authors propose a vocal trajectory segmentation algorithm based on hysteresis defined on pitch–time curves. However, the underlying octave equivalence assumption may not be fulfilled in traditional Georgian vocal music. For a recent overview article of singing voice analysis, we refer to [88].

Furthermore, there are various studies on Indian Raga music, which are related to our work. In [65], a global pitch histogram ("pitch inventory") of the whole recording is computed. Then, informed by the histogram's peaks, stable regions are derived using empirically chosen thresholds for duration and fluctuation tolerance. In [101], the authors compute the local slope of the F0-trajectory and obtain stable regions by thresholding and quantization. However, due to the underlying scale assumptions, such approaches can not be directly applied to analyzing traditional Georgian singing, where pitch drifts may occur over the course of the song.

6.3 Stable Region Detection

In this section, we introduce our main technical contributions. We follow the notion of an F0-trajectory from Section 2.3.1. The experiments in this chapter use F0-trajectories with a time resolution of 5.8 msec per time index and a frequency resolution of 10 cents. Furthermore, we set $\omega_{\mathrm{ref}} = 55$ Hz. Figure 6.2a shows an F0-trajectory, which will serve as our running example in the remainder of this section. In the first two seconds, two notes are played on a piano without interruption. Subsequently, in the next two seconds, there are two sung notes smoothly connected by a pitch slide. Finally, the recording contains a note sung with vibrato.

The remainder of this section is structured as follows. To motivate the subsequent procedures, we explain a simple median-based filtering approach (Section 6.3.1). Then, we introduce two conceptually different approaches for determining stable regions in F0-trajectories—one based on morphological operations (Section 6.3.2) and the other one based on binary masking (Section 6.3.3).

6.3.1 Median Filtering

For tonal analysis based on F0-trajectories, one often applies some kind of filtering to remove outliers and other undesired pitch fluctuations [110, 198]. For example, by applying a median filter of odd length $L \in \mathbb{N}$, one obtains a smoothed trajectory η^{Median} defined by

$$\eta^{\mathrm{Median}}(n) := \mathrm{median}\left\{\eta(n - \tfrac{L-1}{2} : n + \tfrac{L-1}{2})\right\} \tag{6.1}$$

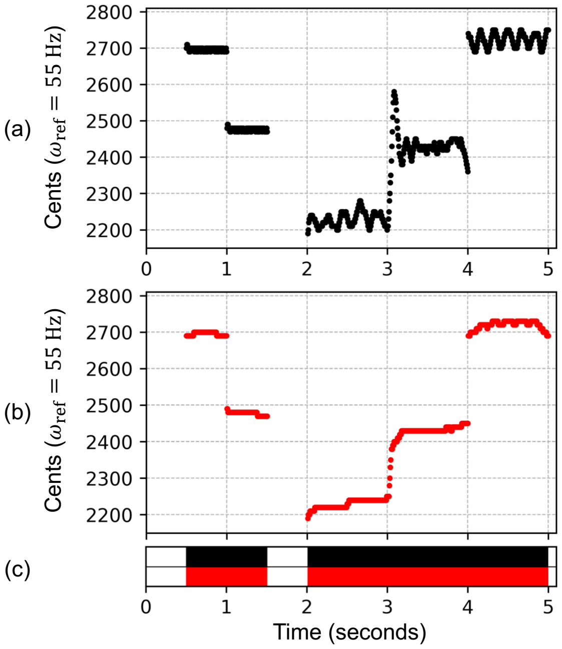


Figure 6.2: Effect of median filtering. **(a)** Original trajectory η. **(b)** Median-filtered trajectory η^{Median}. **(c)** Activation regions of η (black) and η^{Median} (red).

for $n \in \mathbb{Z}$. In this definition, the symbol $*$ is handled as $-\infty$. Figure 6.2b shows η^{Median} of our running example using $L = 69$ (corresponding to 0.4 sec). This example shows how median filtering introduces smoothing while removing outliers (such as the peak around the third second). However, the non-stable transition between the two sung notes remains after filtering. This is not what we aim at. First, we do not want to change frequency values in stable regions (with the goal not to introduce smoothing effects in subsequent tonal analysis steps). Second, we aim at explicitly detecting unstable regions, which can then be removed from the F0-trajectory. In the following, we present two conceptually different approaches that fulfill these requirements.

6.3.2 Morphological Approach

The first approach, which is inspired by work of Vávra et al. [209], uses morphological operations as known in image processing. Applying these operators to F0-trajectories, dilation corresponds to max filtering, and erosion to min filtering. Given a trajectory η, this results in a dilated trajectory η^L_{max} and an eroded trajectory η^L_{min} defined by

$$\eta^L_{\text{max}}(n) := \max\left\{\eta(n - \tfrac{L-1}{2} : n + \tfrac{L-1}{2})\right\}, \tag{6.2a}$$

$$\eta^L_{\text{min}}(n) := \min\left\{\eta(n - \tfrac{L-1}{2} : n + \tfrac{L-1}{2})\right\}, \tag{6.2b}$$

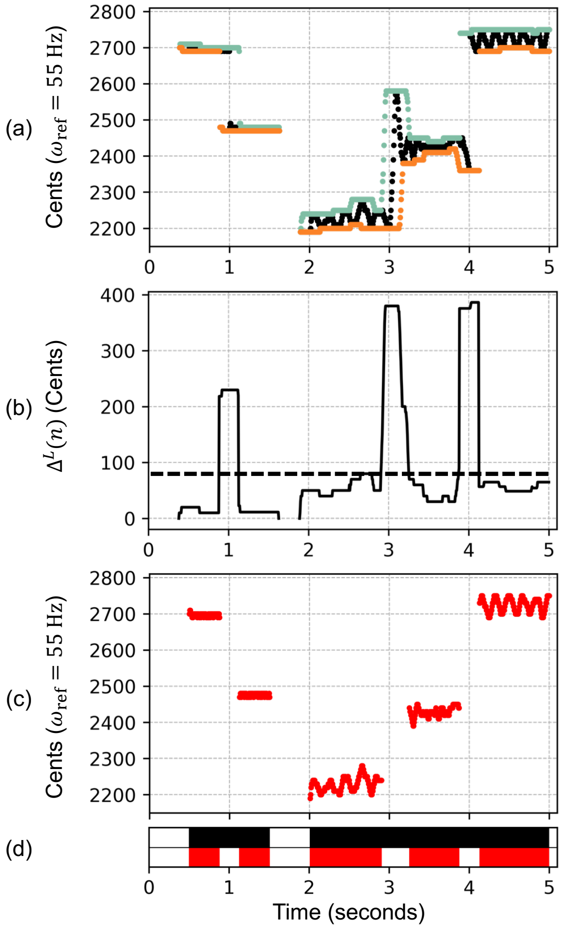


Figure 6.3: Morphological approach for detecting stable regions. **(a)** F0-trajectories η (black), $\eta^L_{\max}$ (green), and $\eta^L_{\min}$ (orange). **(b)** Morphological gradient Δ^L with threshold $\tau = 90$. **(c)** Trajectory η^{Morph} restricted to stable regions. **(d)** Activation regions for η (black) and η^{Morph} (red).

for $n \in \mathbb{Z}$, where $L \in \mathbb{N}$ is assumed to be an odd integer. In max filtering, the symbol $*$ is handled as $-\infty$, whereas in min filtering it is handled as $+\infty$. Figure 6.3a shows the resulting trajectories $\eta^L_{\max}$ and $\eta^L_{\min}$ for our running example using $L = 43$ (corresponding to 0.25 sec). In a next step, we define the difference Δ^L between the dilated and eroded trajectories, also termed morphological gradient [154]:

$$\Delta^L(n) := \eta^L_{\max}(n) - \eta^L_{\min}(n) \tag{6.3}$$

for $n \in \mathbb{Z}$, where we set $\Delta^L(n) = *$ whenever $\eta^L_{\max}(n)$ or $\eta^L_{\min}(n)$ are not defined. As shown in Figure 6.3b, the difference Δ^L is large in non-stable parts (e. g., around the third second), whereas it is small in stable parts (e. g., within each of the piano notes). Fixing a suitable threshold $\tau > 0$ (given in cents), we define the trajectory η^{Morph} by setting

$$\eta^{\mathrm{Morph}}(n) := \begin{cases} \eta(n), & \text{for } |\Delta^{L}(n)| \leq \tau, \\ *, & \text{otherwise.} \end{cases} \tag{6.4}$$

The threshold τ can be seen as a tolerance parameter that specifies the maximally allowed fluctuation under which a trajectory is still considered to be stable. The resulting trajectory η^{Morph} for our running example is depicted in Figure 6.3c using a threshold of $\tau = 90$ cents. As shown in Figure 6.3d, the morphological approach succeeds in identifying stable regions. However, it also introduces a truncation at both sides of sudden jumps (e. g., around the first and fourth second) by half the filter length $(L-1)/2$. In the next section, we show how this truncation effect can be reduced by applying a 2D-masking approach involving some median filtering. Finally, we want to note that considering the morphological gradient is conceptionally similar to the approach based on Gaussian derivate filtering as described in [103]. In our approach, the threshold parameter τ can be adjusted dynamically to account for characteristics of individual trajectories, e. g. by considering the p-quantile of the morphological gradient Δ^{L}.

6.3.3 Masking Approach

We now introduce an alternative approach for detecting stable trajectory regions, which works in the 2D-domain. In a first step, we encode a trajectory η as a binary 2D-representation $\Gamma_R : \mathbb{Z} \times \mathbb{Z} \to \{0,1\}$. Given a frequency resolution of $R \in \mathbb{R}$ (given in cents), Γ_R is defined by

$$\Gamma_R(n,b) := \begin{cases} 1, & \text{for } \left\lfloor \frac{\eta(n)}{R} + 0.5 \right\rfloor = b, \\ 0, & \text{otherwise,} \end{cases} \tag{6.5}$$

with time index $n \in \mathbb{Z}$ and frequency bin index $b \in \mathbb{Z}$ (corresponding to a logarithmic frequency axis). Figure 6.4a shows the binary representation Γ_R using $R = 10$ cents for our running example. In the second step, we introduce some tolerance in frequency direction by vertically applying a max-filtering using a filter length parameter $\beta \in \mathbb{N}_0$ (specified in bins). This results in the representation Γ_R^{β} defined by

$$\Gamma_R^{\beta}(n,b) := \max\{\Gamma_R(n, b-\beta : b+\beta)\}. \tag{6.6}$$

This operation is illustrated by Figure 6.4b using $\beta = 5$ (leading to a frequency width of $2\beta + 1 = 11$ bins corresponding to 110 cents). In a third step, inspired by an algorithm for Harmonic–Percussive Source Separation [62], a median filter of odd length $L \in \mathbb{N}$ is applied in horizontal direction yielding a representation $\Gamma_R^{\beta,L}$:

$$\Gamma_R^{\beta,L}(n,b) := \mathrm{median}\left\{\Gamma_R^{\beta}(n - \tfrac{L-1}{2} : n + \tfrac{L-1}{2}, b)\right\}. \tag{6.7}$$

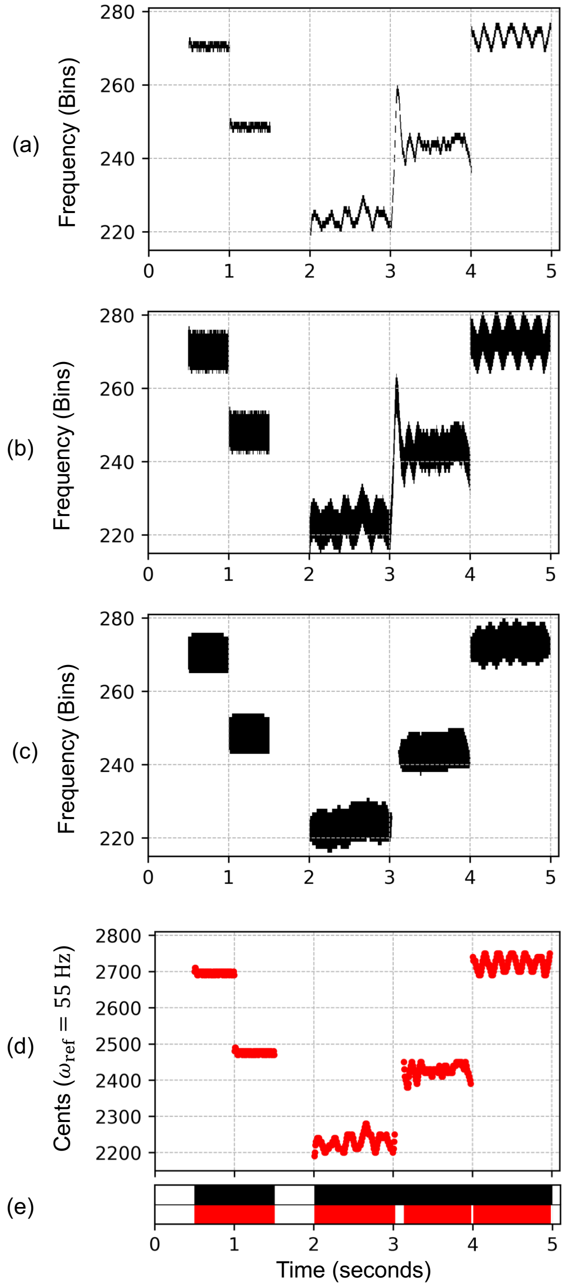


Figure 6.4: Masking approach for detecting stable regions. **(a)** Binary representation Γ_R. **(b)** Max-filtered representation Γ_R^β. **(c)** Median-filtered binary mask $\Gamma_R^{\beta,L}$. **(d)** Trajectory η^{Mask} restricted to stable regions. **(e)** Activation regions for η (black) and η^{Mask} (red).

Applying horizontal median filtering suppresses vertical structures (e. g., pitch slides), while enhancing horizontal structures (corresponding to stable regions), see Figure 6.4c for an illustration when using

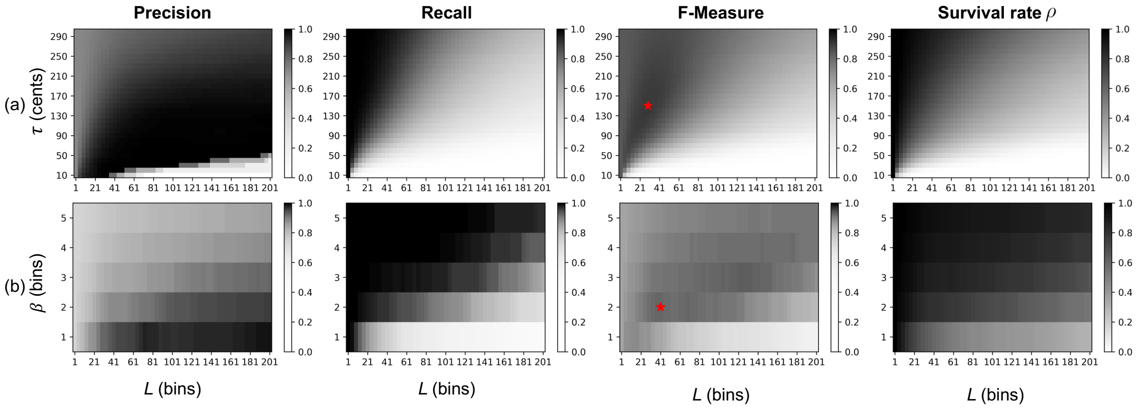


Figure 6.5: Precision, recall, F-Measure, and survival rate ρ of parameter sweeps averaged over five recordings (see Table 6.1). The parameter settings chosen for subsequent experiments are marked with red stars. **(a)** Morphological approach. **(b)** Masking approach.

$L = 43$ (corresponding to 0.25 sec). In the fourth step, the output trajectory η^{Mask} is obtained by setting

$$\eta^{\text{Mask}}(n) := \begin{cases} \eta(n), & \text{if } \Gamma_R^{\beta,L}(n,b) = 1, \\ *, & \text{otherwise,} \end{cases} \tag{6.8}$$

with $b = \lfloor \eta(n)/R + 0.5 \rfloor$. This last step can be thought of as "masking" the input trajectory η using the binary mask $\Gamma_R^{\beta,L}$. Figure 6.4d shows the resulting trajectory η^{Mask} for our running example. Note that, even though the masking procedure involves some quantization parameter R, the final trajectory η^{Mask} coincides with the original trajectory η in stable regions. Similar to the parameter τ for computing η^{Morph}, the parameter β controls the frequency tolerance within stable regions for η^{Mask}. As also indicated by our running example, the truncation effects at sudden jumps introduced by the morphological approach have been eliminated by our masking approach (compare η^{Morph} and η^{Mask} around the first and fourth second). While the 2D-masking approach is computationally more expensive than the 1D-morphological approach, it allows for processing multiple (non-overlapping) trajectories at the same time. Furthermore, one may account for weighted trajectories (e. g., trajectories with assigned amplitude or confidence values) by using real-valued instead of binary masks. Note that both algorithms do not enforce continuity of output trajectories. In particular, strict parameter settings (e. g. small τ and small β) may result in fluctuating sound events (e. g. a note sung with strong vibrato) being split up into several disconnected regions.

6.4 Evaluation

In this section, we report on experiments that indicate the role of the parameters and the behavior of the morphological and the masking approach. In Section 6.4.1, we numerically compare both approaches using a set of manually annotated stable regions in F0-trajectories from the publicly available Erkomaishvili corpus (see Chapter 5). Using suitable parameter settings, we then apply both algorithms to the trajectories

Table 6.1: Precision (P), recall (R), F-Measure (F), and survival rate (ρ) evaluated on the basis of manually annotated F0-trajectories for five Erkomaishvili recordings.

ID	η^{Ann}	η^{Morph}				η^{Mask}			
	ρ	P	R	F	ρ	P	R	F	ρ
001	61%	0.82	0.94	0.88	70%	0.82	0.94	0.88	71%
002	79%	0.94	0.85	0.89	72%	0.93	0.87	0.90	74%
010	68%	0.87	0.92	0.89	72%	0.84	0.95	0.89	77%
087	78%	0.88	0.98	0.93	87%	0.87	0.98	0.92	88%
110	74%	0.90	0.96	0.93	79%	0.88	0.97	0.92	80%

of all 101 recordings in the corpus (see Section 6.4.2). We hypothesize that a consistent detection of stable regions using the two conceptually different approaches is a good indicator that the results are musically meaningful. Finally, in Section 6.4.3, we demonstrate the potential of our approaches for enhancing harmonic interval distributions.

6.4.1 Evaluation Measures and Parameters

In order to compare the algorithms' performance, we annotated stable regions of F0-trajectories extracted from five representative Erkomaishvili recordings. To this end, we used an interactive interface described in [126] to manually remove all unstable trajectory regions that correspond to note transitions and other artifacts. As evaluation metrics, we use standard precision (P), recall (R) and F-measure (F) computed frame-wise on the basis of the trajectories' activations. First, all frames with no specified frequency value in the original trajectory ($\eta(n) = *$) are left unconsidered. Frames classified as stable by our approaches are counted as *true positives* (TP) if they agree with frames annotated as stable, otherwise they are counted as *false positives* (FP). Furthermore, frames annotated as stable are counted as *false negatives* (FN), if they are classified as unstable. Then,

$$\mathrm{P} := \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad \mathrm{R} := \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \quad \mathrm{F} := \frac{2 \cdot \mathrm{P} \cdot \mathrm{R}}{\mathrm{P}+\mathrm{R}}. \tag{6.9}$$

Note that $\mathrm{P} := 0$ for $\mathrm{TP} + \mathrm{FP} = 0$, $\mathrm{R} := 0$ for $\mathrm{TP} + \mathrm{FN} = 0$, and $\mathrm{F} := 0$ for $\mathrm{P} + \mathrm{R} = 0$. Furthermore, we introduce an evaluation measure referred to as *survival rate* and denoted as ρ. This measure, which indicates the percentage of remaining trajectory values after filtering, is defined as follows:

$$\rho := \frac{|\{n : \eta^{\text{Stable}}(n) \neq *\}|}{|\{n : \eta(n) \neq *\}|} \cdot 100, \tag{6.10}$$

with $\eta^{\text{Stable}} = \eta^{\text{Morph}}$ for the morphological approach, $\eta^{\text{Stable}} = \eta^{\text{Mask}}$ for the masking approach and $\eta^{\text{Stable}} = \eta^{\text{Ann}}$ for an annotated trajectory η^{Ann}.

In order to analyze the algorithms' behavior for different parameter settings, we conduct parameter sweeps over L, τ, and β, using a fixed frequency resolution of $R = 10$ cents. For each evaluation metric, we construct a matrix with each entry corresponding to a metric's value for a specific parameter setting averaged over the five annotated recordings. The resulting matrices for precision, recall, F-measure, and

Table 6.2: Evaluation of the masking approach against the morphological approach considering the trajectories of all 101 recordings of the Erkomaishvili corpus (with fixed parameter settings from Section 6.4.1). The mean μ and standard deviation σ refer to statistics taken over the corpus.

	P	R	F	$\rho\ (\eta^{\mathrm{Morph}})$	$\rho\ (\eta^{\mathrm{Mask}})$
μ	0.89	0.94	0.92	73%	77%
σ	0.02	0.01	0.02	5%	5%

survival rate are depicted in Figure 6.5a for the morphological approach and in Figure 6.5b for the masking approach. The visualizations show that τ and β play a similar role: high values of τ and β make the approaches more tolerant to local frequency fluctuations in the trajectories, thus increasing the survival rates. In contrast, when decreasing τ and β, less values remain in the filtered trajectories, leading to lower survival rates. Furthermore, note that increasing the filter length L leads to an increase in precision and a decrease in recall for both approaches. In the case of the morphological approach, very large filter lengths lead to a survival rate of $\rho = 0$ (nothing is remaining), which also leads to a precision of zero.

For our further experiments, we use fixed parameter settings for both approaches that correspond to maxima in the F-measure matrices (see red stars in Figure 6.5). The morphological approach reaches a maximum F-measure of 0.90 for $\tau = 150$ cents and $L = 29$ bins, whereas the masking approach reaches a maximum F-measure of 0.90 for $\beta = 2$ bins and $L = 41$ bins. Using these parameter settings, the evaluation results for our five annotated examples (IDs correspond to songs on the publicly available website[49]) are given in Table 6.1. From the table, we can see that both approaches are able to detect stable regions in all five examples. We want to note that the optimal parameter settings vary from song to song, depending on the occurring note durations, characteristics of pitch slides, and other performance aspects. As an alternative to a fixed setting, one may chose the parameters in a song-dependent way, e. g., by fixing the survival rate. In summary, our experiments on the Erkomaishvili corpus showed that the specific choice of parameters is not crucial within a certain range (see also the F-measure matrices of Figure 6.5).

6.4.2 Consistency

The two approaches for detecting stable regions in trajectories are conceptually different. Nevertheless, in the case of the five annotated recordings, both approaches worked successfully and performed in a similar fashion. Based on the hypothesis that a consistent performance of both approaches is a necessary condition for obtaining meaningful results, we applied both approaches independently to all 101 recordings of the Erkomaishvili corpus. We then compared the results by evaluating the trajectories obtained by the masking approach against the trajectories obtained by the morphological approach using the evaluation metrics defined in Section 6.4.1. The mean μ and standard deviation σ (taken over the corpus) of the evaluation results are shown in Table 6.2. The numbers indicate that both approaches deliver similar results on average with a small standard deviation. Furthermore, both approaches roughly exhibit the same average survival rate for the chosen parameter settings. Beyond these overall measures, we also looked at recordings where the two approaches delivered less consistent results. A manual inspection revealed that

[49] `https://www.audiolabs-erlangen.de/resources/MIR/2017-GeorgianMusic-Erkomaishvili`

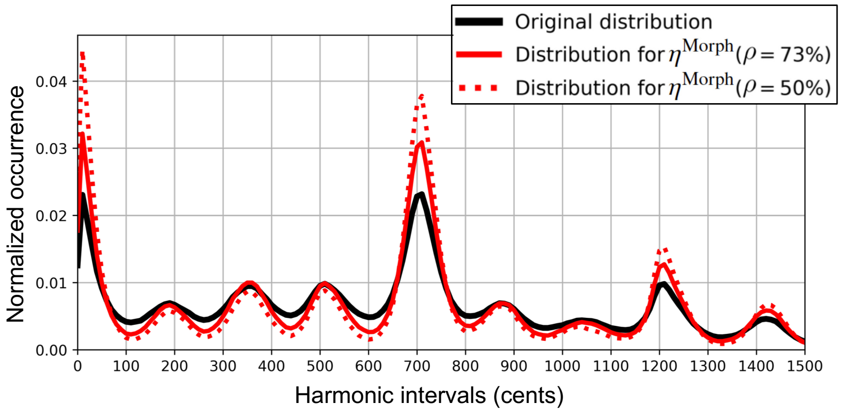


Figure 6.6: Harmonic interval distributions obtained from the entire Erkomaishvili corpus.

these recordings often contain speech-like passages (rather than singing) and extremely short notes such as in the songs with ID 022 and ID 074. Results for all 101 recordings are publicly available through audio–visual interfaces.[50]

6.4.3 Harmonic Interval Analysis

In the following, we want to demonstrate the potential of the presented approaches for interval analysis of Georgian vocal music by computing harmonic interval size distributions from the filtered trajectories (cf. Chapter 5.6, [126, 176]). To this end, we superimpose the filtered trajectories of lead, middle and bass voice and determine the frame-wise intervals for each voice pair (as indicated in Figure 6.1). Then, by accumulating the occurrences of the different intervals over time, we obtain interval histograms. These histograms are normalized (using the ℓ^1-norm) to obtain distributions. Figure 6.6 shows three such distributions obtained by considering all 101 recordings of the Erkomaishvili corpus. The first distribution (black solid line) is based on the original F0-trajectories. The second distribution (solid red line) is obtained by considering only stable regions after morphological filtering. Here, we use the parameter settings discussed in Section 6.4.1. Filtering with the masking approach leads to similar distributions. Note that the filtering leads to a sharper interval distribution emphasizing the peaks at the harmonically relevant intervals while not changing the respective peak locations. Using stricter parameter settings leads to a further sharpening (see red doted line in Figure 6.6). However, overdoing the filtering may drastically reduce the survival rate. This, in turn, may lead to a distortion or even a loss of peak structures corresponding to relevant harmonic intervals.

6.5 Conclusions and Further Notes

In this chapter, we presented two conceptually different approaches for detecting stable regions in F0-trajectories, which perform equally well with respect to a set of manually annotated trajectories. Rather

[50] `https://www.audiolabs-erlangen.de/resources/MIR/2019-ISMIR-StableF0`

than advocating a specific parameter setting, our goal was to introduce these concepts in a mathematical rigorous way while highlighting their potential using the Erkomaishvili corpus as example scenario. In Chapter 7, we will show that stable regions can also be exploited as an indiactor for reliability of F0-estimates when considering multiple, automatically estimated F0-trajectories. In Chapter 8, we will expand our morphological approach for detecting note-like objects in F0-trajectories and show how this approach can be applied in an interactive fashion for tonal analysis (see also Figure 8.4).

7 A Fusion Approach for Reliability Assessment of F0-Estimates

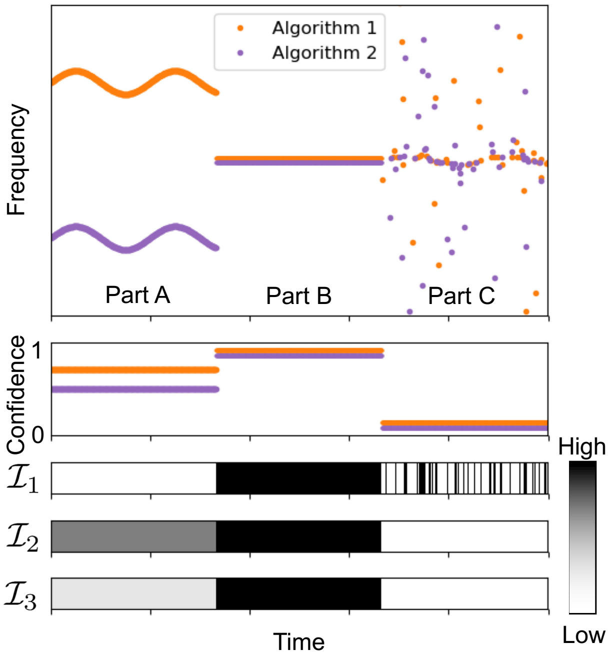
This chapter is based on [162]. The first author Sebastian Rosenzweig is the main contributor to this article. In collaboration with his supervisor Meinard Müller and Frank Scherbaum, he developed the ideas, formalized the approach, and wrote the paper. Furthermore, Sebastian Rosenzweig implemented the approach and conducted the experiments.

Over the last decades, various conceptually different approaches for F0-estimation in monophonic audio recordings have been developed. The algorithms' performances vary depending on the acoustical and musical properties of the input audio signal. A common strategy to assess the reliability (correctness) of an estimated F0-trajectory is to evaluate against an annotated reference. However, such annotations may not be available for a particular audio collection and are typically labor-intensive to generate. In this chapter, we consider an approach to automatically assess the reliability of F0-trajectories estimated from monophonic singing voice recordings. As main contribution, we propose three reliability indicators that are based on the outputs of several algorithms. Besides providing a mathematical description of the indicators, we analyze the indicators' behavior using a set of annotated vocal F0-trajectories. Furthermore, we show the potential of the proposed indicators for exploring unlabeled audio collections on the example of field recordings of traditional Georgian vocal music.

7.1 Introduction

F0-estimates often serve as mid-level representation [14, 37] in MIR tasks such as automatic music transcription [7] and performance analysis [46, 50]. There exist a variety of approaches for monophonic F0-estimation, ranging from model-based methods [27, 48, 109] to more recent deep learning-based methods [68, 97]. A monophonic F0-estimation algorithm typically outputs one F0-value per time instance together with a confidence value that indicates the algorithm's certainty whether the sound source is active or not (sometimes referred to as "voicing"). However, high confidence does not necessarily imply high reliability (correctness) of an estimated F0. For example, typical estimation errors are confusions of the

Figure 7.1: Illustration of reliability indicators for an artificial example. $\mathcal{I}_1$: F0-agreement. $\mathcal{I}_2$: Overall confidence. $\mathcal{I}_3$: F0-trajectory stability.



F0 with higher or lower harmonics (in particular octaves). The performance of a specific F0-estimation algorithm depends on the audio signal's acoustic properties (e.g., microphone characteristics, recording conditions) and musical properties (e.g., instrumentation, singing/playing styles).

In order to assess the accuracy of F0-estimates, a commonly used strategy is to evaluate an algorithm's output against a manually annotated reference, e.g., using the standard metrics defined in [139, 170] or a recently proposed variant [10]. However, manual F0-annotations are labor-intensive to generate and sometimes not available. This motivates the need for automatic approaches that deliver cues on the reliability of F0-estimates. In prior work [49], the authors have suggested a deep-learning-based approach for reliability assessment of F0-estimates from speech recordings. The approach requires access to the algorithms' internal computations, as well as algorithm-specific adaptation and training.

In the following, we develop a more generic approach that is independent of the algorithms' working principle and available implementations. Conceptually similar to the studies in [4, 22], our approach makes use of F0- and confidence outputs of several algorithms. As one main contribution, we introduce three reliability indicators (denoted as $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$) that measure the reliability of an F0-estimate with respect to three different criteria. The working principle is illustrated in Figure 7.1 using an artificial example with two algorithms. $\mathcal{I}_1$ measures the agreement of the algorithms' F0-estimates. In our artificial example, it indicates low agreement in Part A and high agreement in Part B (and for some time instances in Part C). $\mathcal{I}_2$ measures the overall confidence of the algorithms. In our example, $\mathcal{I}_2$ indicates medium to high confidences in Parts A and B, and low confidences in Part C. $\mathcal{I}_3$ measures the stability of the estimated F0-trajectories in a temporal context. This criterion is based on the observation that some algorithms tend to output random-like values in parts where no singing voice is active. Furthermore, in parts where

Figure 7.2: Impressions of the GVM collection. (Reprinted by kind permission of Frank Scherbaum, Potsdam University).

F0-estimation is ambiguous or problematic (e.g., for consonants), estimated F0-trajectories often exhibit abrupt jumps. In our artificial example, $\mathcal{I}_3$ indicates low stability for Part A (due to vibrato) and Part C (due to noise), and high stability for Part B. As a test scenario for our indicators, we consider a collection of multitrack field recordings of polyphonic Georgian vocal music (GVM), also referred to as the GVM collection [178, 180], which will be introduced in Section 7.2. Subsequently, we provide mathematical definitions of the reliability indicators in Section 7.3 and evaluate the indicators' performance on a set of manual F0-annotations extracted from selected songs of the GVM collection in Section 7.4. Finally, we indicate the potential of the proposed indicators for exploring unlabeled audio collections in Section 7.5 and summarize this chapter in Section 7.6.

7.2 GVM Collection

Traditional Georgian vocal music has been an active field of ethnomusicological research since more than 100 years. Besides the musically invaluable recordings of Artem Erkomaishvili (see Chapter 5), the availability of high quality audio recordings of this orally transmitted music is still limited [180]. During the summer of 2016, Frank Scherbaum (in collaboration with Nana Mzhavanadze), performed a three-month field expedition in Georgia with focus on the region Svaneti. Musicologists believe that Svaneti is home to the first stages of Georgian vocal music development.[51] During the expedition, the

[51] https://www.uni-potsdam.de/de/soundscapelab/about-seismosoundscape-lab/people/frank-scherbaum/my-personal-travelogue

Table 7.1: Overview of the multitrack audio recordings in the GVM collection.

Microphone	# Tracks	Duration (hh:mm:ss)
LRX	548	16:52:07
HDS	477	15:21:55
Room	203	06:08:51
Total	1228	38:22:54

ethnomusicologists recorded a new research corpus of traditional Georgian vocal music, including singing, praying, and lamenting, also referred to as the GVM collection.

The GVM collection comprises in total 216 performances, among which there are 37 performances of prayers and 11 performances of funeral songs (see Chapter 8 for a case study on these recordings). The rest are performances of a diverse range of song types, including hymns, ballads, dance songs, and table songs [180]. In total 85 performances stem from the region Svaneti. Figure 7.2 provides an impression of the diversity of the recorded ensembles.

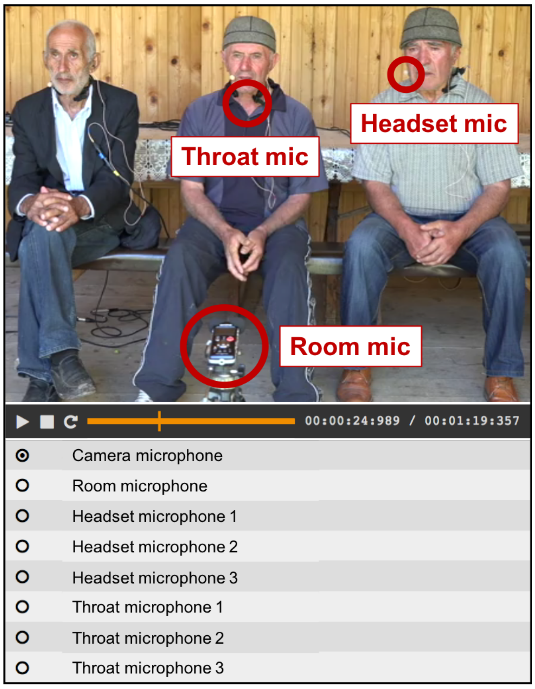
What adds to the uniqueness of the GVM collection is the systematic use of close-up microphones (in particular, headset and throat microphones, see Section 2.2), room microphones, and video cameras for recording the singers. In the following, we will refer to headset and throat microphones as HDS and LRX, respectively. An overview about the multitrack recordings included in the GVM collection is provided in Table 7.1, which shows the outstanding amount of recordings in the collection. To enable an easy and direct access to the collection, we created a interactive web-based interface[52], which hosts the data. The start page of the interface includes an interactive table of all performances, listing their unique GVM-IDs, song and ensemble names, available multitrack data, and recording dates. For each performance, the interface provides a multimedia player based on the library *trackswitch.js* [216], which allows for seamless switching and mixing of the individual tracks (see Figure 7.3). To ensure the preservation for future research, the GVM collection is permanently stored within the *long-term archive of regional scientific research data* (LaZAR), hosted at the University of Jena, Germany [178]. For further details on the GVM collection, we refer to [180].

7.3 Reliability Indicators

In Section 7.3.1, we formalize the notion for our scenario. Then, we summarize the algorithms and annotations used in our investigations in Section 7.3.2. Subsequently, we introduce our three reliability indicators that measure F0-agreement (Section 7.3.3), overall confidence (Section 7.3.4), and F0-trajectory stability (Section 7.3.5).

[52] https://www.audiolabs-erlangen.de/resources/MIR/2017-GeorgianMusic-Scherbaum

Figure 7.3: Web-based interface for accessing the multitrack and video recordings of the GVM collection. Figure taken from [179].



- ● Camera microphone
- ○ Room microphone
- ○ Headset microphone 1
- ○ Headset microphone 2
- ○ Headset microphone 3
- ○ Throat microphone 1
- ○ Throat microphone 2
- ○ Throat microphone 3

7.3.1 Formalization

In our experiments, we consider several F0-estimation algorithms applied to one audio recording. Let M be the number of algorithms. Let us assume, a given F0-estimation algorithm outputs a frequency value as well as a confidence value (a value between 0 and 1) for each discrete time index $n \in [1:N]$ with $N \in \mathbb{N}$. Then, let $\eta : [1:N] \to \mathbb{R}$ be the resulting frequency trajectory and $\gamma : [1:N] \to [0,1]$ the corresponding confidence trajectory. Note that this definition can be understood as special case of the definition in Section 2.3.1 by requiring an algorithm to output one frequency and confidence value for each time index $n \in [1:N]$. For our M algorithms, let

$$\mathcal{T} := \{\eta_1, \ldots, \eta_{\mathrm{M}}\} \tag{7.1}$$

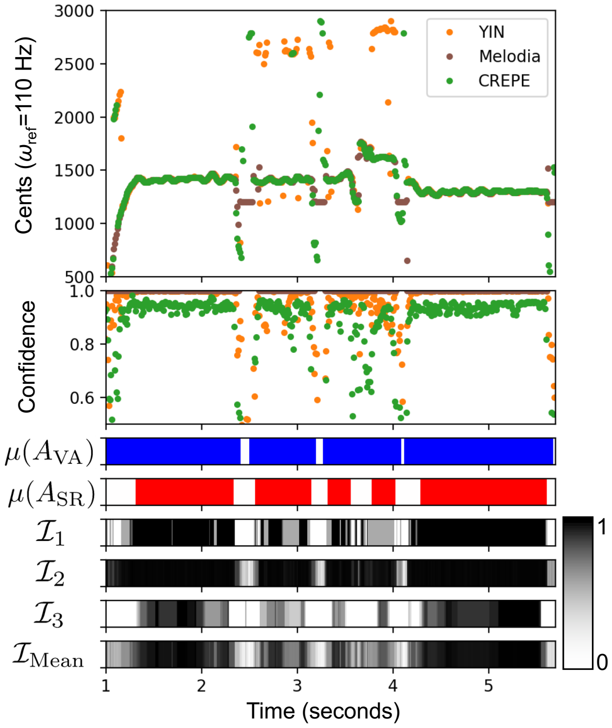
and

$$C := \{\gamma_1, \ldots, \gamma_{\mathrm{M}}\} \tag{7.2}$$

be the corresponding sets of trajectories, where η_m is the frequency trajectory and γ_m the confidence trajectory for the m^{th} algorithm, $m \in [1:M]$.

Furthermore, let $\eta^{\mathrm{Ann}} : [1:N] \to \mathbb{R} \cup \{*\}$ be an F0-annotation, with $\eta^{\mathrm{Ann}}(n) = *$ where the frequency value is left unspecified. We denote the set of all time frames where the annotation is active as $\mu(\eta^{\mathrm{Ann}}) := \{n \in [1:N] : \eta^{\mathrm{Ann}}(n) \neq *\}$.

Figure 7.4: Estimated F0-trajectories, confidences, annotations, and reliability indicators for the middle voice in the song “Kriste Aghsdga”.



7.3.2 Algorithms and Annotations

In our investigations, we consider the three ($M = 3$) algorithms YIN [48] (see Section 2.3.2), Melodia [169] (see Section 2.3.4), and CREPE [97] (see Section 2.3.5). While YIN and CREPE are designed for monophonic F0-estimation, Melodia was originally developed for the task of predominant melody estimation. Note that the selection of algorithms in this work is exemplary and our measures are not restricted to this specific set of algorithms. For extracting F0- and confidence trajectories, we use the publicly available YIN and Melodia Vamp plugins[53] together with the open-source tool Sonic Annotator [28], as well as the CREPE Python package[54]. All algorithms are applied with default parameter settings. For YIN and CREPE, we use the continuous confidence output of the implementations, whereas for Melodia, we derive binary confidence trajectories from the voice activity decision made by the algorithms.

Additionally, we consider two types of manual annotations. $\eta_{\mathrm{VA}}^{\mathrm{Ann}}$ assumes annotated F0-values in cents for parts where the singing voice is active (VA) and the symbol ‘$*$’ elsewhere. Similarly, $\eta_{\mathrm{SR}}^{\mathrm{Ann}}$ assumes annotated F0-values in cents for roughly stable regions (SR) of the F0-trajectory and the symbol ‘$*$’ elsewhere. Note that typically, the F0-values in $\eta_{\mathrm{SR}}^{\mathrm{Ann}}$ form a subset of the F0-values in $\eta_{\mathrm{VA}}^{\mathrm{Ann}}$. We manually generate $\eta_{\mathrm{SR}}^{\mathrm{Ann}}$ using the publicly available tool Tony [111], which is based on the algorithm pYIN [109] (see Section 2.3.3). Furthermore, we generate $\eta_{\mathrm{VA}}^{\mathrm{Ann}}$ by restricting automatically extracted

[53] `https://vamp-plugins.org/`

[54] `https://github.com/marl/crepe`

pYIN trajectories (also obtained using a Vamp plugin) to manually annotated regions where the singing voice is active using Sonic Visualiser [29]. In order to account for different hop sizes of the algorithms and annotations, we resample all F0-trajectories, annotations, and confidences to a time grid with a resolution of 10 ms. Furthermore, we quantize the F0-trajectories and annotations to a frequency resolution of 10 cents using $\omega_{\mathrm{ref}} = 110$ Hz.

As a running example in this section, we consider a recording of the three-voice song "Kriste Aghsdga" (GVM-ID 097) of the GVM collection. The three performing singers frequently use pitch slides at the beginning and end of sung notes, which is a characteristic stylistic element in traditional Georgian vocal music. Figure 7.4 shows a superposition of the resulting F0-trajectories extracted from the throat microphone recording of the middle voice for a short excerpt from our running example. Furthermore, the color-coded activities $\mu(\eta^{\mathrm{Ann}}_{\mathrm{VA}})$ and $\mu(\eta^{\mathrm{Ann}}_{\mathrm{SR}})$ are visualized.

Given the sets $\mathcal{T}$ and C, we now introduce three reliability indicators $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$. The frame-wise arithmetic mean of the three indicators is denoted as $\mathcal{I}_{\mathrm{Mean}}$.

7.3.3 F0-Agreement

For measuring the agreement of the F0-trajectories, we consider $P = \binom{M}{2}$ trajectory pairs $(\eta_i, \eta_j) \in \mathcal{T} \times \mathcal{T}$, with $i < j$. For each pair, we compute the difference between the trajectories by

$$\Delta_p(n) = \begin{cases} 1, & \text{for } |\eta_i(n) - \eta_j(n)| \leq \varepsilon_{\mathcal{I}}, \\ 0, & \text{otherwise,} \end{cases} \tag{7.3}$$

with pair-index $p \in [1 : P]$ and $\varepsilon_{\mathcal{I}}$ being a threshold in cents which defines the strictness of the measure. In our experiments, we set $\varepsilon_{\mathcal{I}} = 10$ cents. Compared to a 50 cents tolerance, which is typically used in standard evaluation metrics for evaluating pitch accuracy [10, 170], the chosen threshold is rather strict. Considering the 10 cents quantization of our trajectories, the threshold accounts for possible rounding artifacts caused by quantization. For practical reasons, we work with a fixed $\varepsilon_{\mathcal{I}}$ in our experiments and leave further investigations on the role of $\varepsilon_{\mathcal{I}}$ to future research. Our first reliability indicator is defined as the arithmetic mean of the differences over all pairs:

$$\mathcal{I}_1(n) := \frac{\sum_{p=1}^{P} \Delta_p(n)}{P}. \tag{7.4}$$

Only if the F0-estimates of all algorithm pairs agree, $\mathcal{I}_1(n) = 1$, as shown in our running example in Figure 7.4. In parts where the F0-estimates strongly deviate (e.g., at 2.5 sec) one obtains $\mathcal{I}_1(n) = 0$. In the part between 2.5–4 sec, there are some octave jumps by YIN and CREPE, which cause the agreement to decrease.

7.3.4 Overall Confidence

Our second reliability indicator combines the confidence outputs of the algorithms and is defined as the arithmetic mean of the confidences over all algorithms:

$$\mathcal{I}_2(n) := \frac{\sum_{m=1}^{M} \gamma_m(n)}{M}. \tag{7.5}$$

Note that in order for $\mathcal{I}_2$ to deliver meaningful indications, all trajectories are required to have values in the same value range, ideally making use of the entire $[0, 1]$ interval. If this requirement is not fulfilled, we use suitable normalization techniques or a binarization of the confidence using the algorithm's voice activity decision to balance out the confidence value distributions. In particular, we use binarized confidence trajectories for Melodia. In Figure 7.4, $\mathcal{I}_2$ indicates high overall confidence in most of the parts where the voice is active, thus showing high agreement with $\mu(\eta_{\mathrm{VA}}^{\mathrm{Ann}})$.

7.3.5 F0-Trajectory Stability

Our third indicator $\mathcal{I}_3$ measures reliability with respect to the local stability of the estimated F0-trajectories. A trajectory region is considered stable if it exhibits a roughly horizontal structure (up to some tolerance). In order to detect such stable regions in an F0-trajectory, we make use of the automatic approach based on morphological filters described in Section 6.3.2. In a first step, we compute two filtered versions of the trajectory, one by using a min-filter (erosion) and one by using a max-filter (dilation) with filter lengths $L \in \mathbb{N}$. L controls the smoothness of the filtered trajectories and affects the sensitivity of the stable region detection to sudden jumps in the trajectories. For practical reasons, we fix $L = 15$ (150 ms) in our experiments. The value roughly corresponds to the filter length determined in the study in Section 6.4.1 and might need to be adapted to other application scenarios. We leave further investigations on the role of L to future work. In a second step, we compute the frame-wise absolute difference between the max- and the min-filtered trajectory (also referred to as envelope width). All regions where the envelope width is lower than or equal to a certain threshold τ given in cents are considered stable. The algorithm outputs an activity function $\mu_{\mathrm{SR}} : [1 : N] \rightarrow \{0, 1\}$, where $\mu_{\mathrm{SR}}(n) = 1$ in stable regions and $\mu_{\mathrm{SR}}(n) = 0$ in unstable regions. In order to account for trajectory fluctuations of different extent, we consider a set of envelope-width thresholds $\mathcal{W} = \{20, 40, 60, 80, 100\}$, with 20 cents being a very strict threshold allowing for almost no trajectory fluctuations, and 100 cents being a generous threshold allowing for fluctuations of up to a semitone (e.g., vibrato).

Let $\mu_{\mathrm{SR},m}^{\tau}$ be the stability indicator for the m^{th} algorithm $m \in [1 : M]$ and threshold $\tau \in \mathcal{W}$. Then, $\mathcal{I}_3$ is defined as the arithmetic mean as follows:

$$\mathcal{I}_3(n) := \frac{\sum_{m=1}^{M} \sum_{\tau \in \mathcal{W}} \mu_{\mathrm{SR},m}^{\tau}(n)}{M \cdot |\mathcal{W}|}, \tag{7.6}$$

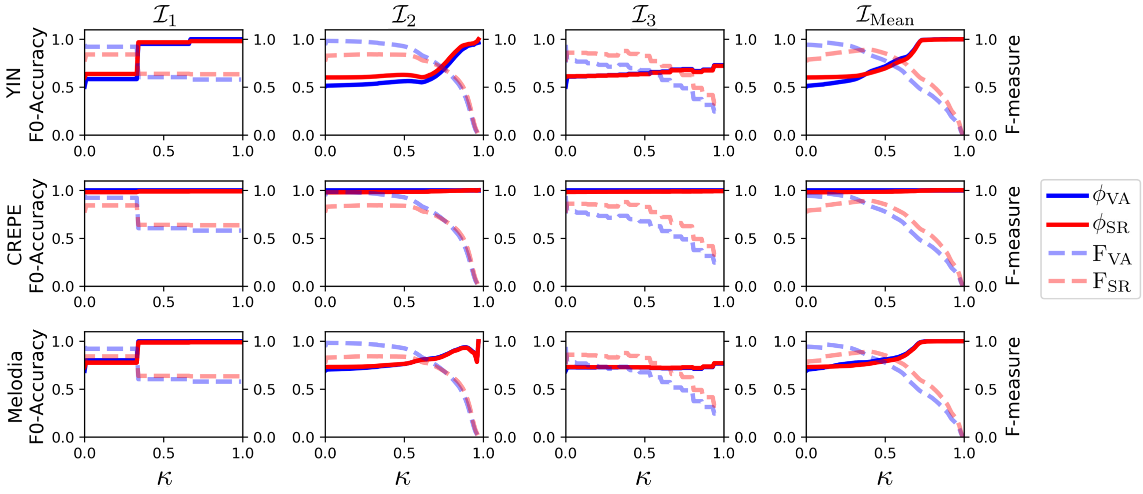


Figure 7.5: F-measure and F0-accuracy for all indicators and algorithms with respect to reliability threshold κ averaged over five recordings.

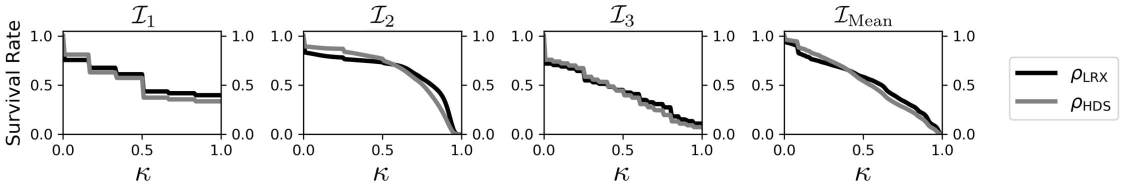


Figure 7.6: Survival rates for LRX and HDS microphones with respect to threshold κ averaged over 249 tracks.

for $n \in [1 : N]$. As one can see in Figure 7.4, $\mathcal{I}_3$ indicates high reliability in regions where all estimated F0-trajectories are roughly stable and therefore strongly coincides with $\mu(\eta^{\text{Ann}}_{\text{SR}})$.

7.4 Evaluation Using Labeled Data

In order to study the behavior of our indicators, we apply different thresholds $\kappa \in [0, 1]$ on our reliability indicators $\mathcal{I} : [1 : N] \to [0, 1]$. The resulting (enduring) subsets of our discrete time axis are given as $\mathcal{E}_\kappa = \{n \in [1 : N] : \mathcal{I}(n) \geq \kappa\}$. The higher κ, the smaller is the obtained subset. For a given subset $\mathcal{E}_\kappa$, we evaluate the agreement with an annotated voice activity $\mu(\eta^{\text{Ann}})$ using the standard retrieval metrics precision (P), recall (R), and F-measure (F) defined as

$$\text{P} := \frac{|\mathcal{E}_\kappa \cap \mu(\eta^{\text{Ann}})|}{|\mathcal{E}_\kappa|}, \quad \text{R} := \frac{|\mathcal{E}_\kappa \cap \mu(\eta^{\text{Ann}})|}{|\mu(\eta^{\text{Ann}})|}, \quad \text{F} := \frac{2 \cdot \text{P} \cdot \text{R}}{\text{P} + \text{R}}. \tag{7.7}$$

Furthermore, we set $\text{P} := 0$ for $|\mathcal{E}_\kappa| = 0$, $\text{R} := 0$ for $|\mu(\eta^{\text{Ann}})| = 0$, and $\text{F} := 0$ for $\text{P} + \text{R} = 0$. The standard definition of the F-measure equally weights precision and recall. The weighting may have to be adapted depending on the application scenario. As a further analysis step, we evaluate the F0-accuracy of estimated F0-trajectories within the subsets with respect to a reference annotation. Given an F0-trajectory

η restricted to the given subset $\mathcal{E}_\kappa$ and an annotation η^{Ann}, we define the F0-accuracy ϕ as

$$\phi := \frac{|\mathcal{E}_\kappa \cap \mu(\eta^{\mathrm{Ann}}) \cap \{n \in [1:N] : |\eta(n) - \eta^{\mathrm{Ann}}(n)| \leq \varepsilon_{\mathrm{e}}\}|}{|\mathcal{E}_\kappa \cap \mu(\eta^{\mathrm{Ann}})|}, \tag{7.8}$$

with ε_{e} being the evaluation tolerance parameter in cents. In our experiments, we use a strict value of $\varepsilon_{\mathrm{e}} = 10$ cents, to basically allow for quantization errors.

In our evaluation, we expand the scenario described in Section 7.3.2 to all five songs of the GVM subset. In the following, we consider the three algorithms YIN, CREPE, and Melodia applied on the throat microphone recordings of the middle voices. Furthermore, we manually generated the annotations $\eta^{\mathrm{Ann}}_{\mathrm{VA}}$ and $\eta^{\mathrm{Ann}}_{\mathrm{SR}}$ for these middle voice tracks (we crosschecked the annotations in spot-checks). The F-measure and F0-accuracy with respect to $\eta^{\mathrm{Ann}}_{\mathrm{VA}}$ are denoted as F_{VA} and ϕ_{VA}, whereas the evaluation measures with respect to $\eta^{\mathrm{Ann}}_{\mathrm{SR}}$ are denoted as F_{SR} and ϕ_{SR}, respectively.

Figure 7.5 shows the evaluation metrics for all algorithms averaged over all five recordings for each reliability indicator and algorithm with respect to the threshold κ. For almost all algorithms and reliability indicators, we observe an increasing F0-accuracy along with an increasing κ. The sudden drop in Melodia's ϕ curves for $\mathcal{I}_2$ occurs due to a high number of octave errors in regions with high confidence in one of the five recordings. For CREPE, the F0-accuracy is close to 1 for all values of κ, which indicates that the algorithm performs well on our annotated data. Note that the F-measure curves for a specific reliability indicator are identical for all algorithms, since the F-measure only depends on the chosen indicator $\mathcal{I}$, threshold κ, and annotation η^{Ann}. For high values of κ, only few F0-values remain, which causes the voice activity F-measures to decrease.

In conclusion, our indicators give cues on the reliability of F0-estimates at a given time instance in the audio signal. However, they are less suitable to assess the accuracy of a specific algorithm's estimate, since high reliability does not guarantee correct estimates (e.g., in the case of all algorithms outputting wrong estimates). The choice of a suitable threshold κ depends on the algorithms' individual performances, the chosen reliability indicator, and the target annotation or application.

7.5 Exploring Unlabeled Audio Collections

In this section, we want to demonstrate the potential of our reliability indicators for exploring unlabeled datasets. When approaching new audio collections, one may want to have a compact overview on how reliably automatic F0-extraction algorithms perform under the acoustical and musical conditions provided by the data. As already shown in Section 3.3.2, HDS and LRX microphone recordings constitute two different acoustic conditions.

In the following, we consider a subset of the GVM collection constituting 85 performances (all performances from the region Svaneti). More specifically, the subset includes 249 tracks (ca. 9 hours duration) for each microphone type. In order to explore the reliabilities measured by our indicators for the two different microphone types, we introduce a measure referred to as *survival rate* and denoted as ρ. The measure indicates the portion of remaining trajectory values after thresholding $\mathcal{I} : [1 : N] \to [0, 1]$ with $\kappa \in [0, 1]$ and is defined as follows:

$$\rho := \frac{|\mathcal{E}_\kappa|}{N}. \tag{7.9}$$

The survival rates for LRX and HDS microphone signals are denoted as ρ_{LRX} and ρ_{HDS}, respectively. In this experiment, we expand the setup described in Section 7.3.2 by adding pYIN to our set of algorithms. Figure 7.6 shows the two survival rates averaged over all 249 tracks with respect to the threshold κ. The graphs show that for high values of κ, ρ_{LRX} is larger than ρ_{HDS}, whereas for low values of κ, ρ_{HDS} is larger than ρ_{LRX}. This suggests a slightly better discriminability between reliable and unreliable frames for LRX signals.

Rather than advocating a specific indicator or a specific threshold κ, we see the proposed reliability indicators as a toolkit for measuring reliability of automatically extracted F0-trajectories with respect to F0-agreement, overall confidence, and F0-trajectory stability. Depending on the application, one may consider different indicators or suitably weighted combinations of them. Furthermore, one may adapt the selection of F0-extraction algorithms and fine-tune the individual indicators' parameters ($\varepsilon_{\mathcal{I}}$, L, and τ) to account for the specific acoustical and musical properties of the audio material.

7.6 Conclusions

In this chapter, we presented three indicators for measuring the reliability of F0-trajectories extracted from singing voice recordings. The indicators are based on the outputs of several algorithms and measure reliability with respect to F0-agreement, overall confidence, and F0-trajectory stability. As one of our main contributions, we introduced the reliability indicators in a mathematically rigorous way. Furthermore, we evaluated the behavior of the indicators on a set of manually annotated vocal F0-trajectories. While our indicators cannot replace manual F0-annotations, they can be used as an efficient tool to obtain cues on the reliability of automatically extracted F0-trajectories. In this way, our work paves the way for tonal analysis (e.g., melodic or harmonic intervals) and exploration of large unlabeled audio collections such as the GVM collection.

8 Towards Computational Ethnomusicology: A Case Study of Georgian Funeral Songs

This chapter is based on [164]. The first author Sebastian Rosenzweig is the main contributor to this article. In collaboration with his supervisor Meinard Müller and Frank Scherbaum, he developed the ideas, formalized and implemented the approaches, conducted the experiments and wrote the article. Frank Scherbaum and Nana Mzhavanadze created the annotations.

In this chapter, we show how computational methods, such as the approach for detecting stable regions in F0-trajectories from Section 6.3.2, can be interactively applied for research in the field of computational ethnomusicology. To this end, we conduct a case study on three-voiced funeral songs from Svaneti in North-West Georgia (also referred to as Zär). As one contribution of this chapter, we present an annotated multitrack dataset of Zär recordings which we release under an open-source license for research purposes. As a second contribution, we introduce two interactive computational tools for detecting stable, note-like events and compensating pitch drifts in performances. Our tools were developed in close collaboration with ethnomusicologists and allow for incorporating domain knowledge (e.g., on singing styles or musically relevant harmonic intervals) in the different processing steps. In a case study using our Zär dataset, we evaluate our tools by computing pitch inventories (pitch-class histograms) and subsequently discuss the potential of interactive computational tools for interdisciplinary research.

8.1 Introduction

Three-voiced funeral songs (or dirges, also referred to as Zär) from the region Svaneti in North-West Georgia have gained special attention among ethnomusicologists since they represent one of the oldest forms of collective music-making in Georgia [95]. Zär performances exhibit two musical peculiarities, which can be observed when looking at the F0-trajectories of the singers' voices as depicted in Figure 8.1a. First, the singers tend to use pitch slides at the beginning and end of sung notes (also referred to as portamento). Second, throughout a Zär performance, the singers may jointly and intentionally drift upwards in pitch by even more than 500 cents [174]. The presence of pitch slides and pitch drifts constitutes

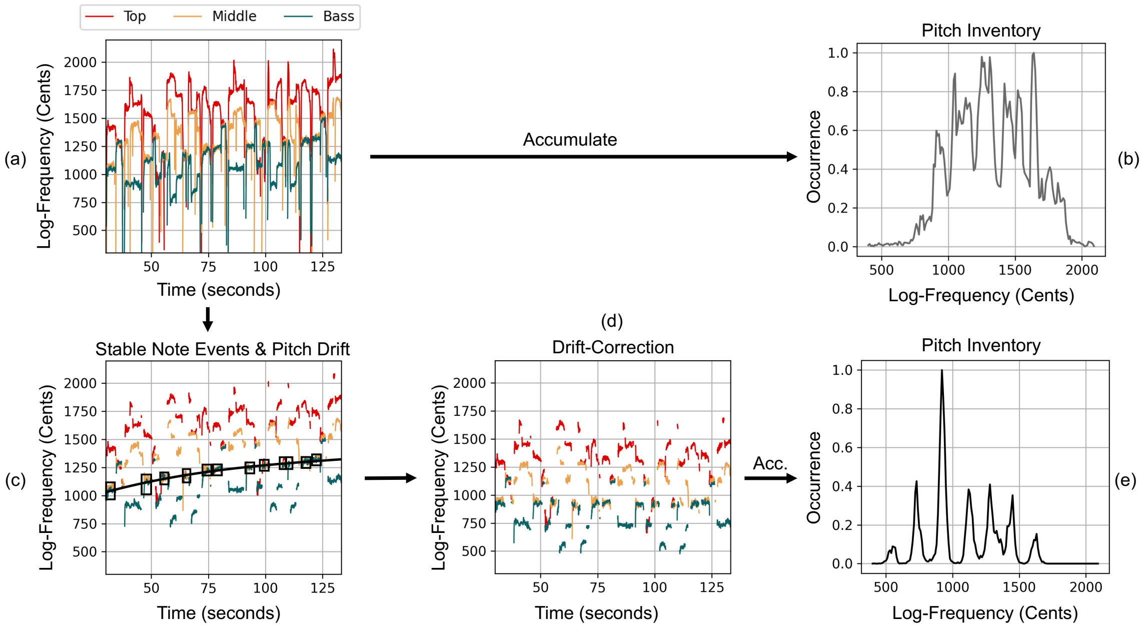


Figure 8.1: Pitch inventory computation for a three-voiced Zär performance. **(a)** Original F0-trajectories. **(b)** Pitch Inventory based on (a). **(c)** Annotated stable note events and pitch drift curve (black line). **(d)** Drift-corrected stable note events. **(e)** Pitch inventory based on (d).

a challenge for tonal analysis or transcription, as we will illustrate in the following. An important part of tonal analysis is the determination of pitch inventories (or pitch-class histograms) [67, 102, 105, 172, 204], which can be computed by accumulating the F0-values over time. As one can see in Figure 8.1b, pitch slides and drifts may result in noisy and blurry pitch inventories, which are hard to interpret or even meaningless for tonal analysis.

To tackle this problem, one strategy is to remove pitch slides and to compensate for pitch drifts prior to computing pitch inventories. Such tasks typically need to be conducted by experts with domain knowledge. In the context of four ethnomusicological studies on a set of eleven multitrack recordings of Zär performances [129, 130, 174, 175], domain experts annotated stable note events (F0-values between pitch slides) for all voices, as depicted in Figure 8.1c. Subsequently, the ethnomusicologists selected note events that best reflect the pitch drift of the performances (see the black boxes in Figure 8.1c) and determined pitch drifts through polynomial curve fitting (see the black line in Figure 8.1c). After drift correction with the (suitably normalized) drift curve, one obtains the drift-corrected stable note events as depicted in Figure 8.1d and the pitch inventory as depicted in Figure 8.1e. As one can see, in contrast to the uncorrected pitch inventory from Figure 8.1b, the pitch inventory based on the annotated material exhibits a sharper distribution. Through comparison of the pitch inventories for all eleven performances (determined in the same way), ethnomusicologists could show that the melodic step sizes in Zär vary between approximately 150 and 180 cents, which is an important cue towards understanding the traditional Georgian tuning system [174, 181]. However, conducting such annotation processes using existing semi-automatic annotation tools is labor-intensive and requires manual corrections. This also makes it hard to conduct similar studies on larger corpora.

In this chapter, we show that computational tools can support the analysis of field recordings by automizing some of the labor-intensive annotation tasks under the guidance of a domain expert. As one contribution, we compiled a dataset including the multitrack recordings and the carefully crafted annotations from the musicological studies, which we release under an open-source license for research purposes.[55] As our main technical contribution of this chapter, we present two computational tools with visual feedback mechanisms that allow for incorporating musical expert knowledge to the different processing steps. Our first tool, based on the approach for detecting stable regions in F0-trajectories from Section 6.3.2, enables the user to determine stable, note-like events. The method's parameters can be tuned according to musical characteristics such as the singing style. Our second tool is based on a filtering technique for musically relevant harmonic intervals (such as the unison or the fifth in Georgian vocal music [34, 161, 176]) to compensate for the pitch drift of a performance. In a case study based on our Zär dataset, we compare pitch inventories computed with our computational tools to the ones obtained from the expert annotations [129, 130, 174].

The remainder of this chapter is structured as follows. We describe related work in Section 8.2 and introduce our Zär dataset in Section 8.3. In Section 8.4, we formalize our computational tools. In Section 8.5, we describe our case study and discuss the potential of computational tools for ethnomusicological research. Finally, we summarize this chapter und outline future work in Section 8.6.

8.2 Related Work

Pitch slides and pitch drifts are a frequently observed phenomenon in a cappella singing, not least due to the great versatility of the human voice [197]. However, the musicological perspective on these phenomena often depends on the cultural context. For instance, pitch slides are often considered a sign of insufficient voice control in Western amateur choral singing while being a frequently and consciously used stylistic element in other music cultures such as traditional Georgian vocal music (see Chapter 5) or Indian Raga music [65, 101]. Similarly, pitch drifts are typically seen as unintended artifacts of tuning in Western ensemble singing [1, 86] while they are known to be a part of the performance practice in several non-Western music traditions [2, 96, 114, 117], including Georgian Zär [129, 130, 174, 175]. Thus, computational analysis of field recordings requires tools that can be adapted to the musical scenario by including musical or culture-specific knowledge [73]. Additionally, such tools need to offer suitable feedback mechanisms, e.g., visualizations or sonifications [205], which help to understand and guide computational methods.

[55] `https://www.audiolabs-erlangen.de/resources/MIR/2022-GeorgianMusic-Zaer`

Over the last years, a variety of tools for annotating and analyzing music recordings with a focus on Western music has been released [29, 107, 116, 126, 133, 146]. One of the most popular tools for transcribing monophonic audio recordings is the open source software Tony [111]. After loading an audio file, the tool automatically computes an F0-trajectory using the algorithm pYIN [109] (see Section 2.3.3). Via an interactive graphical user interface (GUI) with audiovisual feedback, the user can remove F0-values or choose alternative estimates. For transcription purposes, Tony automatically detects sung notes by segmenting the estimated F0-values into note objects using an HMM. Each note object is defined by a start time and end time in seconds, as well as an assigned F0-value (corresponding to the note's pitch). Using the interactive GUI, a user can split, merge, create or delete note objects. Finally, annotated F0-trajectories and note objects can be exported in a variety of text formats, including CSV and TXT. Tony does not offer functionalities to account for pitch drifts in performances.

The increasing scientific interest in non-Western music traditions [65, 73, 103, 132, 136, 188, 203, 205, 208] has led to the development of tools designed for processing and analyzing music in tuning systems other than 12-TET. One prominent example is Tarsos [191], a platform for analyzing pitch inventories and musical scales. Its GUI offers interactive sonifications and visualizations as well as sliders to control the included computational tools. After loading an audio file, the tool automatically computes an F0-trajectory (using the YIN algorithm [48] by default) and a pitch histogram. As opposed to Tony, Tarsos does not include functionalities to correct F0-estimates. However, it includes a tuneable "steady state filter" which allows a user to remove pitch slides in F0-trajectories. F0-trajectories and pitch histograms can be exported to different text formats and images. As for Tony, Tarsos lacks the functionality to compensate for pitch drifts in performances, which is essential for our Zär scenario.

8.3 Zär Dataset

In the following, we describe our Zär dataset consisting of multitrack recordings (Section 8.3.1), as well as F0-annotations (Section 8.3.2), stable note events (Section 8.3.3), and pitch drift annotations (Section 8.3.4). We also discuss how we cimpute pitch inventories from the annotations (Section 8.3.5).

8.3.1 Multitrack Recordings

Our Zär dataset is based on eleven recordings (GVM-IDs 198–208) from the GVM collection (see Section 7.2). We include the multitrack recordings of these performances as mono WAV files with a sampling frequency of 22 050 Hz and 16-bit encoding. For all performances, the dataset contains at least three throat microphone signals of at least three singers (the recordings are named with suffixes `ALRX1M`, `ALRX2M`, `ALRX3M`). The number of headset and room microphone signals varies for each performance. The eleven performances have a total duration of roughly 42 minutes.

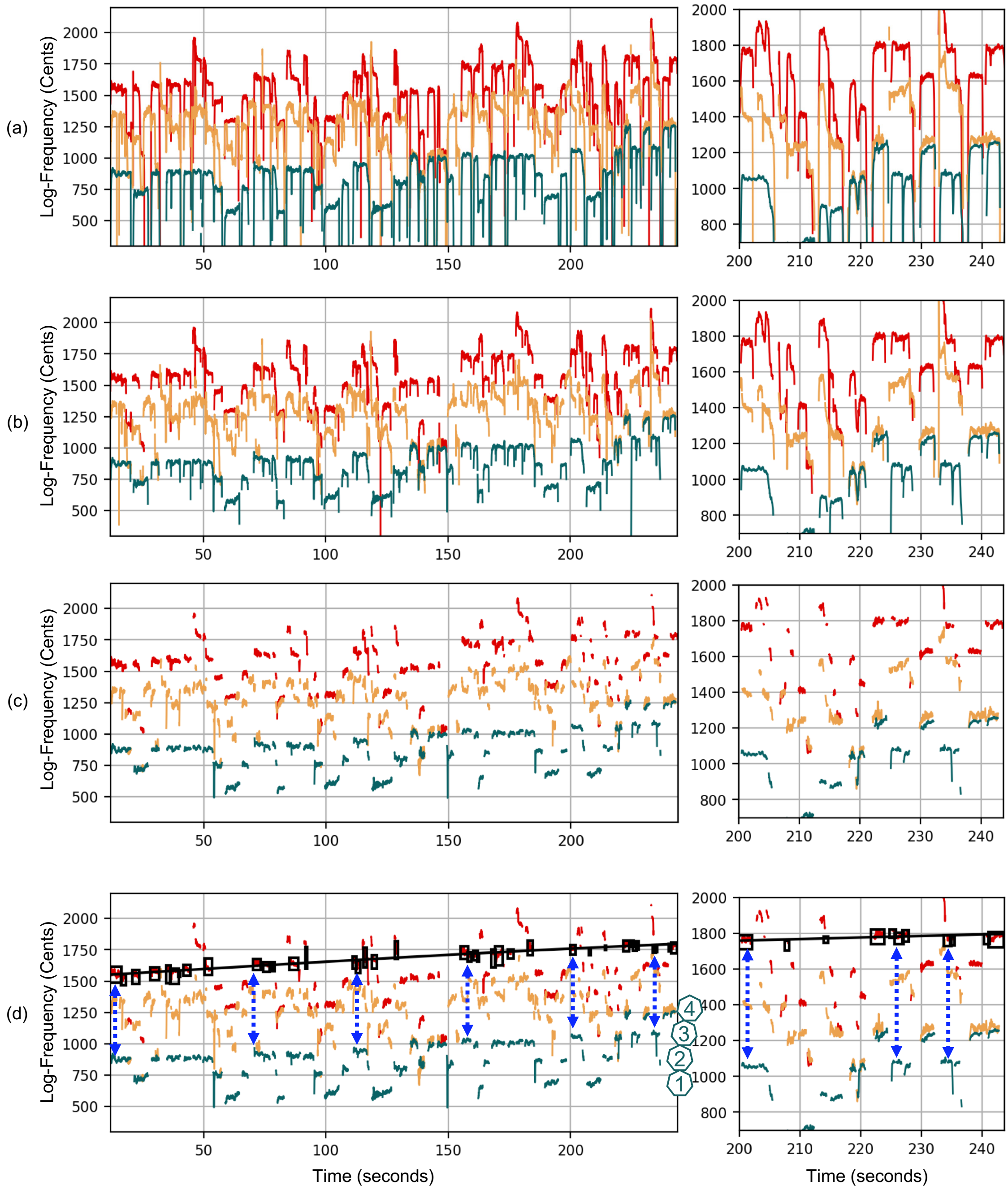


Figure 8.2: Annotation process for the Zär performance with GVM-ID 201. The right column shows zoomed regions. **(a)** F0-trajectories of top, middle, and bass voice. **(b)** F0-trajectories corresponding to Tony note objects. **(c)** Stable note events. **(d)** Selected stable note events (black rectangles) and fitted drift curve (black line).

8.3.2 F0-Annotations

In the context of previous studies [129, 130, 174], a Georgian ethnomusicologist semi-automatically annotated F0-trajectories of the three voices for all of the performances using the open-source tool

Tony [111]. Subsequently, a domain expert double-checked the annotations. Figure 8.2a shows the F0-annotation for the top, middle, and bass voice of the performance with GVM-ID 201, which serves as our running example in the remainder of this chapter. The trajectories show that sung notes often start, end, or are continuously connected with pitch slides, which is a frequently used stylistic element of traditional Georgian music (see Chapter 6). The F0-annotations are included as CSV files in our Zär dataset with a frequency resolution of 10 cents and a time resolution of 10 msec.

8.3.3 Stable Note Events

As mentioned in Section 8.2, Tony automatically segments the annotated F0-trajectories into note objects. The F0-trajectories corresponding to the note objects are depicted in Figure 8.2b. As one can see, Tony's automatic segmentation algorithm shortens most of the pitch slides. However, during tonal analysis or transcription of Zär performances, the remaining slides can still lead to a significant amount of blurring and inaccuracies. One way to remove the remaining pitch slides is to use the manual correction functionalities of Tony. However, this is a time-consuming and tedious task, which is infeasible when considering larger collections. In [174], the ethnomusicologists used a heuristic to remove pitch slides by cutting off 0.15 sec at the beginning and the end of each F0-trajectory within each note object. We refer to the shortened F0-trajectories as "stable note events." The stable note events for our running example are depicted in Figure 8.2c. As one can see, most of the pitch slides have been removed using this simple heuristic. In Section 8.4.1, we present a computational tool that helps to automize the detection of stable regions in F0-trajectories using interactive filtering techniques.

8.3.4 Pitch Drift Annotations

For determining the pitch drift in Zär performances, one can exploit the importance of specific harmonic intervals in traditional Georgian vocal music. For instance, parallel fifths, which are often avoided in Western composed music, frequently occur in Georgian polyphonic singing [34, 161, 176]. Often, the top and bass voice sing a fifth apart (700 cents), representing the "harmonic frame" of the performance. Additionally, the unison interval (0 cents) is of great relevance in Zär performances. Essentially all songs from the region Svaneti (not only funeral dirges) end in unison. In addition, throughout a performance, the three voices of a Svan song repeatedly meet in unison, which gives the associated pitches a particular musical importance (ethnomusicologists also consider unisons as "reference pitches" in traditional Georgian singing). As a consequence, ethnomusicologists hypothesize that the pitch drift of a performance can be documented through such musically important harmonic intervals [129, 130, 174].

In [174], the ethnomusicologists followed a two-step process to determine the pitch drift in the performances. First, using a visualization as depicted in Figure 8.2c, the experts visually identified a small number of stable note events in one of the voices that, according to their musical expertise, best reflect the pitch drift

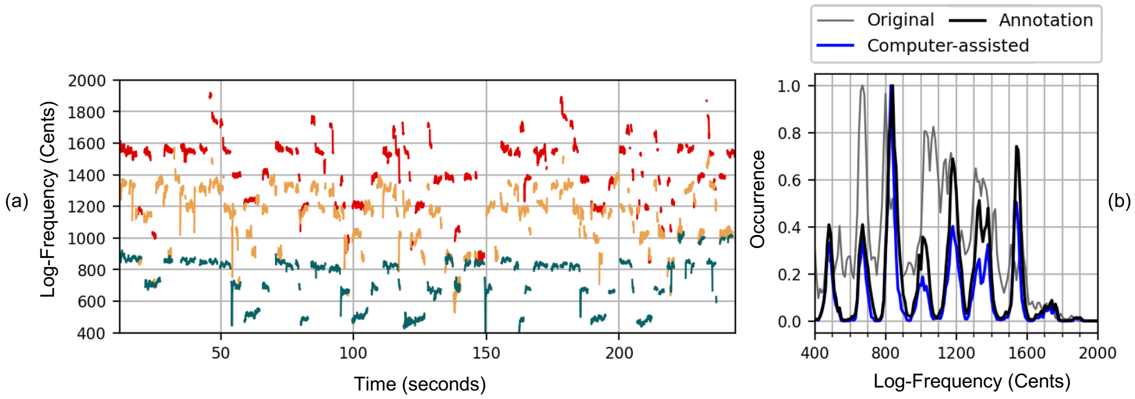


Figure 8.3: Pitch inventory computation for performance with GVM-ID 201. **(a)** Drift-corrected stable note events. **(b)** Max-normalized pitch inventories.

of the performance. For our running example, the experts selected stable note events of the top voice, which are indicated with black rectangles in Figure 8.2d. As one can see, the note events were chosen to correspond to the same *scale degree* (a group of note events that roughly correspond to the same pitch after removing the drift of the performance). For instance, in Figure 8.2d, the four scale degrees of the bass voice are marked using numbers in ascending order. We see that not all scale degrees are equally suitable to determine the pitch drift of the performance since some scale degrees (such as scale degree 1 of the bass voice) contain only a few stable note events. The identification of scale degrees and the selection of suitable note events require musical knowledge and need to be done with care. On closer inspection of our example, we also see that the selected note events often go along with parallel fifths of top and bass voice (see blue arrows), which shows the relevance of the fifth interval for recognizing the pitch drift of the performance.

Second, to model the pitch drift of the performance, the ethnomusicologists fitted a polynomial curve through the selected stable note events. The study in [174] revealed that polynomials of third order are sufficiently suited to describe the pitch drift of the performances. Figure 8.2d shows the fitted polynomial (black solid line) for our example performance. Our Zär dataset includes computed drift curves with a time resolution of 10 msec as CSV files. In Section 8.4.2, we present a computational tool based on interactive filtering techniques for harmonic intervals and scale degrees that supports the manual selection process of note events for determining the pitch drift.

8.3.5 Pitch Inventories

One key towards understanding traditional Georgian tuning lies in analyzing the pitch inventories of the singers. To compute pitch inventories, the ethnomusicologists in [174] first drift-corrected the annotated stable note events with the pitch drift curve. Figure 8.3a shows the drift-corrected stable regions from Figure 8.2c using the drift curve from Figure 8.2d. As one can see, the scale degrees of the three voices follow a roughly horizontal line. Subsequently, the experts computed histograms over the drift-corrected stable note events. The black line in Figure 8.3b shows the obtained max-normalized pitch inventory with a binning resolution of 10 cents. In contrast to the pitch inventory computed on the original F0-trajectories without drift correction (gray line), the annotated pitch inventory exhibits a heptatonic peak structure (seven melodic intervals per octave). The spacing between the peaks of a pitch inventory reflects the average melodic step sizes used in the performance. The pitch inventory of our running example reveals step sizes of 150–180 cents, which coincides with step sizes measured in the other Zär performances, as well as in historic recordings of liturgical chants by the former master chanter Artem Erkomaishvili [181]. Through such analysis, ethnomusicologists can obtain important cues on the tonal organization of traditional Georgian vocal music. For an in-depth musicological analysis of Zär, we refer to [129, 130, 174, 175].

8.4 Interactive Computational Tools

In the following, we first introduce our computational tool for detecting stable regions (Section 8.4.1) and subsequently our tool for determining pitch drift (Section 8.4.2).

8.4.1 Stable Region Detection

In the following, we describe a computer-assisted approach for detecting stable regions in F0-trajectories. The method is based on an the approach described in Section 6.3.2 [158]. To better discuss the properties of our tool, we will recapitulate the basic steps of our approach. Furthermore, we follow the mathematical notion of an F0-trajectory as introduced in Section 2.3.1. We explain our method using an excerpt of the top voice in the performance with GVM-ID 201. Figure 8.4a shows the given trajectory η (black line), which contains several pitch slides. In a first step, we compute a dilated (max-filtered) trajectory $\eta^L_{\max}$ and an eroded (min-filtered) trajectory $\eta^L_{\min}$ defined by

$$\eta^L_{\max}(n) := \max\left\{\eta\left(n - \tfrac{L-1}{2} : n + \tfrac{L-1}{2}\right)\right\}, \tag{8.1a}$$

$$\eta^L_{\min}(n) := \min\left\{\eta\left(n - \tfrac{L-1}{2} : n + \tfrac{L-1}{2}\right)\right\}, \tag{8.1b}$$

for $n \in \mathbb{Z}$, where $L \in \mathbb{N}$ is assumed to be an odd integer. In max-filtering, the symbol $*$ is handled as $-\infty$, whereas in min-filtering it is handled as $+\infty$. Figure 8.4b shows the resulting trajectories $\eta^L_{\min}$ (orange) and $\eta^L_{\max}$ (green) for our running example using $L = 15$ (150 msec). In a next step, we compute the difference Δ^L between the dilated and eroded trajectories, also termed morphological gradient [154]:

$$\Delta^L(n) := |\eta^L_{\max}(n) - \eta^L_{\min}(n)|, \tag{8.2}$$

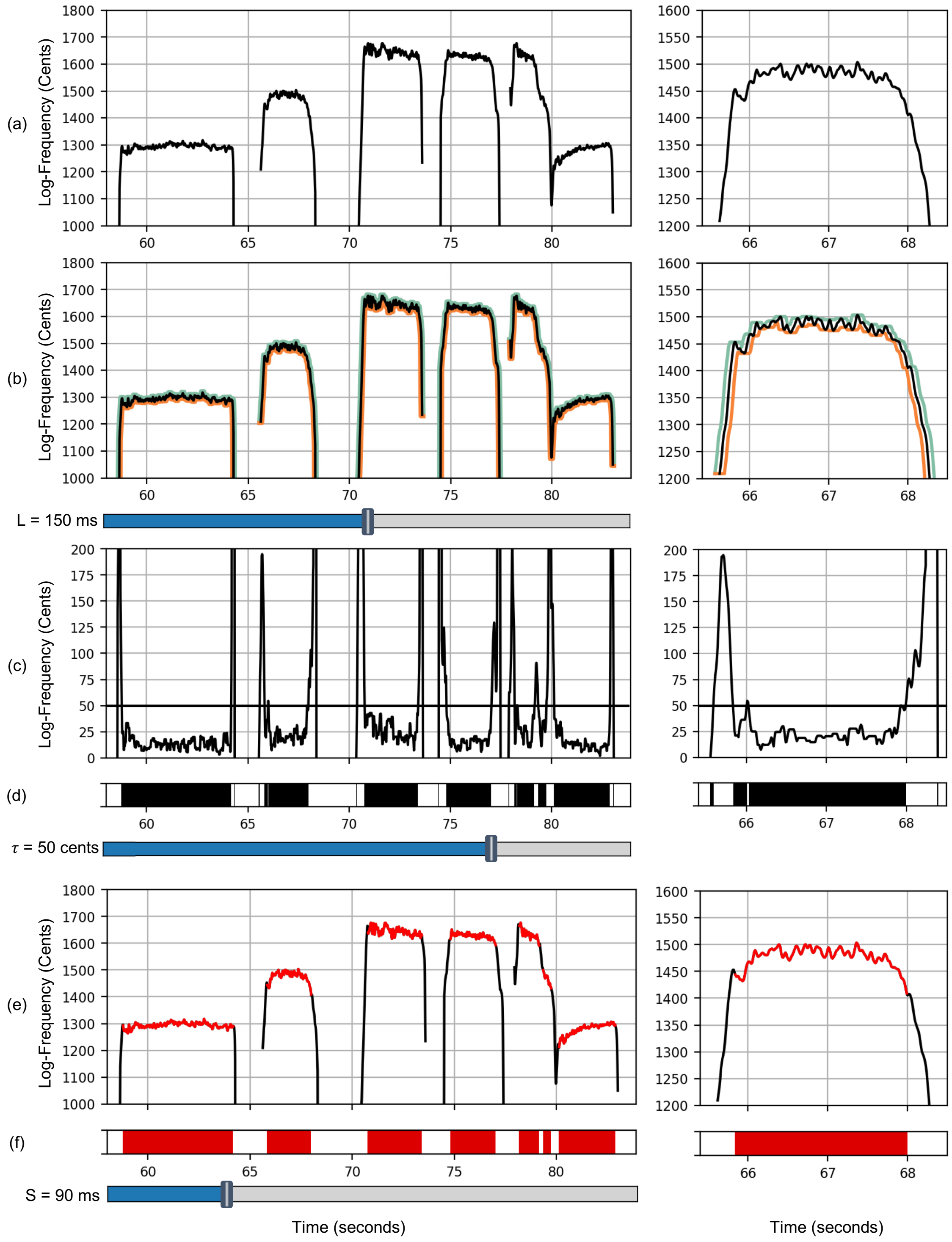


Figure 8.4: Interactive detection of stable regions for the example of the top voice in performance with GVM-ID 201. The right column shows zoomed regions. **(a)** Original trajectory. **(b)** Min-filtered trajectory η^L_{min} (orange) and max-filtered trajectory η^L_{max} (green) for given filter length $L = 15$ (150 msec). **(c)** Morphological gradient Δ^L **(d)** Activation function $\mu^{L,\tau}$ after thresholding with $\tau = 50$ cents. **(e)** Trajectory η^{Stable} restricted to stable regions (red). **(f)** Activation function $\mu^{\mathrm{L},\tau,\mathrm{S}}$ after smoothing with $S = 9$ (90 msec).

for $n \in \mathbb{Z}$, where we set $\Delta^L(n) = *$ whenever $\eta^L_{\max}(n)$ or $\eta^L_{\min}(n)$ are not defined. Figure 8.4c shows Δ^L for our running example. As one can see, Δ^L is large in non-stable parts (e.g., during pitch slides), whereas it is small in stable parts. After thresholding Δ^L with a chosen threshold $\tau > 0$ (given in cents), we obtain an activation function $\mu^{L,\tau}$ defined by

$$\mu^{L,\tau}(n) := \begin{cases} 1, & \text{for } \Delta^L(n) \leq \tau, \\ 0, & \text{otherwise,} \end{cases} \tag{8.3}$$

with $\mu^{L,\tau}(n) = 1$ indicating stable regions and $\mu^{L,\tau}(n) = 0$ indicating unstable (or undefined) regions. Figure 8.4d shows the activations $\mu^{L,\tau}$ for our running example after thresholding with $\tau = 50$ cents. As one can see, most of the stable regions of the trajectory have been correctly identified. However, there are some short passages that have been wrongly identified as stable regions (false positives). These passages result from filtering artifacts and can often be found at the beginning and end of pitch slides (e.g., at around 65.5 sec) or in between two fastly succeeding notes (e.g., at around 79 sec). Also, there are some short interruptions in stable regions (false negatives), e.g., at 66 sec.

In practice, one may want to remove these outliers and obtain coherent entities of F0-values, similar to the stable note events from Section 8.3.3. Therefore, as an extension to the original approach described in Chapter 6, we propose an optional smoothing step of the trajectory activations $\mu^{L,\tau}$ by applying a median filter:

$$\mu^{\mathrm{L},\tau,\mathrm{S}}(n) := \mathrm{median}\left\{\mu^{L,\tau}\left(n - \tfrac{S-1}{2} : n + \tfrac{S-1}{2}\right)\right\}, \tag{8.4}$$

where $S \in \mathbb{N}$ is assumed to be an odd integer and the symbol $*$ is handled as $-\infty$. Note that by setting $S = 1$, no smoothing is applied. In a final step, we compute the trajectory restricted to stable regions η^{Stable} by

$$\eta^{\mathrm{Stable}}(n) := \begin{cases} \eta(n), & \text{for } \mu^{\mathrm{L},\tau,\mathrm{S}}(n) = 1, \\ *, & \text{otherwise.} \end{cases} \tag{8.5}$$

Figure 8.4e,f show the resulting trajectory and its activation function after smoothing with a median filter of length $S = 9$ (90 msec). As one can see, most of the outliers have been removed (only the outlier at around 79 sec remains). One may further tackle such outliers by applying additional heuristics such as removing detected stable regions that fall below a certain minimal length or that exceed a certain variance. Note that the algorithm leaves the frequency values of the original F0-trajectory unaltered (e.g., no quantization or smoothing of frequency values), which is important for subsequent tonal analysis steps.

In practice, an ethnomusicologist (without explicit knowledge in signal processing) can use interactive visualizations similar to Figure 8.4 for tuning the three parameters L, τ, and S of our algorithm. The min-/max- filter length L controls the sensitivity of the method to (sudden) fluctuations in the F0-trajectory. Small L may lead to an increased number of false positives, while large L lead to an increased number of

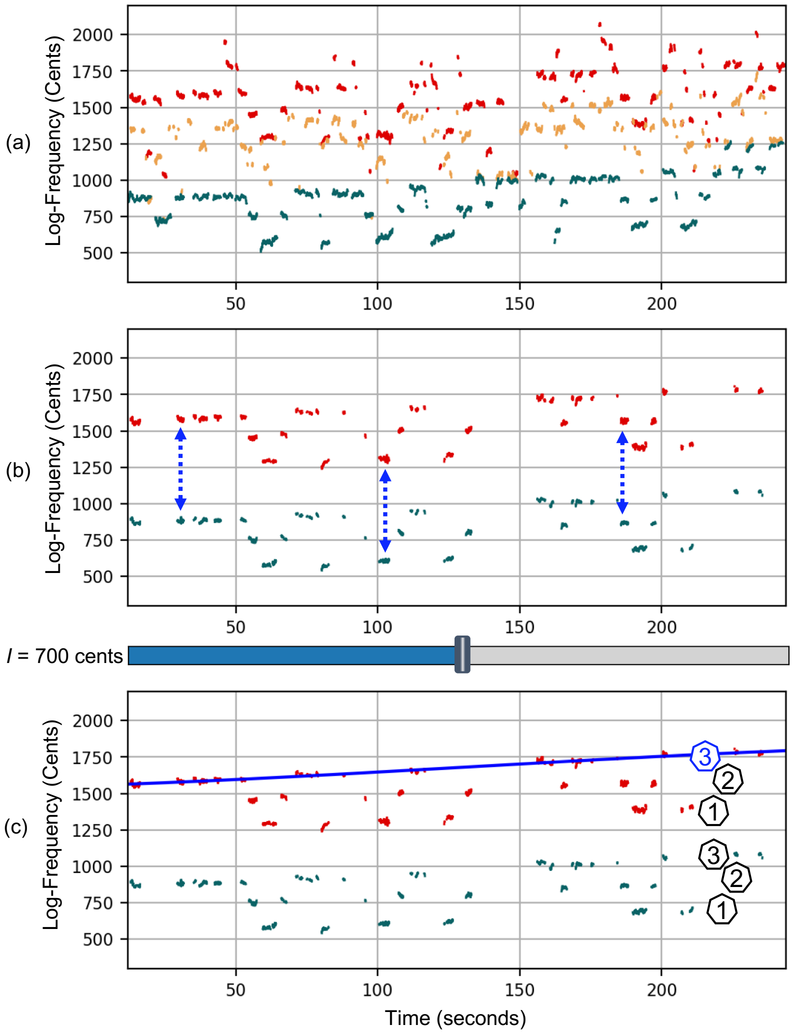


Figure 8.5: Interactive drift estimation illustrated by the performance with GVM-ID 201. **(a)** Stable note events. **(b)** Interval-filtered F0-trajectories with interval $I = 700$ cents using the top and bass voice ($M = 2$). **(c)** Polynomial fitting using the F0-values of a manually specified scale degree.

false negatives (in particular at the beginning and end of stable regions), see Section 6.3.2. The threshold τ can be seen as a tolerance parameter that specifies the maximally allowed fluctuation under which a trajectory is still considered to be stable. Therefore, τ may be tuned according to the singing style (e.g., the amount of vibrato) used in the performance or the singing proficiency. After determining L and τ, the smoothing filter of length S can be tuned to refine the detection by removing outliers observed in the activation function $\mu^{L,\tau}$, which results in stable, note-like events. In summary, the three parameters of our tool have an explicit and easy-to-understand meaning, which is important for use in interdisciplinary research.

8.4.2 Drift Estimation

In the following, we describe a computer-assisted approach for estimating the pitch drift of a Zär performance using interactive filtering techniques for harmonic intervals and scale degrees. Our approach is based on the hypothesis that certain (musically important) harmonic intervals capture the pitch drift of a performance (see Section 8.3.4). We explain our method along with Figure 8.5, using again the performance with GVM-ID 201 as an example. In the following, we assume a set of M trajectories

$$\mathcal{T} := \{\eta_1, ..., \eta_M\}, \tag{8.6}$$

where η_m is the F0-trajectory of the m^{th} voice, $m \in [1 : M]$. In our example, we consider F0-trajectories restricted to stable regions for the top, middle, and bass voice ($M = 3$) as depicted in Figure 8.5a. As one can see, the three singers continuously drift upwards over the course of the performance.

As discussed in Section 8.3.4, the pitch drift of a Zär performance is captured by certain musically important harmonic intervals (e.g., the unison or the fifth). Therefore, in a first step, we filter the given F0-trajectories with respect to a user-specified harmonic interval. For a given $m \in [1 : M]$ and an interval I in cents, let $\mathcal{H}_m^I$ denote the set that contains all time indices n for which there is at least one other trajectory η_k, $k \in [1 : M] \setminus \{m\}$ that is I cents apart up to a tolerance ε in cents. In other words:

$$\mathcal{H}_m^I := \big\{n \in \mathbb{Z} | \exists k \in [1 : M] \setminus \{m\} : I - \varepsilon \leq |\eta_m(n) - \eta_k(n)| \leq I + \varepsilon\big\}. \tag{8.7}$$

We then define the interval-filtered F0-trajectory η_m^I for voice m by

$$\eta_m^I(n) := \begin{cases} \eta_m(n), & \text{for } n \in \mathcal{H}_m^I, \\ *, & \text{otherwise.} \end{cases} \tag{8.8}$$

Figure 8.5b illustrates our interval-filtering for the fifth interval ($I = 700$ cents) with $\varepsilon = 20$ cents for the top and bass voice ($M = 2$). It can be seen that the remaining F0-values of top and bass voice are spaced roughly 700 cents apart (as indicated by the blue dotted arrows).

Similar to Section 8.3.4, in the next step, the user selects a scale degree that best reflects the pitch drift of the performance. In our example, the user chooses the third scale degree of the bass voice (counting the remaining scale degrees after interval filtering from low to high). The F0-values corresponding to the chosen scale can be obtained using a suitable clustering algorithm.

In our work, we use a simple two-step clustering method inspired by the Radon Transform [143]: first, we rotate the interval-filtered trajectories around the coordinate origin such that the entropy of a computed pitch inventory is minimized. The entropy indicates the peakedness of a distribution while being low for peaked distributions and high for flat distributions. Thus, the rotation angle that minimizes entropy constitutes an approximation of the linear drift slope of the performance. This entropy-minimizing rotation angle can be determined automatically through an exhaustive search over a musically meaningful range of angles. In our experiments, we assume that the singers do not drift more than ±1200 cents (or an octave) over the course of a performance, which is a reasonable choice for Zär performances [129, 174]. Second, we perform k-means clustering on the interval-filtered and rotated trajectories, with k corresponding to the number of scale degrees of each voice ($k = 3$ in our example). As in Section 8.3.4, we fit a polynomial of third order using the F0-values that correspond to the chosen scale degree. The resulting drift curve for our example is indicated by the blue line in Figure 8.5c. Note that polynomial fitting through a certain scale degree of the bass voice should result in a similar drift curve than polynomial fitting through the same scale degree of the top voice.

In practice, using our tool and visualizations similar to Figure 8.5, an ethnomusicologist can interactively explore and analyze how different harmonic intervals I and scale degrees reflect the pitch drift of the performance. An additional indicator for the correctness of a determined pitch drift are pitch inventories. The blue line in Figure 8.3b shows the pitch inventory of our running example obtained through drift-correcting the detected stable regions from Figure 8.5a with the normalized drift curve from Figure 8.5c. As one can see, the computed pitch inventory has a similar peak structure to the annotated pitch inventory, which indicates that the pitch drift has correctly been determined.

8.5 Computer-Assisted Analysis of Zär Performances

In this section, we discuss how our interactive computational tools can be applied to support ethnomusicological research. First, in a case study on Georgian Zär, we use our tools to reproduce pitch inventories of a previous study (Section 8.5.1). Second, in the light of our experimental results, we discuss the potential of computational tools for ethnomusicological research (Section 8.5.2).

8.5.1 A Case Study on Pitch Inventories

We now show how a domain expert can use our computational tools to reproduce the pitch inventories from the previous study on Georgian Zär [174]. Since the musical meaning of pitch inventories is hard to quantify and evaluate, we discuss different qualitative aspects of our work along with the visualizations in Figure 8.6. Figure 8.6a shows the annotated stable note events and pitch drifts (black lines) from our Zär dataset (Section 8.3.4) as reference.

In our case study, we start with the F0-annotations of the three voices described in Section 8.3.2. Note that in case no F0-annotations are at hand, one can use automatic approaches such as the one proposed in Chapter 7 to obtain reliable F0-estimates. In a first step, we use the tool introduced in Section 8.4.1 to determine stable regions in the F0-trajectories. Using interactive visualizations such as Figure 8.4, a domain expert can tune parameters L, τ, and S such that the pitch slides are removed. In our study, we set

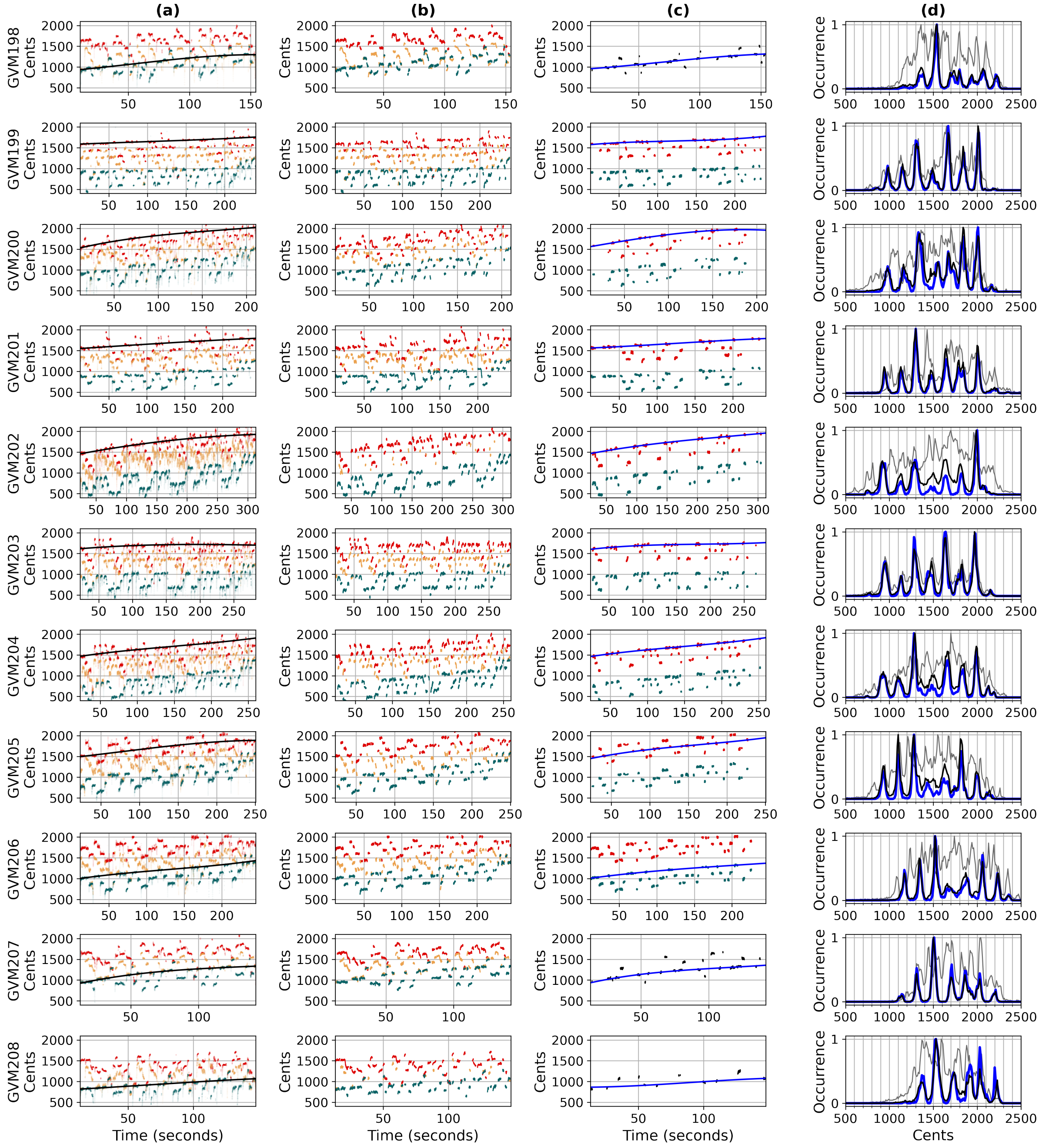


Figure 8.6: Pitch inventory computation for all eleven Zär performances. **(a)** Manually annotated stable note events and reference pitch drift. **(b)** Trajectories restricted to stable regions using the interactive tool from Section 8.4.1. **(c)** Interval-filtered trajectories and estimated pitch drift using the interactive tool from Section 8.4.2. **(d)** Pitch inventories (for legend, see Figure 8.3b).

$L = 15$ bins (150 msec), which corresponds to the value that the domain experts chose for determining stable note events from Section 8.3.3. Furthermore, we set $\tau = 50$ cents, which is a reasonable value for Georgian singing. To refine the detection, we empirically determined $S = 9$ bins (90 msec), which removes short outliers. Finally, we remove stable regions that are shorter than 100 msec to further refine the detection. The resulting trajectories restricted to stable regions are depicted in Figure 8.6b. Overall, the filtered trajectories resemble the manually annotated stable note events from Figure 8.6a.

In a second step, we use the tool introduced in Section 8.4.2 to determine the pitch drifts of the performances. Using musical domain knowledge and interactive visualizations such as Figure 8.5, an expert can choose an interval I and a scale degree that best capture the pitch drift of a performance. As explained in Section 8.3.5, the unison and the fifth interval are of special musical importance in Georgian Zär. For the performances with GVM-ID 199–206, which exhibit very prominent parallel fifths, we filtered for the fifth interval ($I = 700 \pm 20$ cents) of top and bass voice. For the performances with GVM-ID 198, 207, and 208, which exhibit less prominent parallel fifths, we chose to filter for the unison interval ($I = 0 \pm 20$ cents) considering all voices. The interval filtered trajectories of all performances are depicted in Figure 8.6c. Subsequently, we use the clustering algorithm described in Section 8.4.2 to automatically determine the scale degrees of the interval filtered trajectories. For the performances 199–206, we determined $k = 6$ clusters (3 for each voice), and for the performances 198, 207, and 208, we determined $k = 3$ clusters. In our case study, we selected similar scale degrees as the domain expert in the manual study. The fitted polynomial drift curves through the selected scale degrees are shown as blue lines in Figure 8.6c. As one can see, the drift curves have a similar progression compared to the annotated pitch drifts in Figure 8.6a. Note that instead of advocating a specific interval or scale degree, this case study shows only one way how the pitch drift in Zär performances can be determined. For instance, our computational tools enable domain experts to explore interval filtering for different harmonic intervals as well as suitable combinations, which may lead to more accurate drift estimates. We leave an investigation of these aspects for future work.

In a final step, we compensate for the pitch drift of the trajectories restricted to stable regions from Figure 8.6b with the normalized drift curves from Figure 8.6c before computing pitch inventories. In our experiments, we use a binning resolution of 10 cents and max-normalize all pitch inventories. Note that for tonal analysis, one is mainly interested in the relative peak positions of pitch inventories (see Section 8.3.5). In order to facilitate the visual comparison of pitch inventories across performances, we shift all F0-values by a constant amount (which differs for each Zär) in such a way that the final long note (with a duration of at least 1 sec) in the middle voice has a pitch of 1500 cents. Figure 8.6d shows the pitch inventories obtained from the original F0-trajectories without drift correction (gray), the pitch inventories based on the reference annotations (black), and the pitch inventories obtained using our copmuter-assisted tools (blue). We can see that the pitch inventories computed with the help of our interactive tools are very similar to the reference pitch inventories.

8.5.2 Applications to Computational Ethnomusicology

The study described in Section 8.3 exemplifies many of the challenges that ethnomusicological studies on field recordings face. Even relatively basic analysis tasks, such as the computation of pitch inventories, typically require multiple annotation steps conducted by domain experts. State-of-the-art annotation tools such as Tony or Praat face several limitations since they are either designed for Western music or lack necessary functionalities. This often leads to labor-intensive annotation processes with tedious manual correction steps. While these efforts were made for the eleven performances of our Zär dataset, similar studies on larger corpora, such as the whole GVM collection [180] (see Section 7.2), would be very time-consuming to perform. Also, such highly manual annotation and analysis processes can suffer from subjective decisions, thus making it hard to reproduce the results.

As our case study showed, computational tools can support ethnomusicological studies on field recordings by taking over specific, well-defined tasks of the annotation process under the guidance of a domain expert. Through tuning a few musically motivated parameters and suitable interactive visualizations, a domain expert could reproduce the pitch inventories that were obtained by tedious manual annotations in significantly less time. In this way, our computer-assisted procedure can accelerate and simplify musicological analyses as well as enable the exploration of large music corpora to gain new musicological insights.

8.6 Conclusions and Future Work

In this chapter, we presented a publicly available dataset based on eleven performances of three-voice Georgian Zär, which includes expert annotations of F0-trajectories, stable note events, and pitch drifts. The dataset is of high value for ethnomusicological research and the preservation of the Georgian musical heritage. Furthermore, we introduced two computational tools based on interactive filtering techniques for detecting stable regions in F0-trajectories and determining the pitch drift of the performances. In a case study on pitch inventories of Zär performances, we showed that our computational approaches can help to make ethnomusicological research on Georgian Zär and possibly other non-Western singing traditions more efficient. Furthermore, our tools open up new ways to explore data collections. To make our tools reusable in future research, we plan to release a publicly available toolbox for computational ethnomusicology. Furthermore, in close collaboration with ethnomusicologists, we will further explore and expand our toolbox for tonal analysis of the complete GVM collection.

9 Summary and Future Work

In this thesis, we developed computational tools that can be used in an interactive fashion for analyzing multitrack recordings of polyphonic vocal music. To test and evaluate these tools, we considered two concrete musical scenarios.

In Part I, we addressed Western choral music. The lack of suitable publicly available multitrack research corpora on choral singing motivated us to record and create Dagstuhl ChoirSet (Chapter 3). In two case studies on intonation analysis and multiple-F0 estimation, we showed that the different musical and acoustical dimensions of DCS open up a variety of scenarios for MIR research. As one technical contribution of this thesis, we formalized a method for applying time-varying pitch shifts to audio signals using non-linear TSM and resampling techniques (Chapter 4). Furthermore, we implemented our approach as part of a publicly available toolbox called `libtsm`. Using DCS recordings as an application scenario, we showed that our adaptive pitch-shifting approach is a powerful tool to compensate local and global intonation deviations in recordings of polyphonic singing. Additionally, our work has set the foundation for further research on intonation processing in choral singing, such as for developing a differentiable intonation cost measure [185].

In Part II, we considered traditional Georgian vocal music. In this context, we have curated a corpus of historic tape recordings of the former master chanter Artem Erkomaishvili (Chapter 5). The carefully annotated and organized corpus constitutes a vital basis for studying traditional Georgian singing as well as MIR tasks such as F0-estimation, source separation, and score following. By providing public access to the Erkomaishvili corpus via an interactive web-based interface, we contributed to preserving and disseminating the rich yet endangered cultural heritage. As further technical contributions of this thesis, we formalized, implemented, and experimentally validated interactive signal processing tools for analyzing multitrack singing voice recordings. First, we developed two approaches for detecting stable regions in F0-trajectories based on morphological filters and binary masks (Chapter 6). The two approaches perform equally well with respect to reference annotations and constitute an important tool for tonal analysis of traditional Georgian vocal music. Second, we developed three indicators that fuse the outputs of several F0-estimation algorithms for assessing the reliability of automatically extracted F0-estimates (Chapter 7). We evaluated the behavior of our indicators on a set of manually annotated vocal F0-trajectories and showed their potential for analyzing large unlabeled audio collections such as the GVM collection (see Section 7.2). Third, through close collaboration with ethnomusicologists, we developed interactive computational tools for identifying note-like events in F0-trajectories as well as measuring and compensating pitch drifts in F0-trajectories through interval-based filtering techniques (Chapter 8). In the context of a case study on Zär recordings, we demonstrate how domain experts can interactively apply these tools to obtain drift-corrected stable note-like events as required for subsequent tonal analysis.

In summary, this thesis demonstrated that interactive computational tools and suitable feedback mechanisms (e.g., visualizations) can substantially support interdisciplinary research on polyphonic singing. Following good scientific practices for transparent, reproducible, and sustainable research [113], we made our corpora and accompanying tools publicly available and accessible. One tangible task for future research consists in applying our tools for tonal analysis of the complete GVM collection. In conjunction with the analysis of Artem Erkomaishvili's recordings, such studies will allow musicologists to gain a more profound understanding of the traditional Georgian tuning system. Using Georgian singing and other (non-Western) vocal music traditions as application scenarios, one may further expand and improve our computational tools. A promising research direction is the development of hybrid approaches for singing voice analysis that combine classical signal processing concepts (as used in this thesis) with the benefits of recent data-driven techniques [36, 59, 147]. Such methods may learn, e.g., culture-specific scales to improve (multiple-)F0 estimation and intonation processing tasks. In this context, the interdisciplinary exchange with musicologists is crucial to ensure the acceptance, applicability, and interpretability of the developed models (as also shown in Chapter 8). Additionally, for recording polyphonic vocal performances, one may explore different sensor types that overcome some limitations of the larynx microphones used in this thesis. For instance, contact microphones based on piezo sensors of high sensitivity [85] have shown great potential for recording singing voice signals with high acoustic quality and few cross-talk while also capturing a singer's heartbeat. In combination with suitable computational tools, such recording techniques would open up new paths for understanding how singers perform and interact with each other.

Appendix

A A Web-Based Interface for Practicing Choral Parts

This chapter is based on [160]. The first author Sebastian Rosenzweig is the main contributor to this late-breaking demo abstract. Together with his supervisor Meinard Müller, he developed the ideas and wrote the paper. Lukas Dietz implemented the web-based interface under the supervision of Sebastian Rosenzweig. The recordings used for demonstration purposes have been provided by the Carus publishing house.

Choir singers typically practice their choral parts individually in preparation for joint rehearsals. Over the last years, applications have become popular that support individual rehearsals, e.g., with sing-along and score-following functionalities. In the following, we present a web-based interface with real-time intonation feedback for choir rehearsal preparation. The interface combines several open-source tools of the MIR community.

A.1 Introduction

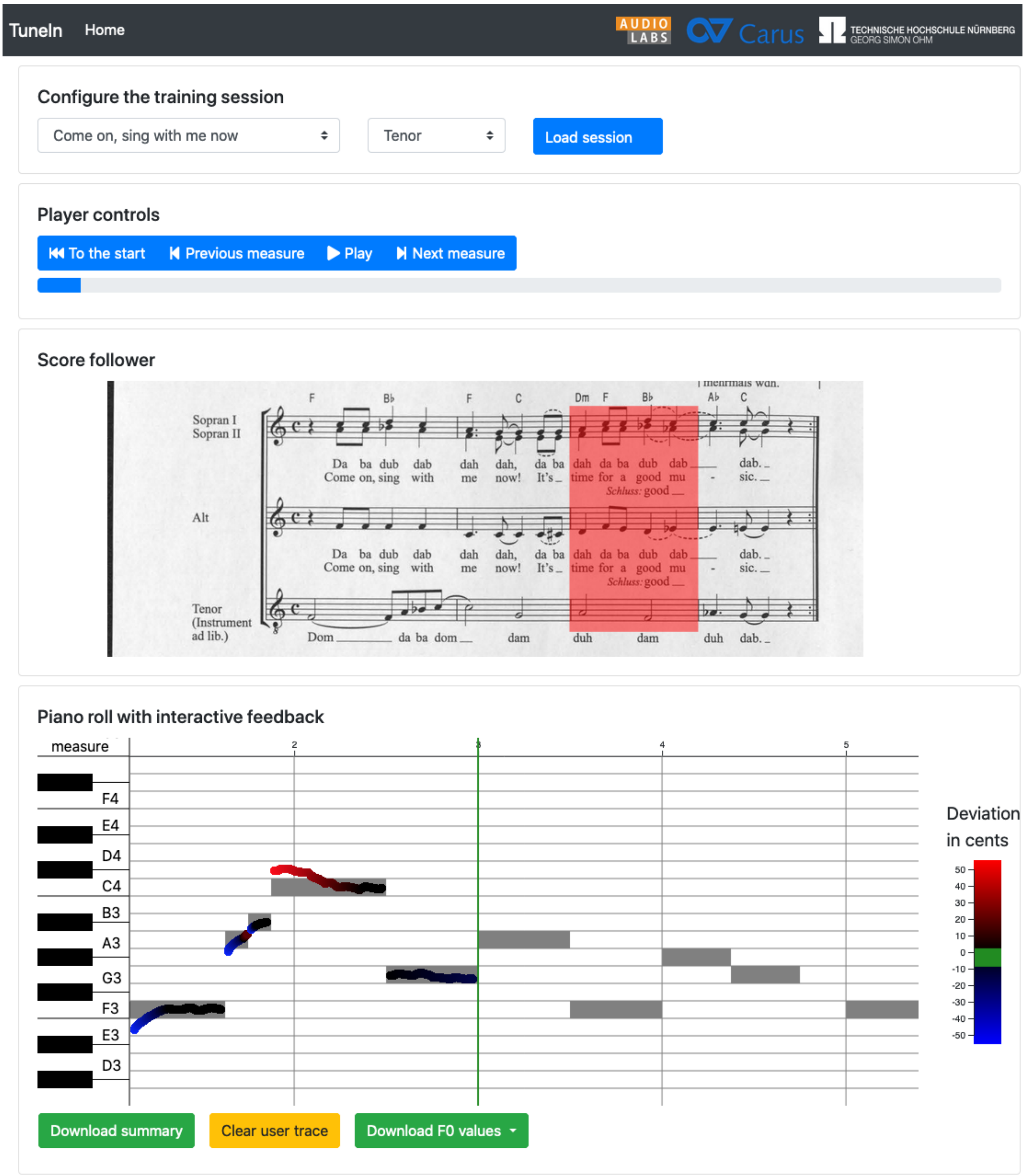
Choirs aim at blending the voices of different choral parts to create a cohesive whole. To this end, choirs spend a significant amount of rehearsal time on improving timing and intonation. Since joint rehearsal time is limited, singers often need to practice their parts individually (e.g., at home) as preparation for the rehearsals. However, individual rehearsals face several practical limitations due to the lack of fellow singers, interaction, and feedback.

Over the last years, interactive applications that support choristers in individual rehearsals have become popular. Three popular commercial examples are *Singerhood*[56], *cantāmus*[57], and *carus music*[58]. The general concept of these apps is similar: after selecting a piece, the user can sing along to a choir recording of the selected piece while reading its score on the screen. *Singerhood* includes multitrack choir recordings

[56] https://singerhood.com

[57] https://cantamus.app/

[58] https://www.carus-verlag.com/en/digital-media/carus-music-the-choir-app



Configure the training session

Come on, sing with me now

Tenor

Figure A.1: Web-based interface for practicing choral parts on the example of the piece "Come on, sing with me now" composed by Werner Rizzi, which is part of the Carus songbook *SingSangSong III*.

and allows for adjusting the volumes of different choral parts for playback. *cantāmus* offers synthesized singing voice accompaniments generated from uploaded scores. *carus music* is based on scores of music editions by the Carus publishing house and includes a score following functionality (music-synchronous highlighting of notes in the score during playback). Furthermore, *carus music* offers a piano playback of the chosen choral part as "coach" for the singer.

Similar to these applications, we have developed a web-based interface to support choir singers during individual rehearsals. Beyond playback and score following functionalities, our interface provides real-time feedback on the singer's intonation. Our interface combines several open-source tools that have been developed by the MIR community. For demonstration purposes, we use recordings and sheet music of choral pieces with piano accompaniment from the Carus music editions. The pieces are arranged for vocal training with children and youth.[59]

A.2 Technical Realization

Our web-based interface, called "TuneIn", can be accessed via the following link:

`https://www.audiolabs-erlangen.de/resources/MIR/TuneIn`

The interface is organized into different modules (see Figure A.1). In the first module, the user can configure the training session by choosing a piece and a choral part. The second module contains an HTML5 audio player with controls and a progress bar that plays back a mix of all voices. The third module contains a score follower. We use the tool proposed in [222], which displays a digital scan of the sheet music and highlights the currently played back measure. To this end, we annotated measure positions in the audio recordings and the scanned sheet music (in the form of bounding boxes given in pixel positions).

As the main feature, which is often not offered by other rehearsal applications, the fourth module includes a piano roll representation that indicates real-time intonation feedback during the singer's performance. The piano roll shows the notes of the chosen part obtained from a music XML version of the score. The representation is synchronized with the audio recording using beat annotations. When the user sings along to the playback, the interface records the singers' voice and estimates its F0 in real-time using CREPE (see Section 2.3.5 and [97]) and the JavaScript library *tensorflow.js*[60]. Furthermore, we determine deviations in cents of the estimated F0-values from the MIDI center frequency. Since the piano accompaniment prevents the choir from drifting in intonation in the chosen recordings, the computed deviations can serve as an indicator for the intonation quality of the singer's performance. The singer's F0-trajectory and the color-coded deviations are visualized superimposed with the piano roll representation (red: positive deviation, blue: negative deviation). The singer can download the performance as an image or the estimated F0-trajectory as a CSV file for later analysis.

[59] `https://www.carus-verlag.com/en/focus/singing-with-children-and-young-people/`

[60] `https://www.tensorflow.org/js`

A.3 Conclusions

Seen individually, the utilized tools are not novel. However, their combination to a web-based, platform-independent interface with intonation feedback can be beneficial for choir singers during individual rehearsals. The modular structure of the interface and the usage of open source tools simplifies expanding functionality and repertoire in future work. Furthermore, our interface can serve as a starting point for exploring different (multitrack) audio players and score following techniques as well as a platform for interactive evaluation of F0-estimation algorithms.

Abbreviations

12-TET	Twelve-Tone Equal Temperament
AMT	Automatic Music Transcription
CMNDF	Cumulative Mean Normalized Difference Function
CPDL	Choral Public Domain Library
CSD	Choral Singing Dataset
CSV	Comma-Separated Values
DAW	Digital Audio Workstation
DCS	Dagstuhl ChoirSet
DFG	Deutsche Forschungsgemeinschaft
DFT	Discrete Fourier Transform
DOI	Digital Object Identifier
DYN	Dynamic Microphone
F0	Fundamental Frequency
FT	Fourier Transform
FFT	Fast Fourier Transform
GCH	Georgian Chant Hymns
GMM	Gaussian Mixture Model
GVM	Georgian Vocal Music
HDS	Headset Microphone
HPS	Harmonic–Percussive Separation
HMM	Hidden Markov Model
HSM	Headset Microphone
IC	Intonation Cost
ID	Identifier
IF	Instantaneous Frequency
LRX	Larynx/Throat Microphone
MEI	Music Encoding Initiative
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
OLA	Overlap-Add
OMR	Optical Music Recognition
ORTF	Office de Radiodiffusion Télévision Française
QNR	Quarter Note Reference
SATB	Soprano, Alto, Tenor, Bass
STFT	Short-Time Fourier Transform
TROMPA	Towards Richer Online Music Public-Domain Archives
TSM	Time-Scale Modification
UNESCO	United Nations Educational, Scientific and Cultural Organization
WAB	Werkverzeichnis Anton Bruckner
WSOLA	Waveform-Similarity Overlap-Add

Bibliography

[1] Per-Gunnar Alldahl. *Choral Intonation*. Gehrmans Musikförlag, 2008. ISBN 9789177482567.

[2] Rytis Ambrazevičius, Robertas Budrys, and Irena Višnevska. *Scales in Lithuanian traditional music: Acoustics, cognition, and contexts*. Arx reklama, 2015.

[3] Andreas Arzt. *Flexible and Robust Music Tracking*. PhD thesis, Universität Linz, 2016.

[4] Stefan Balke, Jakob Abeßer, Jonathan Driedger, Christian Dittmar, and Meinard Müller. Towards evaluating multiple predominant melody annotations in jazz recordings. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 246–252, New York City, New York, USA, 2016. doi: 10.5281/zenodo.1415076.

[5] Judith Bauer. Deep-learning approaches for fundamental frequency estimation of music recordings. Master Thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2019.

[6] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3): 407–434, 2013. doi: 10.1007/s10844-013-0258-3.

[7] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, 36(1):20–30, 2019. doi: 10.1109/MSP.2018.2869928.

[8] Jesse Berezovsky. The structure of musical harmony as an ordered phase of sound: A statistical mechanics approach to music theory. *Science Advances*, 5(5):eaav8490, 2019. doi: 10.1126/sciadv.aav8490.

[9] Rachel M. Bittner. *Data-Driven Fundamental Frequency Estimation*. PhD thesis, New York University, 2018.

[10] Rachel M. Bittner and Juan J. Bosch. Generalized metrics for single-f0 estimation evaluation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 738–745, Delft, The Netherlands, 2019. doi: 10.5281/zenodo.3527916.

[11] Rachel M. Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. MedleyDB: A multitrack dataset for annotation-intensive MIR research. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 155–160, Taipei, Taiwan, 2014. doi: 10.5281/zenodo.1417889.

[12] Rachel M. Bittner, Eric Humphrey, and Juan P. Bello. PySox: Leveraging the audio signal processing power of SoX in Python. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016.

[13] Rachel M. Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan P. Bello. Deep salience representations for F0 tracking in polyphonic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 63–70, Suzhou, China, 2017. doi: 10.5281/zenodo.1417937.

[14] Rachel M. Bittner, Justin Salamon, Juan J. Bosch, and Juan Pablo Bello. Pitch contours as a mid-level representation for music informatics. In *Proceedings of the AES International Conference on Semantic Audio*, pages 100–107, Erlangen, Germany, 2017. URL `http://www.aes.org/e-lib/browse.cfm?elib=18756`.

[15] Rachel M. Bittner, Magdalena Fuentes, David Rubinstein, Andreas Jansson, Keunwoo Choi, and Thor Kell. mirdata: Software for reproducible usage of datasets. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 99–106, Delft, The Netherlands, 2019. doi: 10.5281/zenodo.3527750.

[16] Merlijn Blaauw and Jordi Bonada. A neural parametric singing synthesizer modeling timbre and expression from natural songs. *Applied Sciences*, 7(1313), 2017. doi: 10.3390/app7121313.

[17] Sebastian Böck, Florian Krebs, and Markus Schedl. Evaluating the online capabilities of onset detection methods. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 49–54, Porto, Portugal, 2012. doi: 10.5281/zenodo.1416035.

[18] Sebastian Böck, Matthew E. P. Davies, and Peter Knees. Multi-task learning of tempo and beat: Learning one to improve the other. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 486–493, Delft, The Netherlands, 2019.

[19] Paul Boersma. Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10):341–345, 2001.

[20] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, José R. Zapata, and Xavier Serra. Essentia: An audio analysis library for music information retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 493–498, Curitiba, Brazil, 2013. doi: 10.5281/zenodo.1415016.

[21] Ken Bogdanowicz and Robert Belcher. Using multiple processors for real-time audio effects. In *Audio Engineering Society Conference*. Audio Engineering Society, 1989.

[22] Juan J. Bosch and Emilia Gómez. Melody extraction in symphonic classical music: a comparative study of mutual agreement between humans and algorithms. In *Proceedings of the Conference on Interdisciplinary Musicology (CIM)*, 2014.

[23] Robert Bristow-Johnson. A detailed analysis of a time-domain formant-corrected pitch-shifting algorithm. In *Audio Engineering Society Convention*. Audio Engineering Society, 1993.

[24] Paul Brossier, Juan Pablo Bello, and Mark Plumbley. Fast labelling of notes in music signals. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Barcelona, Spain, 2004. doi: 10.5281/zenodo.1416132.

[25] Paul Brossier, Juan Pablo Bello, and Mark Plumbley. Real-time temporal segmentation of note objects in music signals. In *Proceedings of the International Computer Music Conference (ICMC)*, Florida, USA, 2004.

[26] Donald Byrd and Jakob G. Simonsen. Towards a standard testbed for optical music recognition: Definitions, metrics, and page images. *Journal of New Music Research*, 44(3):169–195, 2015. doi: 10.1080/09298215.2015.1045424.

[27] Arturo Camacho and John G. Harris. A sawtooth waveform inspired pitch estimator for speech and music. *The Journal of the Acoustical Society of America*, 124(3):1638–1652, 2008.

[28] Chris Cannam, Michael O. Jewell, Christophe Rhodes, Mark Sandler, and Mark d'Inverno. Linked data and you: Bringing music research software into the semantic web. *Journal of New Music Research*, 39(4): 313–325, 2010. doi: 10.1080/09298215.2010.522715.

[29] Chris Cannam, Christian Landone, and Mark B. Sandler. Sonic Visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the International Conference on Multimedia*, pages 1467–1468, Florence, Italy, 2010.

[30] Estefanía Cano, Gerald Schuller, and Christian Dittmar. Pitch-informed solo and accompaniment separation towards its use in music education applications. *EURASIP Journal on Advances in Signal Processing*, 2014 (23), 2014. doi: 10.1186/1687-6180-2014-23.

[31] Estefanía Cano, Derry FitzGerald, Antoine Liutkus, Mark D. Plumbley, and Fabian-Robert Stöter. Musical source separation: An introduction. *IEEE Signal Processing Magazine*, 36(1):31–40, 2019. doi: 10.1109/MSP.2018.2874719.

[32] Pritish Chandna, Merlijn Blaauw, Jordi Bonada, and Emilia Gómez. A vocoder based method for singing voice extraction. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019. IEEE. doi: 10.1109/ICASSP.2019.8683323.

[33] Francis Charpentier and M. G. Stella. Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2015–2018, Tokyo, Japan, 1986. IEEE. doi: 10.1109/ICASSP.1986.1168657.

[34] Evsevi Chokhonelidze. Some characteristic features of the voice coordination and harmony in Georgian multipart singing. In *Echoes from Georgia: Seventeen Arguments on Georgian Polyphony*, pages 135–145. Nova Science Publishers, 2010.

[35] Georgios Chrysochoidis, Georgios Kouroupetroglou, and Sergios Theodoridis. Vibrato detection in byzantine chant music. In *International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pages 636–639, Athens, Greece, 2014.

[36] Joseph T. Colonel and Joshua Reiss. Reverse engineering of a recording mix with differentiable digital signal processing. *The Journal of the Acoustical Society of America*, 150(1):608–619, 2021. doi: 10.1121/10.0005622.

[37] Bas Cornelissen, Willem H. Zuidema, and John Ashley Burgoyne. Cosine contours: a multipurpose representation for melodies. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 135–142, online, 2021. doi: 10.5281/zenodo.5624531.

[38] Albin Correya, Dmitry Bogdanov, Luis Joglar-Ongay, and Xavier Serra. Essentia.js: A JavaScript library for music and audio analysis on the web. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 605–612, Montréal, Canada, 2020. doi: 10.5281/zenodo.4245510.

[39] Helena Cuesta. *Data-driven Pitch Content Description of Choral Singing Recordings*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2022. under review.

[40] Helena Cuesta, Emilia Gómez, Agustín Martorell, and Felipe Loáiciga. Analysis of intonation in unison choir singing. In *Proceedings of the International Conference of Music Perception and Cognition (ICMPC)*, pages 125–130, Graz, Austria, 2018.

[41] Helena Cuesta, Emilia Gómez, and Pritish Chandna. A framework for multi-f0 modeling in SATB choir recordings. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 447–453, Málaga, Spain, 2019.

[42] Helena Cuesta, Brian McFee, and Emilia Gómez. Multiple F0 estimation in vocal ensembles using convolutional neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 302–309, Montréal, Canada, 2020. doi: 10.5281/zenodo.4245434.

[43] Michael Scott Cuthbert and Christopher Ariza. Music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 637–642, Utrecht, The Netherlands, 2010. doi: 10.5281/zenodo.1416114.

[44] Andrzej Czyzewski, Marek Dziubinski, Andrzej Ciarkowski, Maciej Kulesza, Przemyslaw Maziewski, and Jozef Kotus. New algorithms for wow and flutter detection and compensation in audio. *Proceedings of the Audio Engineering Society (AES) Convention*, 6353, 2005.

[45] Andrzej Czyzewski, Przemyslaw Maziewski, and Adam Kupryjanow. Reduction of parasitic pitch variations in archival musical recordings. *Signal Processing*, 90(4):981–990, 2010.

[46] Jiajie Dai and Simon Dixon. Analysis of interactive intonation in unaccompanied SATB ensembles. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 599–605, Suzhou, China, 2017. doi: 10.5281/zenodo.1418327.

[47] Jiajie Dai and Simon Dixon. Singing together: Pitch accuracy and interaction in unaccompanied unison and duet singing. *Journal of the Acoustical Society of America (JASA)*, 145(2):663–675, 2019.

[48] Alain de Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America (JASA)*, 111(4):1917–1930, 2002.

[49] Boyuan Deng, Denis Jouvet, Yves Laprie, Ingmar Steiner, and Aghilas Sini. Towards confidence measures on fundamental frequency estimations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5605–5609, New Orleans, Louisiana, USA, 2017. IEEE. doi: 10.1109/ICASSP.2017.7953229.

[50] Johanna Devaney. *An Empirical Study of the Influence of Musical Context on Intonation Practices in Solo Singers and SATB Ensembles*. PhD thesis, McGill University, Montreal, Canada, 2011.

[51] Johanna Devaney and Daniel P. W. Ellis. An empirical approach to studying intonation tendencies in polyphonic vocal performances. *Journal of Interdisciplinary Music Studies*, 2(1&2):141–156, 2008.

[52] Johanna Devaney, Michael I. Mandel, and Ichiro Fujinaga. A study of intonation in three-part singing using the automatic music performance analysis and comparison toolkit (AMPACT). In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 511–516, Porto, Portugal, 2012. doi: 10.5281/zenodo.1416210.

[53] José Miguel Díaz-Báñez and Juan-Carlos Rizo. An efficient DTW-based approach for melodic similarity in flamenco singing. In *International Conference on Similarity Search and Applications*, pages 289–300, 2014.

[54] Mark Dolson. The phase vocoder: A tutorial. *Computer Music Journal*, 10(4):14–27, 1986.

[55] Jonathan Driedger and Meinard Müller. TSM Toolbox: MATLAB implementations of time-scale modification algorithms. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 249–256, Erlangen, Germany, 2014.

[56] Jonathan Driedger and Meinard Müller. A review on time-scale modification of music signals. *Applied Sciences*, 6(2):57–82, 2016. doi: 10.3390/app6020057.

[57] Jonathan Driedger, Meinard Müller, and Sebastian Ewert. Improving time-scale modification of music signals using harmonic–percussive separation. *IEEE Signal Processing Letters*, 21(1):105–109, 2014.

[58] Georgi Dzhambazov, Ajay Srinivasamurthy, Sertan Sentürk, and Xavier Serra. On the use of note onsets for improved lyrics-to-audio alignment in Turkish makam music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 716–722, New York City, New York, USA, 2016. doi: 10.5281/zenodo.1284501.

[59] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. DDSP: Differentiable digital signal processing. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL `https://openreview.net/forum?id=B1x1ma4tDr`.

[60] Malkhaz Erkvanidze. The Georgian musical system. In *Proceedings of the International Workshop on Folk Music Analysis (FMA)*, pages 74–79, Dublin, Ireland, 2016.

[61] Sebastian Ewert, Meinard Müller, and Peter Grosche. High resolution audio synchronization using chroma onset features. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1869–1872, Taipei, Taiwan, 2009. doi: 10.1109/ICASSP.2009.4959972.

[62] Derry FitzGerald. Harmonic/percussive separation using median filtering. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 246–253, Graz, Austria, 2010.

[63] J. L. Flanagan and R. M. Golden. Phase vocoder. *Bell System Technical Journal*, 45:1493–1509, 1966.

[64] Dennis Gabor. Theory of communication. *Journal of the Institution of Electrical Engineers (IEE)*, 93(26): 429–457, 1946.

[65] Kaustuv Kanti Ganguli and Preeti Rao. On the distributional representation of ragas: experiments with allied raga pairs. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 1(1):79–95, 2018. doi: 10.5334/tismir.11.

[66] Martin Gasser, Andreas Arzt, Thassilo Gadermaier, Maarten Grachten, and Gerhard Widmer. Classical music on the web – user interfaces and data representations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 571–577, Málaga, Spain, 2015. doi: 10.5281/zenodo.1417717.

[67] Ali C. Gedik and Barış Bozkurt. Pitch-frequency histogram-based music information retrieval for Turkish music. *Signal Processing*, 90(4):1049–1063, 2010.

[68] Beat Gfeller, Christian Frank, Dominik Roblek, Matthew Sharifi, Marco Tagliasacchi, and Mihajlo Velimirovic. SPICE: Self-supervised pitch estimation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1118–1128, 2020. doi: 10.1109/TASLP.2020.2982285.

[69] Volker Gnann, Markus Kitza, Julian Becker, and Martin Spiertz. Least-squares local tuning frequency estimation for choir music. In *Proceedings of the Audio Engineering Society (AES) Convention*, New York City, New York, USA, 2011.

[70] Simon Godsill, Peter Rayner, and Olivier Cappé. Digital audio restoration. In *Applications of Digital Signal Processing to Audio and Acoustics*, pages 133–194. Springer, 2002.

[71] Simon J. Godsill and P. J. W. Rayner. The restoration of pitch variation defects in gramophone recordings. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 148–151, New Paltz, New York, USA, 1993.

[72] Emilia Gómez and Jordi Bonada. Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to A cappella singing. *Computer Music Journal*, 37(2):73–90, 2013.

[73] Emilia Gómez, Perfecto Herrera, and Francisco Gómez-Martin. Computational ethnomusicology: Perspectives and challenges. *Journal of New Music Research*, 42(2):111–112, 2013. doi: 10.1080/09298215.2013.818038.

[74] Rong Gong, Rafael Caro Repetto, and Xavier Serra. Creating an A cappella singing audio dataset for automatic jingju singing evaluation research. In *Proceedings of the International Workshop on Digital Libraries for Musicology*, pages 37–40, 2017.

[75] Masataka Goto and Yoichi Muraoka. Issues in evaluating beat tracking systems. In *Working Notes of the IJCAI-97 Workshop on Issues in AI and Music-Evaluation and Assessment*, pages 9–16, 1997.

[76] Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the Audio Engineering Society (AES) International Conference*, London, UK, 2004.

[77] Matan Gover and Phillipe Depalle. Score-informed source separation of choral music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 231–239, Montréal, Canada, 2020. doi: 10.5281/zenodo.4245412.

[78] John Graham. *The Transcription and Transmission of Georgian Liturgical Chant*. PhD thesis, Princeton University, 2015.

[79] Anke Grell, Johan Sundberg, Sten Ternström, Martin Ptok, and Eckart Altenmüller. Rapid pitch correction in choir singers. *The Journal of the Acoustical Society of America*, 126(1):407–413, 2009.

[80] Louis Grijp and Ineke van Beersum. Under the green linden. 163 dutch ballads from the oral tradition recorded by ate doornbosch. 2008.

[81] Azadeh Haghparast, Henri Penttinen, and Vesa Välimäki. Real-time pitch-shifting of musical signals by a time-varying factor using normalized filtered correlation time-scale modification. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 7–14, Bordeaux, France, 2007.

[82] Christopher Harte, Mark B. Sandler, Samer Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 66–71, London, UK, 2005. doi: 10.5281/zenodo.1415114.

[83] Frank Havrøy. 'You Cannot Just Say: "I am Singing the Right Note"'. Discussing intonation issues with Neue Vocalsolisten Stuttgart. *Music & Practice*, 1, 2013. doi: 10.32063/0104.

[84] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software (JOSS)*, 5(50):2154, 2020. doi: 10.21105/joss.02154.

[85] Tatsuya Hirahara, Shota Shimizu, and Makoto Otani. Acoustic characteristics of non-audible murmur. In *The Japan China Joint Conference of Acoustics*, volume 100, page 4000, 2007.

[86] David M. Howard. Intonation drift in a capella soprano, alto, tenor, bass quartet singing with key modulation. *Journal of Voice*, 21(3):300–315, 2007. doi: 10.1016/j.jvoice.2005.12.005.

[87] David M. Howard, Helena Daffern, and Jude Brereton. Four-part choral synthesis system for investigating intonation in a cappella choral singing. *Logopedics Phoniatrics Vocology*, 38(3):135–142, 2013. doi: 10.3109/14015439.2013.812143.

[88] Eric J. Humphrey, Sravana Reddy, Prem Seetharaman, Aparna Kumar, Rachel M. Bittner, Andrew Demetriou, Sankalp Gulati, Andreas Jansson, Tristan Jehan, Bernhard Lehner, Anna Krupse, and Luwei Yang. An introduction to signal processing for singing-voice analysis: High notes in the effort to automate the understanding of vocals in music. *IEEE Signal Processing Magazine*, 36(1):82–94, 2019. doi: 10.1109/MSP.2018.2875133.

[89] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 448–456, Lille, France, 2015.

[90] Vignesh Ishwar, Shrey Dutta, Ashwin Bellur, and Hema A. Murthy. Motif spotting in an alapana in carnatic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 499–504, Curitiba, Brazil, 2013. doi: 10.5281/zenodo.1416332.

[91] Dasaem Jeong, Taegyun Kwon, Chaelin Park, and Juhan Nam. PerformScore: Toward performance visualization with the score on the web browser. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017.

[92] Ilia Jgharkava. Pearls of Georgian Chant. CD, 2016. produced by the Georgian Chanting Foundation & Tbilisi State Conservatoire.

[93] Joseph Jordania. *Who Asked the First Question? The Origins of Human Choral Singings, Intelligence, Language and Speech.* Tbilisi State University Press, Tbilisi, Georgia, 2006.

[94] Joseph Jordania. Why do people sing? music in human evolution. *Music in human evolution*, 1, 2011.

[95] Nino Kalandadze-Makharadze. The funeral Zari in traditional male polyphony. In *Proceedings of the International Symposium on Traditional Polyphony*, pages 166–178, Tbilisi, Georgia, 2004.

[96] Richard Keeling. Contrast of song performance style as a function of sex role polarity in the Hupa Brush Dance. *Ethnomusicology*, pages 185–212, 1985.

[97] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. CREPE: A convolutional representation for pitch estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165, Calgary, Canada, 2018. doi: 10.1109/ICASSP.2018.8461329.

[98] Anssi P. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 216–221, Victoria, BC, Canada, 2006. doi: 10.5281/zenodo.1416740.

[99] Anssi P. Klapuri. Multipitch analysis of polyphonic music and speech signals using an auditory model. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):255–266, 2008. doi: 10.1109/TASL.2007.908129.

[100] Anssi P. Klapuri and Manuel Davy, editors. *Signal Processing Methods for Music Transcription.* Springer, New York, 2006. ISBN 0-387-30667-6.

[101] Gopala Krishna Koduri, Sankalp Gulati, Preeti Rao, and Xavier Serra. Rāga recognition based on pitch distribution methods. *Journal of New Music Research*, 41(4):337–350, 2012. doi: 10.1080/09298215.2012.735246.

[102] Gopala Krishna Koduri, Joan Serrà, and Xavier Serra. Characterization of intonation in carnatic music by parametrizing pitch histograms. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 199–204, Porto, Portugal, 2012. doi: 10.5281/zenodo.1416902.

[103] Nadine Kroher and Emilia Gómez. Automatic transcription of flamenco singing from polyphonic music recordings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):901–913, 2016. doi: 10.1109/TASLP.2016.2531284.

[104] Nadine Kroher, José Miguel Díaz-Báñez, Joaquín Mora, and Emilia Gómez. Corpus COFLA: A research corpus for the computational study of flamenco music. *Journal on Computing and Cultural Heritage (JOCCH)*, 9(2):10:1–10:21, 2016.

[105] Jiei Kuroyanagi, Shoichiro Sato, Meng-Jou Ho, Gakuto Chiba, Joren Six, Peter Pfordresher, Adam Tierney, Shinya Fujii, and Patrick Savage. Automatic comparison of human music, speech, and bird song suggests uniqueness of human scales. In *Folk Music Analysis Conference*, pages 35–40, 2019.

[106] Olivier Lartillot and Petri Toiviainen. MIR in MATLAB (II): A toolbox for musical feature extraction from audio. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 127–130, Vienna, Austria, 2007. doi: 10.5281/zenodo.1417145.

[107] Edith L. M. Law, Luis von Ahn, Roger B. Dannenberg, and Mike Crawford. TagATune: A game for music and sound annotation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 361–364, Vienna, Austria, 2007. doi: 10.5281/zenodo.1415568.

[108] Keith Lent. An efficient method for pitch shifting digitally sampled sounds. *Computer Music Journal*, 13(4): 65–71, 1989.

[109] Matthias Mauch and Simon Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, Florence, Italy, 2014. doi: 10.1109/ICASSP.2014.6853678.

[110] Matthias Mauch, Klaus Frieler, and Simon Dixon. Intonation in unaccompanied singing: Accuracy, drift, and a model of reference pitch memory. *Journal of the Acoustical Society of America*, 136(1):401–411, 2014. doi: 10.1121/1.4881915.

[111] Matthias Mauch, Chris Cannam, Rachel Bittner, George Fazekas, Justing Salamon, Jiajie Dai, Juan Bello, and Simon Dixon. Computer-aided melody note transcription using the Tony software: Accuracy and efficiency. In *Proceedings of the International Conference on Technologies for Music Notation and Representation*, 2015.

[112] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. Librosa: Audio and music signal analysis in Python. In *Proceedings the Python Science Conference*, pages 18–25, Austin, Texas, USA, 2015. doi: 10.25080/Majora-7b98e3ed-003.

[113] Brian McFee, Jong Wook Kim, Mark Cartwright, Justin Salamon, Rachel M. Bittner, and Juan Pablo Bello. Open-source practices for music signal processing research: Recommendations for transparent, sustainable, and reproducible audio research. *IEEE Signal Processing Magazine*, 36(1):128–137, 2019. doi: 10.1109/MSP.2018.2875349.

[114] Mervyn McLean. A preliminary analysis of 87 maori chants. *Ethnomusicology*, 8(1):41–48, 1964.

[115] Andrew McLeod, Rodrigo Schramm, Mark Steedman, and Emmanouil Benetos. Automatic transcription of polyphonic vocal music. *Applied Sciences*, 7(12), 2017. doi: 10.3390/app7121285.

[116] Blai Meléndez-Catalán, Emilio Molina, and Emilia Gómez. BAT: An open-source, web-based audio events annotation tool. In *Web Audio Conference*, 2017.

[117] Dirk Moelants, Olmo Cornelis, and Marc Leman. Exploring African tone scales. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 489–494, Kobe, Japan, 2009. doi: 10.5281/zenodo.1416338.

[118] Emilio Molina, Lorenzo J. Tardón, Ana M. Barbancho, and Isabel Barbancho. Sipth: Singing transcription based on hysteresis defined on the pitch-time curve. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):252–263, 2015. doi: 10.1109/TASLP.2014.2331102.

[119] Eric Moulines and Jean Laroche. Non-parametric techniques for pitch-scale and time-scale modification of speech. *Speech Communication*, 16(2):175–205, 1995.

[120] Meinard Müller. *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*. Springer Verlag, 2nd edition, 2021. ISBN 978-3-030-69807-2. doi: 10.1007/978-3-030-69808-9.

[121] Meinard Müller and Sebastian Rosenzweig. PCP notebooks: A preparation course for Python with a focus on signal processing. *Journal of Open Source Education (JOSE)*, 5(47):148:1–5, 2022. doi: 10.21105/jose.00148.

[122] Meinard Müller and Frank Zalkow. FMP Notebooks: Educational material for teaching and learning fundamentals of music processing. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 573–580, Delft, The Netherlands, 2019. doi: 10.5281/zenodo.3527872.

[123] Meinard Müller, Frank Kurth, and Tido Röder. Towards an efficient algorithm for automatic score-to-audio synchronization. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 365–372, Barcelona, Spain, 2004. doi: 10.5281/zenodo.1416302.

[124] Meinard Müller, Peter Grosche, and Frans Wiering. Robust segmentation and annotation of folk song recordings. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 735–740, Kobe, Japan, 2009. doi: 10.5281/zenodo.1417099.

[125] Meinard Müller, Peter Grosche, and Frans Wiering. Automated analysis of performance variations in folk song recordings. In *Proceedings of the International Conference on Multimedia Information Retrieval (MIR)*, pages 247–256, Philadelphia, Pennsylvania, USA, 2010. doi: 10.1145/1743384.1743429.

[126] Meinard Müller, Sebastian Rosenzweig, Jonathan Driedger, and Frank Scherbaum. Interactive fundamental frequency estimation with applications to ethnomusicological research. In *Proceedings of the AES International Conference on Semantic Audio*, pages 186–193, Erlangen, Germany, 2017.

[127] Meinard Müller, Andreas Arzt, Stefan Balke, Matthias Dorfer, and Gerhard Widmer. Cross-modal music retrieval and applications: An overview of key methodologies. *IEEE Signal Processing Magazine*, 36(1): 52–62, 2019. doi: 10.1109/MSP.2018.2868887.

[128] Meinard Müller, Emilia Gómez, and Yi-Hsuan Yang. Computational methods for melody and voice processing in music recordings (Dagstuhl seminar 19052). *Dagstuhl Reports*, 9(1):125–177, 2019. doi: 10.4230/DagRep.9.1.125.

[129] Nana Mzhavanadze and Frank Scherbaum. Svan funeral dirges (Zär): Musicological analysis. *Musicologist*, 4:168–197, 2020. doi: 10.33906/musicologist.782185.

[130] Nana Mzhavanadze and Frank Scherbaum. Svan funeral dirges (Zär): Cultural context. *Musicologist*, 5: 133–165, 2021. doi: 10.33906/musicologist.906765.

[131] Ryo Nishikimi, Eita Nakamura, Masataka Goto, Katsutoshi Itoyama, and Kazuyoshi Yoshii. Scale- and rhythm-aware musical note estimation for vocal F0 trajectories based on a semi-tatum-synchronous hierarchical hidden semi-markov model. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 376–382, Suzhou, China, 2017. doi: 10.5281/zenodo.1416330.

[132] Yuto Ozaki, John M. McBride, Emmanouil Benetos, Peter Pfordresher, Joren Six, Adam Tierney, Polina Proutskova, Emi Sakai, Haruka Kondo, Haruno Fukatsu, Shinya Fujii, and Patrick E. Savage. Agreement among human and automated transcriptions of global songs. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 500–508, online, 2021. doi: 10.5281/zenodo.5624529.

[133] Sachin Pant, Vishweshwara Rao, and Preeti Rao. A melody detection user interface for polyphonic music. In *National Conference on Communications (NCC)*, pages 1–5, Chennai, India, 2010.

[134] Maria Panteli. *Computational Analysis of World Music Corpora*. PhD thesis, Queen Mary University of London, UK, 2018.

[135] Maria Panteli, Rachel M. Bittner, Juan Pablo Bello, and Simon Dixon. Towards the characterization of singing styles in world music. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 636–640, New Orleans, Louisiana, USA, 2017. doi: 10.1109/ICASSP.2017.7952233.

[136] Maria Panteli, Emmanouil Benetos, and Simon Dixon. A review of manual and computational approaches for the study of world music corpora. *Journal of New Music Research*, 47(2):176–189, 2018. doi: 10.1080/09298215.2017.1418896.

[137] Karl Pedersen and Mark Grimshaw-Aagaard. *The Recording, Mixing, and Mastering Reference Handbook*. Oxford University Press, 2018.

[138] Darius Petermann, Pritish Chandna, Helena Cuesta, Jordi Bonada, and Emilia Gómez. Deep learning based source separation applied to choir ensembles. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 733–739, Montreal, Canada, 2020. doi: 10.5281/zenodo.4245536.

[139] 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*, 15(4):1247–1256, 2007. doi: 10.1109/TASL.2006.889797.

[140] M. R. Portnoff. Implementation of the digital phase vocoder using the fast Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(3):243–248, 1976.

[141] Zdenek Prusa and Nicki Holighaus. Phase vocoder done right. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 976–980, Kos, Greece, 2017.

[142] Laurent Pugin, Rodolfo Zitellini, and Perry Roland. Verovio: A library for engraving mei music notation into SVG. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 107–112, Taipei, Taiwan, 2014. doi: 10.5281/zenodo.1417589.

[143] John Radon. Über die bestimmung von funktionen längs gewisser mannigfaltigkeiten. *Berichte über die Verhandlungen der Königlich-Sächsischen Gesellschaft der Wissenschaften zu Leipzig. Mathematisch-Physische Klasse.*, 69:262–277, 1917.

[144] Colin Raffel and Daniel P. W. Ellis. Intuitive analysis, creation and manipulation of MIDI data with pretty_midi. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014.

[145] Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. MIR_EVAL: A transparent implementation of common MIR metrics. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 367–372, Taipei, Taiwan, 2014. doi: 10.5281/zenodo.1416528.

[146] António Ramires, Frederic Font, Dmitry Bogdanov, Jordan B. L. Smith, Yi-Hsuan Yang, Joann Ching, Bo-Yu Chen, Yueh-Kao Wu, Hsu Wei-Han, and Xavier Serra. The freesound loop dataset and annotation tool. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 287–294, Montreal, Canada, 2020. doi: 10.5281/zenodo.4245430.

[147] Marco A. Martínez Ramírez, Oliver Wang, Paris Smaragdis, and Nicholas J. Bryan. Differentiable signal processing with black-box audio effects. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP)*, pages 66–70, Toronto, ON, Canada, 2021. IEEE. doi: 10.1109/ICASSP39728.2021.9415103.

[148] Preeti Rao, Joe Cheri Ross, Kaustuv Kanti Ganguli, Vedhas Pandit, Vignesh Ishwar, Ashwin Bellur, and Hema A. Murthy. Classification of melodic motifs in raga music with time-series matching. *Journal of New Music Research*, 43(1):115–131, 2014.

[149] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R. S. Marcal, Carlos Guedes, and Jaime S. Cardoso. Optical music recognition: State-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, 2012. doi: 10.1007/s13735-012-0004-6.

[150] Lise Regnier and Geoffroy Peeters. Singing voice detection in music tracks using direct voice vibrato detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1685–1688, Taipei, Taiwan, 2009. doi: 10.1109/ICASSP.2009.4959926.

[151] Rafael Caro Repetto and Xavier Serra. Creating a corpus of Jingju (Beijing Opera) music and possibilities for melodic analysis. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 313–318, Taipei, Taiwan, 2014. doi: 10.5281/zenodo.1416030.

[152] Rafael Caro Repetto, Rong Gong, Nadine Kroher, and Xavier Serra. Comparison of the singing style of two jingju schools. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 507–513, Málaga, Spain, 2015. doi: 10.5281/zenodo.1416692.

[153] Rafael Caro Repetto, Niccolò Pretto, Amin Chaachoo, Baris Bozkurt, and Xavier Serra. An open corpus for the computational research of arab-andalusian music. In *Proceedings of the International Conference on Digital Libraries for Musicology*, pages 78–86, Paris, France, 2018.

[154] Jean-François Rivest, Pierre Soille, and Serge Beucher. Morphological gradients. *Journal of Electronic Imaging*, 2(4):326–336, 1993.

[155] Andrew Robertson. Decoding tempo and timing variations in music recordings from beat annotations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 475–480, Porto, Portugal, 2012. doi: 10.5281/zenodo.1416806.

[156] Gerard Roma, Owen Green, and Pierre Alexandre Tremblay. Time scale modification of audio using non-negative matrix factorization. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 1–6, Birmingham, UK, 2019.

[157] Sebastian Rosenzweig. Audio processing techniques for analyzing Georgian vocal music. Master Thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2017.

[158] Sebastian Rosenzweig, Frank Scherbaum, and Meinard Müller. Detecting stable regions in frequency trajectories for tonal analysis of traditional Georgian vocal music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 352–359, Delft, The Netherlands, 2019. doi: 10.5281/zenodo.3527816.

[159] Sebastian Rosenzweig, Helena Cuesta, Christof Weiß, Frank Scherbaum, Emilia Gómez, and Meinard Müller. Dagstuhl ChoirSet: A multitrack dataset for MIR research on choral singing. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 3(1):98–110, 2020. doi: 10.5334/tismir.48.

[160] Sebastian Rosenzweig, Lukas Dietz, Johannes Graulich, and Meinard Müller. TuneIn: A web-based interface for practicing choral parts. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Montreal, Canada, 2020.

[161] Sebastian Rosenzweig, Frank Scherbaum, David Shugliashvili, Vlora Arifi-Müller, and Meinard Müller. Erkomaishvili Dataset: A curated corpus of traditional Georgian vocal music for computational musicology. *Transactions of the International Society for Music Information Retrieval (TISMIR)*, 3(1):31–41, 2020. doi: 10.5334/tismir.44.

[162] Sebastian Rosenzweig, Frank Scherbaum, and Meinard Müller. Reliability assessment of singing voice F0-estimates using multiple algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 261–265, Toronto, Canada, 2021. doi: 10.1109/ICASSP39728.2021.9413372.

[163] Sebastian Rosenzweig, Simon Schwär, Jonathan Driedger, and Meinard Müller. Adaptive pitch-shifting with applications to intonation adjustment in a cappella recordings. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 121–128, Vienna, Austria, 2021.

[164] Sebastian Rosenzweig, Frank Scherbaum, and Meinard Müller. Computer-assisted analysis of field recordings: A case study of Georgian funeral songs. *ACM Journal on Computing and Cultural Heritage (JOCCH)*, 2022. to appear.

[165] Joe Cheri Ross, Vinutha T. P., and Preeti Rao. Detecting melodic motifs from audio for hindustani classical music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 193–198, Porto, Portugal, 2012. doi: 10.5281/zenodo.1417587.

[166] Salim Roucos and Alexander M. Wilgus. High quality time-scale modification for speech. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 10, pages 493–496, Tampa, FL, USA, 1985. doi: 10.1109/ICASSP.1985.1168381.

[167] Daniel Röwenstrunk, Thomas Prätzlich, Thomas Betzwieser, Meinard Müller, Gerd Szwillus, and Joachim Veit. Das Gesamtkunstwerk Oper aus Datensicht – Aspekte des Umgangs mit einer heterogenen Datenlage im BMBF-Projekt "Freischütz Digital". *Datenbank-Spektrum*, 15(1):65–72, 2015. doi: 10.1007/s13222-015-0179-0.

[168] Matti Ryynänen and Anssi P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008.

[169] Justin Salamon and Emilia Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012. doi: 10.1109/TASL.2012.2188515.

[170] Justin Salamon, Emilia Gómez, Daniel P. W. Ellis, and Gaël Richard. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, 2014. doi: 10.1109/MSP.2013.2271648.

[171] Saurjya Sarkar, Emmanouil Benetos, and Mark Sandler. Vocal harmony separation using time-domain neural networks. In *Proceedings of the Annual Conference of the International Speech Communication Association (Interspeech)*, pages 3515–3519, Brno, Czech Republic, 2021. doi: 10.21437/Interspeech.2021-1531.

[172] Shoichiro Sato, Joren Six, Peter Pfordresher, Shina Fujii, and Patrick Savage. Automatic comparison of global children's and adult songs supports a sensorimotor hypothesis for the origin of musical scales. In *Proceedings of the International Workshop on Folk Music Analysis (FMA)*, pages 41–46, Birmingham, UK, 2019.

[173] Frank Scherbaum. On the benefit of larynx-microphone field recordings for the documentation and analysis of polyphonic vocal music. *Proceedings of the International Workshop Folk Music Analysis*, pages 80–87, 2016.

[174] Frank Scherbaum and Nana Mzhavanadze. Svan funeral dirges (Zär): Musical acoustical analysis of a new collection of field recordings. *Musicologist*, 4:138–167, 2020. doi: 10.33906/musicologist.782094.

[175] Frank Scherbaum and Nana Mzhavanadze. Svan funeral dirges (Zär): Language-music relation and phonetic properties. *Musicologist*, 5:66–82, 2021. doi: 10.33906/musicologist.875348.

[176] Frank Scherbaum, Meinard Müller, and Sebastian Rosenzweig. Analysis of the Tbilisi State Conservatory recordings of Artem Erkomaishvili in 1966. In *Proceedings of the International Workshop on Folk Music Analysis (FMA)*, pages 29–36, Málaga, Spain, 2017.

[177] Frank Scherbaum, Meinard Müller, and Sebastian Rosenzweig. Rechnergestützte Musikethnologie am Beispiel historischer Aufnahmen mehrstimmiger georgischer Vokalmusik. In *Proceedings of the Jahrestagung der Gesellschaft für Informatik (GI)*, pages 163–175, Chemnitz, Germany, 2017.

[178] Frank Scherbaum, Nana Mzhavanadze, and Elguja Dadunashvili. A web-based, long-term archive of audio, video, and larynx-microphone field recordings of traditional Georgian singing, praying and lamenting with special emphasis on Svaneti. *International Symposium on Traditional Polyphony*, 2018.

[179] Frank Scherbaum, Sebastian Rosenzweig, Meinard Müller, Daniel Vollmer, and Nana Mzhavanadze. Throat microphones for vocal music analysis. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.

[180] Frank Scherbaum, Nana Mzhavanadze, Sebastian Rosenzweig, and Meinard Müller. Multi-media recordings of traditional Georgian vocal music for computational analysis. In *Proceedings of the International Workshop on Folk Music Analysis (FMA)*, pages 1–6, Birmingham, UK, 2019.

[181] Frank Scherbaum, Nana Mzhavanadze, Simha Arom, Sebastian Rosenzweig, and Meinard Müller. *Tonal Organization of the Erkomaishvili Dataset: Pitches, Scales, Melodies and Harmonies*. Universitätsverlag Potsdam, 2020. doi: 10.25932/publishup-47614.

[182] Christian Schörkhuber, Anssi Klapuri, and Alois Sontacchi. Audio pitch shifting using the constant-q transform. *Journal of the Audio Engineering Society*, 61(7/8):562–572, 2013.

[183] Rodrigo Schramm and Emmanouil Benetos. Automatic transcription of a cappella recordings from multiple singers. In *Proceedings of the AES International Conference on Semantic Audio*, pages 108–115, Erlangen, Germany, 2017.

[184] Rodrigo Schramm, Andrew McLeod, Mark Steedman, and Emmanouil Benetos. Multi-pitch detection and voice assignment for a cappella recordings of multiple singers. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 552–559, Suzhou, China, 2017. doi: 10.5281/zenodo.1417671.

[185] Simon Schwär, Sebastian Rosenzweig, and Meinard Müller. A differentiable cost measure for intonation processing in polyphonic music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 626–633, Online, 2021. doi: 10.5281/zenodo.5624601.

[186] Sertan Şentürk. *Computational Analysis of Audio Recordings and Music Scores for the Description and Discovery of Ottoman-Turkish Makam Music*. PhD thesis, Universitat Pompeu Fabra, 2016.

[187] Xavier Serra. Computational approaches to the art music traditions of India and Turkey. *Journal of New Music Research, Special Issue on Computational Approaches to the Art Music Traditions of India and Turkey*, 43(1):1–2, 2014.

[188] Xavier Serra. Creating research corpora for the computational study of music: The case of the CompMusic project. In *Proceedings of the AES International Conference on Semantic Audio*, London, UK, 2014.

[189] William Sethares. Adaptive tunings for musical scales. *The Journal of the Acoustical Society of America*, 96, 1994.

[190] David Shugliashvili. *Georgian Church Hymns, Shemokmedi School*. Georgian Chanting Foundation, 2014.

[191] Joren Six, Olmo Cornelis, and Marc Leman. Tarsos, a modular platform for precise pitch analysis of Western and non-Western music. *Journal of New Music Research*, 42(2):113–129, 2013. doi: 10.1080/09298215.2013.797999.

[192] Julius Orion Smith. *Physical audio signal processing: For virtual musical instruments and audio effects*. W3K publishing, 2010.

[193] Ajay Srinivasamurthy, Gopala Krishna Koduri, Sankalp Gulati, Vignesh Ishwar, and Xavier Serra. Corpora for music information research in indian art music. In *Proceedings of the Joint Conference 40th International Computer Music Conference (ICMC) and 11th Sound and Music Computing Conference (SMC)*, Athens, Greece, 2014.

[194] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

[195] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. Open-Unmix – A reference implementation for music source separation. *Journal of Open Source Software*, 4(41), 2019. doi: 10.21105/joss.01667.

[196] Li Su, Tsung-Ying Chuang, and Yi-Hsuan Yang. Exploiting frequency, periodicity and harmonicity using advanced time–frequency concentration techniques for multipitch estimation of choir and symphony. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 393–399, New York City, New York, USA, 2016. doi: 10.5281/zenodo.1414838.

[197] Johan Sundberg. *The Science of the Singing Voice*. Northern Illinois University Press, 1987. ISBN 9780875805429.

[198] Johan Sundberg. Perceptual aspects of singing. *Journal of voice*, 8(2):106–122, 1994.

[199] Verena Thomas, Christian Fremerey, Meinard Müller, and Michael Clausen. Linking sheet music and audio – challenges and new approaches. In Meinard Müller, Masataka Goto, and Markus Schedl, editors, *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*, pages 1–22. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2012.

[200] Ingo R. Titze. The human instrument. *Scientific American*, 298(1):94–101, 2008.

[201] Zaal Tsereteli and Levan Veshapidze. On the Georgian traditional scale. pages 288–295, Tbilisi, Georgia, 2014.

[202] George Tzanetakis. Music analysis, retrieval and synthesis of audio signals MARSYAS. In *Proceedings of the ACM International Conference on Multimedia (ACM-MM)*, pages 931–932, Vancouver, British Columbia, Canada, 2009. doi: 10.1145/1631272.1631459.

[203] George Tzanetakis. Computational ethnomusicology: A music information retrieval perspective. In *Proceedings of the Joint Conference 40th International Computer Music Conference (ICMC) and 11th Sound and Music Computing Conference (SMC)*, pages 69–73, Athens, Greece, 2014.

[204] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003. doi: 10.1076/jnmr.32.2.143.16743.

[205] George Tzanetakis, Ajay Kapur, W. Andrew Schloss, and Matthew Wright. Computational ethnomusicology. *Journal of Interdisciplinary Music Studies*, 1(2):1–24, 2007.

[206] Burak Uyar, Hasan Sercan Atli, Sertan Sentürk, Baris Bozkurt, and Xavier Serra. A corpus for computational research of turkish makam music. In *Proceedings of the International Workshop on Digital Libraries for Musicology*, pages 1–7, London, UK, 2014.

[207] Parishwad P. Vaidyanathan. *Multirate systems and filter banks*. Pearson Education India, 2006.

[208] Peter van Kranenburg, Martine de Bruin, and Anja Volk. Documenting a song culture: the Dutch Song Database as a resource for musicological research. *International Journal on Digital Libraries*, 20(1):13–23, 2019.

[209] František Vávra, Pavel Nový, Hana Mašková, Michala Kotlíková, and Arnoštka Netrvalová. Morphological filtration for time series. In *Conference on Applied Mathematics (APLIMAT)*, pages 983–990, Bratislava, Slovakia, 2004.

[210] Werner Verhelst and Marc Roelands. An overlap–add technique based on waveform similarity (WSOLA) for high quality time–scale modification of speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Minneapolis, USA, 1993. doi: 10.1109/ICASSP.1993.319366.

[211] Anja Volk and Peter Van Kranenburg. Melodic similarity among folk songs: An annotation study on similarity-based categorization in music. *Musicae Scientiae*, 16(3):317–339, 2012. doi: 10.1177/1029864912448329.

[212] Anja Volk, Frans Wiering, and Peter Van Kranenburg. Unfolding the potential of computational musicology. In *Proceedings of the International Conference on Informatics and Semiotics in Organisations (ICISO)*, pages 137–144, Leeuwarden, The Netherlands, 2011.

[213] Sanna Wager, George Tzanetakis, Cheng-i Wang, and Minje Kim. Deep autotuner: A pitch correcting network for singing performances. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 246–250, Barcelona, Spain, 2020. doi: 10.1109/ICASSP40776.2020.9054308.

[214] Christof Weiß, Sebastian J. Schlecht, Sebastian Rosenzweig, and Meinard Müller. Towards measuring intonation quality of choir recordings: A case study on Bruckner's Locus Iste. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 276–283, Delft, The Netherlands, 2019. doi: 10.5281/zenodo.3527798.

[215] Graham F. Welch. The assessment of singing. *Psychology of Music*, 22(1):3–19, 1994.

[216] Nils Werner, Stefan Balke, Fabian-Robert Stöter, Meinard Müller, and Bernd Edler. trackswitch.js: A versatile web-based audio player for presenting scientific results. In *Proceedings of the Web Audio Conference (WAC)*, London, UK, 2017.

[217] Luwei Yang, Elaine Chew, and Khalid Z. Rajab. Logistic modeling of note transitions. In *International Conference on Mathematics and Computation in Music (MCM)*, pages 161–172, London, UK, 2015.

[218] Luwei Yang, Khalid Z. Rajab, and Elaine Chew. AVA: an interactive system for visual and quantitative analyses of vibrato and portamento performance styles. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 108–114, New York City, New York, USA, 2016. doi: 10.5281/zenodo.1415592.

[219] Luwei Yang, Khalid Z. Rajab, and Elaine Chew. The filter diagonalisation method for music signal analysis: frame-wise vibrato detection and estimation. *Journal of Mathematics and Music*, 11(1):42–60, 2017.

[220] Sangeon Yong, Soonbeom Choi, and Juhan Nam. PyTSMod: A python implementation of time-scale modification algorithms. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Montréal, Canada, 2020.

[221] Frank Zalkow, Sebastian Rosenzweig, Johannes Graulich, Lukas Dietz, El Mehdi Lemnaouar, and Meinard Müller. A web-based interface for score following and track switching in choral music. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018.

[222] Frank Zalkow, Angel Villar Corrales, TJ Tsai, Vlora Arifi-Müller, and Meinard Müller. Tools for semi-automatic bounding box annotation of musical measures in sheet music. In *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019.

[223] José R. Zapata, Matthew E. P. Davies, and Emilia Gómez. Multi-feature beat tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):816–825, 2014. doi: 10.1109/TASLP.2014.2305252.

[224] Udo Zölzer. *DAFX: Digital Audio Effects*. John Wiley & Sons, 2nd edition, 2011. ISBN 0470665998.