TU WIEN Informatics

Suche und Klassifizierung von Ornamenten in Lautentabulaturen

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Informatik

eingereicht von

Julia Maria Jaklin, BA, MA
Matrikelnummer 00073806

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.univ.Prof. Dr. Andreas Rauber

Wien, 1. Dezember 2025

	Julia Maria Jaklin	Andreas Rauber

Informatics

Search and Classification of Ornaments in Lute Tablature

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Informatics

by

Julia Maria Jaklin, BA, MA
Registration Number 00073806

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.univ.Prof. Dr. Andreas Rauber

Vienna, December 1, 2025

Julia Maria Jaklin   Andreas Rauber

Erklärung zur Verfassung der Arbeit

Julia Maria Jaklin, BA, MA

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel" habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, habe ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 1. Dezember 2025

Julia Maria Jaklin

Acknowledgements

I want to thank Olja and Reinier from the E-LAUTE-team for working together on the poster that was the basis for this thesis. I want to thank Susanne and Peter for proofreading and helpful comments along the way. And finally, I also want to thank Peter for his ongoing support and encouragement throughout my studies.

Kurzfassung

Die zunehmende digitale Erschließung historischer Lautenmusik eröffnet neue Möglichkeiten für die musikwissenschaftliche Forschung. Lautentabulaturen sind maschinenlesbar verfügbar, wodurch sich Fragen zu Bearbeitungsprozessen, melodischer Struktur und Verzierungspraxis systematisch und großflächig untersuchen lassen. Damit steigen auch die Möglichkeiten der Entwicklung von automatisierten Analyseverfahren, die die Notation der Lautenmusik inhaltlich auswertbar machen. Eine Herausforderung stellt die Identifikation und Klassifizierung von Ornamenten dar, die in der Lautenmusik explizit notiert sind und wichtige Hinweise auf musikalische Praxis liefern. Vor diesem Hintergrund untersucht die vorliegende Bachelorarbeit, ob historische Regeln zur Lautenverzierung algorithmisch formalisiert werden können. Grundlage dafür sind die von Vincenzo Galilei 1584 formulierten Regeln, wie sie in der Dissertation von Hiroyuki Minamino (1988) zusammengefasst sind. Die Arbeit bietet einen Überblick über Ansätze zur Suche in polyphoner Musik und diskutiert deren Grenzen im Kontext der Lautentabulaturen. Darauf aufbauend wird ein regelbasierter Algorithmus entwickelt, der die Ausgabe des `tabmapper`-Tools aus Reinier de Valks `abtab`-Toolbox analysiert und anschließend Galileis Regeln zur Klassifikation der identifizierten Verzierungstöne anwendet. Die Ergebnisse leisten einen Beitrag zur Entwicklung automatisierter Analysemethoden im Rahmen des E-LAUTE-Projekts und fördern ein tieferes Verständnis historischer Lautenornamentik und der Lautenmusik insgesamt.

Abstract

The growing digital availability of historical lute music is creating new opportunities for musicological research. As lute tablatures become accessible in machine-readable form, questions concerning adaptation processes, melodic structure, and ornamentation practice can be investigated on a much broader and more systematic scale. This development increases the possibilities for developing automated analytical methods capable of interpreting lute notation in a meaningful way. One of the challenges is the identification and classification of ornaments, which are explicitly notated in lute music and offer valuable insights into historical performance practice. In this context, the present bachelor thesis explores whether historical rules of lute ornamentation can be formalized into an algorithmic approach. The work draws on the ornamentation rules formulated by Vincenzo Galilei in 1584, as summarized in Hiroyuki Minamino's 1988 dissertation. The thesis provides an overview of existing approaches to searching within polyphonic music and discusses their limitations when applied to lute tablature. Building on this foundation, it develops a rule-based algorithm that analyzes the output of the `tabmapper` tool from Reinier de Valk's `abtab` toolbox and applies Galilei's rules to classify the identified ornamental tones. The results contribute to ongoing efforts within the E-LAUTE project to develop automated tools for analyzing lute intabulations and deepen our understanding of historical lute ornamentation and lute music in general.

Contents

CHAPTER 1

Introduction

Historical lute tablature presents a rich but challenging body of musical material for digital scholarship. Developed in the 15th and 16th centuries, its notational systems encode finger positions rather than pitches, and they exhibit substantial diversity in layout and symbols (see section 2.1). The musical texture ranges from simple, chord-based passages to polyphonic writing, and notated ornamentation and accidentals introduce additional variability. These characteristics make lute tablature difficult to analyze computationally and complicate the development of tools for searching, comparing, or structurally interpreting its contents.

These challenges become especially relevant in projects that aim to provide large-scale digital access to lute repertories. The E-LAUTE project[1], within which this research is situated, seeks to create a comprehensive digital platform[2] for historical lute music. For such a platform, search capabilities must extend beyond basic metadata and support retrieval based on melodic similarity, structural patterns, and relationships to vocal models. Achieving this for lute tablature, however, requires addressing fundamental methodological problems in representing and querying a notation that is both polyphonic and non-pitch-based.

These issues connect directly to broader challenges in music information retrieval (MIR). While search techniques for monophonic melodies are well established, the task becomes significantly more complex for symbolically encoded, unvoiced polyphonic music. Current digital music libraries commonly offer only limited content-based retrieval: repositories such as the Petrucci Music Library[3] and the Choral Public Domain Library[4] provide PDF scores and metadata, but no meaningful melodic or structural search. Databases

[1] https://e-laute.info/

[2] https://edition.onb.ac.at/context:elaute

[3] https://imslp.org

[4] https://cpdl.org

like Themefinder[5] and RISM[6] allow users to search for short melodic patterns, yet their applicability to polyphonic repertoires or non-pitch notations is restricted. Research projects such as the Josquin Research Project[7] focus on a specific repertoire and therefore do not address the broader challenges posed by historical or instrument-specific notations. Optical music recognition (OMR) technologies hold promise for increasing access, but they remain unreliable for historical sources, where diverse and often irregular notational practices—especially in handwritten materials—produce high error rates [1].

Against this background, the present study focuses on one specific and underexplored aspect: the computational treatment of notated ornamentation in lute tablature. Ornamentation varies across treatises and styles and can obscure underlying melodic or structural patterns, posing challenges for similarity search and analysis. This leads to the central research question of this work:

> RQ: How can the rules for ornamenting found in a historical lute treatise be formalized into an algorithmic approach to identify and classify ornamentation in lute tablature?

Section 2.1 introduces the fundamentals of lute tablature. Section 2.2 reviews literature on (polyphonic) music similarity, which provides the conceptual basis for search and classification. Given the complexity of the task, a simple rule-based approach was developed to explore ornamentation search in lute tablature; this implementation is presented in chapter 3.

[5] http://www.themefinder.org
[6] https://opac.rism.info/rism/Search/
[7] https://josquin.stanford.edu/

CHAPTER 2

Background & Related Work

The development of a strategy to search and classify ornaments in German lute tablature requires understanding of two quite different domains: the unique characteristics of tablature notation and lute music, as well as the theoretical foundations of musical similarity and the computational approaches available for (polyphonic) music retrieval.

2.1 Lute Tablature

The lute was one of the most popular instruments in Europe from the Middle Ages to the Renaissance. It is a plucked string instrument with a rounded back with strings arranged in usually 5 or 6 courses, of two strings each (see Figure 2.1). Both strings of one course are plucked with a quill as a plectrum, later also the fingers were used. The highest-pitched course usually consists of only a single string, though. The neck of the lute is divided into small sections with frets made of gut. As with other lute-like instruments, as well as guitars, each fret represents a half-tone on the Western diatonic tone scale.

The playing style changed throughout the centuries, but by the early 16th century, a style had been established that allowed for solo polyphonic performance, incorporating several voices simultaneously. [2, 6-7] By then, three different notations had been developed to notate music for the lute: French tablature, Italian tablature, and German tablature[1], with German tablature said to be the oldest one [3]. In the oldest printed work about musical instruments by Sebastian Virdung, *Musica getutscht* (Basel, 1511), it is reported

[1] Recently, the naming of the different notation types have been discussed as the notation types were not only used in the regions than the names suggest. Researchers have proposed to use the terms letter, number and matrix tablature instead of French, Italian and German tablature respectively. This topic was discussed at the Basel Study Days on Lute & Tablature 2025, 11.09.-13.09.2025 in Basel, Switzerland. Due to the established use of the traditional names, I will continue to use them in this thesis.

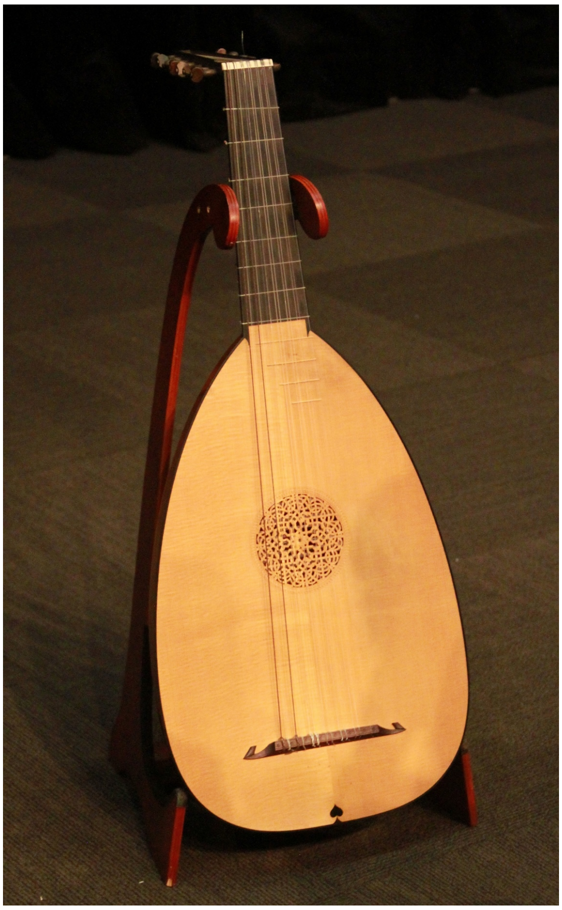

Figure 2.1: Picture of a renaissance lute. Usually, the strings of a lute are arranged in courses, of two strings each. The highest-pitched course often consists of only a single string. The neck of the lute is divided into small sections with frets made of gut. "Lute" by Princess Ruto, CC BY 2.0. “Lute” by Princess Ruto, CC BY 2.0

that German lute tablature was allegedly invented by the blind musician Konrad Paumann around the middle of the 15th century.

The first surviving scores were written for 5-course lutes, but around the end of the 15th century, 6-course lutes became more widespread as well. Conceptually, lute tablature is a form of direct ’finger notation’, where the position of the fingers on the strings is notated, rather than the pitch of the notes, like in indirect ’pitch notation’ such as Common Western Music Notation (CWMN). See Figure 2.2 for an example of German lute tablature.

Many lute pieces are adaptations of vocal compositions, such as motets, chansons, and madrigals. These works were typically written for four or five singing voices, and by the end of the 15th century even for six or more. In lute scholarship, such polyphonic vocal works are referred to as vocal models, because they served as the source material from

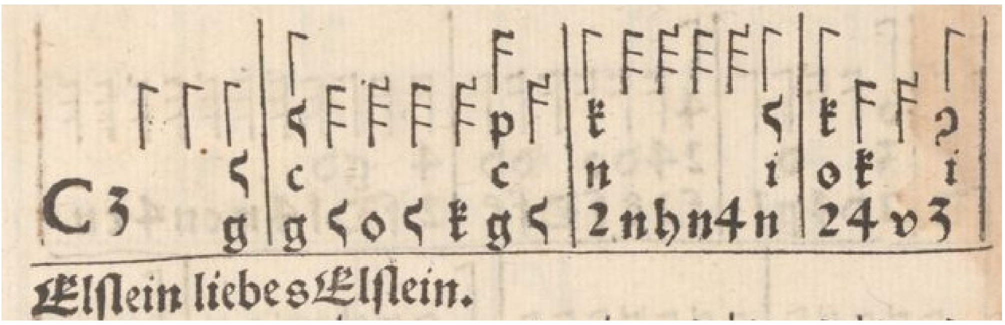

Figure 2.2: The beginning of "Elslein liebes Elslein" in the original source from Hans Judenkünig, Ain schone kunstliche vnderweisung, 1523, folio 24v. In the top line the rhythm signs are visible, with resemblance to modern rhythm flags. Below, the numbers and letters indicate the notes to be played, chords are symbols stacked above each other. The first chord consists of the highest empty string (5) and the second fret on the fourth course (g). The measures are separated with vertical lines. See Figure 2.3 for the transcription.

which lutenists created intabulations—instrumental versions arranged for the lute.

Adapting a vocal model for lute required substantial reworking to fit the idiomatic and technical constraints of the instrument. The intabulator needed to condense the multi-voice texture into a playable form, often redistributing or omitting certain notes. Voice crossings and unison doublings had to be removed, and awkward harmonic configurations, such as large spacings or voice-leading that resulted in unplayable fingerings, had to be reshaped. In addition, performers frequently introduced embellishments, including diminutions and rapid ornamental figures, to enhance the musical texture and exploit the expressive possibilities of the lute [2, 5-6].

When using algorithmic methods to analyze lute tablature, the need for a digital representation of lute tablature arises. In the E-LAUTE project, the tablature module of the Music Encoding Initiative (MEI) is used to transcribe the music as faithfully as possible. See Figures 2.3, 2.4, 2.5, 2.6, and 2.7 for different examples of transcriptions.

The transcription process is laborious and requires a deep understanding of the tablature notation. The E-LAUTE project relies on guidelines established within the project to ensure consistency and accuracy in the transcription process.[2]

2.1.1 Lute Tablature Systems

In contrast to German lute tablature, French and Italian lute tablature use a system of lines and letters (in the case of French) or numbers (in the case of Italian) to indicate the position of the fingers on the strings, a graphic representation of the finger board. When looking at the tablature, the player can imagine the strings of the lute as the lines of the

[2] See the E-LAUTE project website for more details on the transcription guidelines: https://e-laute.info

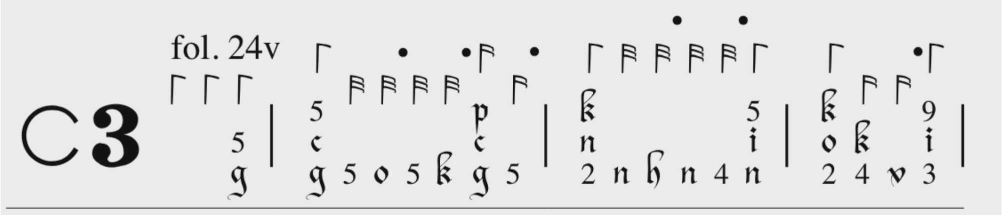

Figure 2.3: The same beginning of "Elslein liebes Elslein" in the diplomatic transcription in German lute tablature from the E-LAUTE project, using the MEI module for lute tablature and rendered using Verovio in the mei-friend web editor. The diplomatic transcription aims to preserve the original notation as closely as possible. There is also an edited transcription available with editorial changes, but for this small example, the two transcriptions are identical. See Figure 2.6 for the diplomatic transcription or Figure 2.7 for the polyphonic transcription.

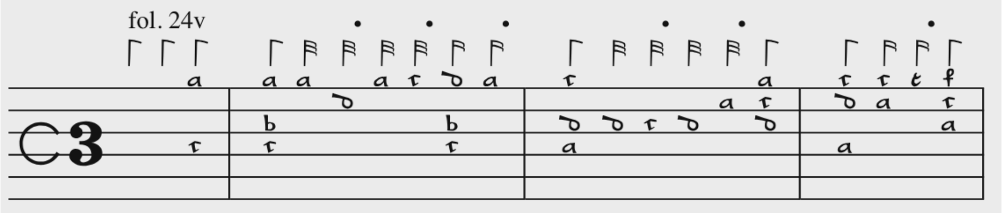

Figure 2.4: The beginning of "Elslein liebes Elslein" in the transcription in French Lute Tablature from the E-LAUTE project, using the MEI module for lute tablature and rendered using Verovio in the mei-friend web editor. Finger placements are represented as letters with *a* being the empty string, *b* the first fret and so on. The uppermost line represents the lowest string.

tablature, with the numbers or letters indicating the frets on which the fingers should be placed. See Figures 2.4 and 2.5 for an example of French and Italian lute tablature.

Rhythm is notated above the tablature-stave in all three notations similarly, although with varying symbols, depending on the scribe and origin of the source [4, 54-56]. Vertical bar separators are also used in all three types. To be able to interpret the tablature in a pitch notation, the tuning of the lute must be known (or assumed). Virdung describes the tuning of the 6-course lute as follows: A-d-g-b-e'-a', but also tunings in G are common, such as G-c-f-a-d'-g' [2].

2.1.2 German Lute Tablature

German lute tablature, on the other hand, is a further abstraction as it uses no lines to indicate the finger placement on the strings, but a system of letters and numbers to indicate the position of the finger on the string. See Figures 2.2 (original source) and 2.3 (transcription) for an example of German lute tablature.

German tablature has been argued to be especially well suited for the representation of polyphonic music [2, 3]. Some sources (Newsiedler 1536 and Ochsenkuhn 1558) present

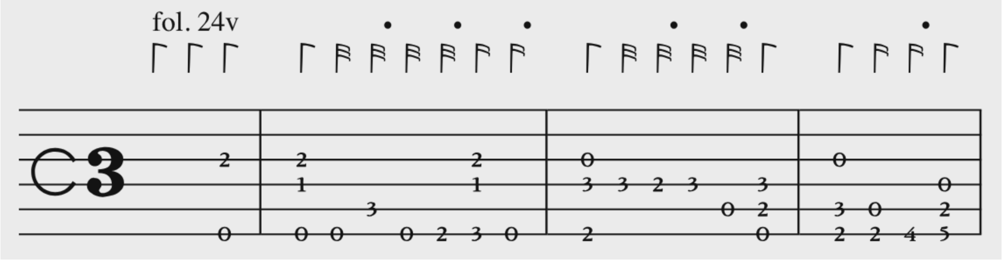

Figure 2.5: The beginning of "Elslein liebes Elslein" in the transcription in Italian lute tablature from the E-LAUTE project, using the MEI module for lute tablature and rendered using Verovio in the mei-friend web editor. Finger placements are represented as numbers with 0 being the empty string and 1 the first fret and so on. The lowest line represents the lowest string.

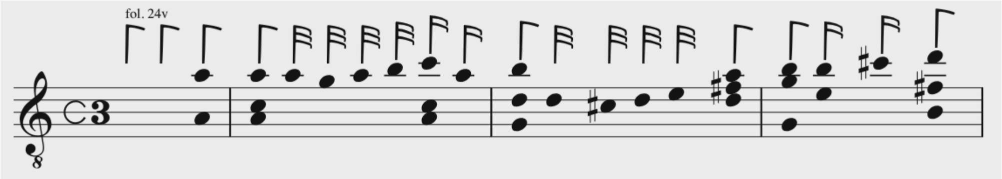

Figure 2.6: The beginning of "Elslein liebes Elslein" in the diplomatic transcription in so-called notehead notation from the E-LAUTE project, using the MEI module for German lute tablature and rendered using Verovio in the mei-friend web editor. This notehead notation is a representation of the tablature in pitch notation, preserving the original tablature arrangement relatively closely, but providing better readability for people not as familiar with tablature notation. This notation though loses the specific information on which string and by which finger a tone is to be played.

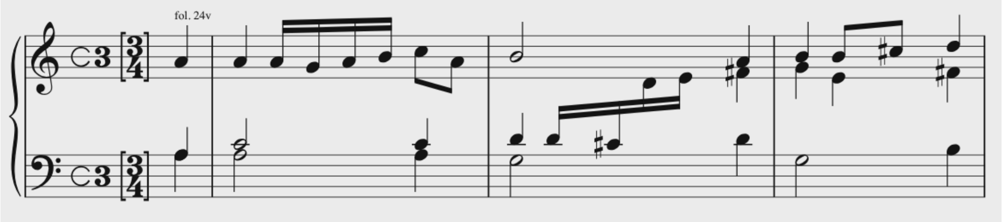

Figure 2.7: The beginning of "Elslein liebes Elslein" in the common music notation from the E-LAUTE project, using the MEI module for German lute tablature and rendered using Verovio in the mei-friend web editor. This representation is also in pitch-notation and the rhythm has also been incorporated directly into the staff, presenting the music in a way that is familiar to most musicians and musicologists today.

the music in a way that allows distinction between different voices of the polyphony. However, many other sources do not follow this practice; instead, the symbols are often arranged in a way that makes it difficult to distinguish the voices, with the different polyphonic lines becoming apparent only when the piece is performed [2, 3-4].

To familiarize players with the symbols used to indicate the position on the fretboard, they are laid out in a so-called 'Lautenkragen' (lute collar), a kind of key that shows the player how to interpret the tablature, which is shown in Figure 2.8 and Table 2.1.

The Lautenkragen from Hans Judenkünigs *Ain schone kunstliche vnderweisung* (See Figure 2.8 and the transcription in table 2.1) shows the symbols used in German lute tablature and their corresponding location on the fingerboard. The evolution of the symbols through time can be seen in this figure, as earlier lutes only had 5 courses, while later lutes had 6.

The notation system follows two distinct patterns depending on the string, as shown in Table 2.1. The lowest course (6th string) uses capital letters A through I, where A represents the empty string and each subsequent fret is denoted by the next letter in alphabetical order (B, C, D, etc.). This different nomination scheme is used as the lowest string/course was added later, historically, and the system is not flexible to account for different numbers of courses. The "9" on Course 5, Fret 5 was seemingly added as another symbol was needed to fill the gap. This way, letters with and without bar are simply shifted 5 frets up (an interval of a fourth). Note that letters in this scheme still only represent finger positions on the fretboard and do not tell you about the musical pitch or notes themselves, because different tunings retune the courses.

The upper five courses (strings 1-5) follow a more complex pattern:

- Empty strings are numbered 1 through 5 (from lowest to highest)
- The first few frets use lowercase letters starting with 'a' on the lowest of these strings
- Higher frets use letters with an overline (bar) above them: $\overline{a}$, $\overline{b}$, etc.
- The letter sequence continues alphabetically across strings and up the fret-board

Empty	Fret 1	Fret 2	Fret 3	Fret 4	Fret 5	Fret 6	Fret 7	Fret 8
5	e	k	p	v	9	$\overline{e}$	$\overline{k}$	$\overline{p}$
4	d	i	o	t	z	$\overline{d}$	$\overline{i}$	$\overline{o}$
3	c	h	n	s	y	$\overline{c}$	$\overline{h}$	$\overline{n}$
2	b	g	m	r	x	$\overline{b}$	$\overline{g}$	$\overline{m}$
1	a	f	l	q	w	$\overline{a}$	$\overline{f}$	$\overline{l}$
A	B	C	D	E	F	G	H	I

Table 2.1: German lute tablature symbols organized by string and fret position, showing the symbol used for each position on the fret-board, the rows show the highest to lowest course, 5 being the highest course and A the lowest. Transcription of the Lautenkragen (lute collar) from *Ain schone kunstliche vnderweisung*, Hans Judenkünig, 1523, folio 15v. See Figure 2.8 to view the original.

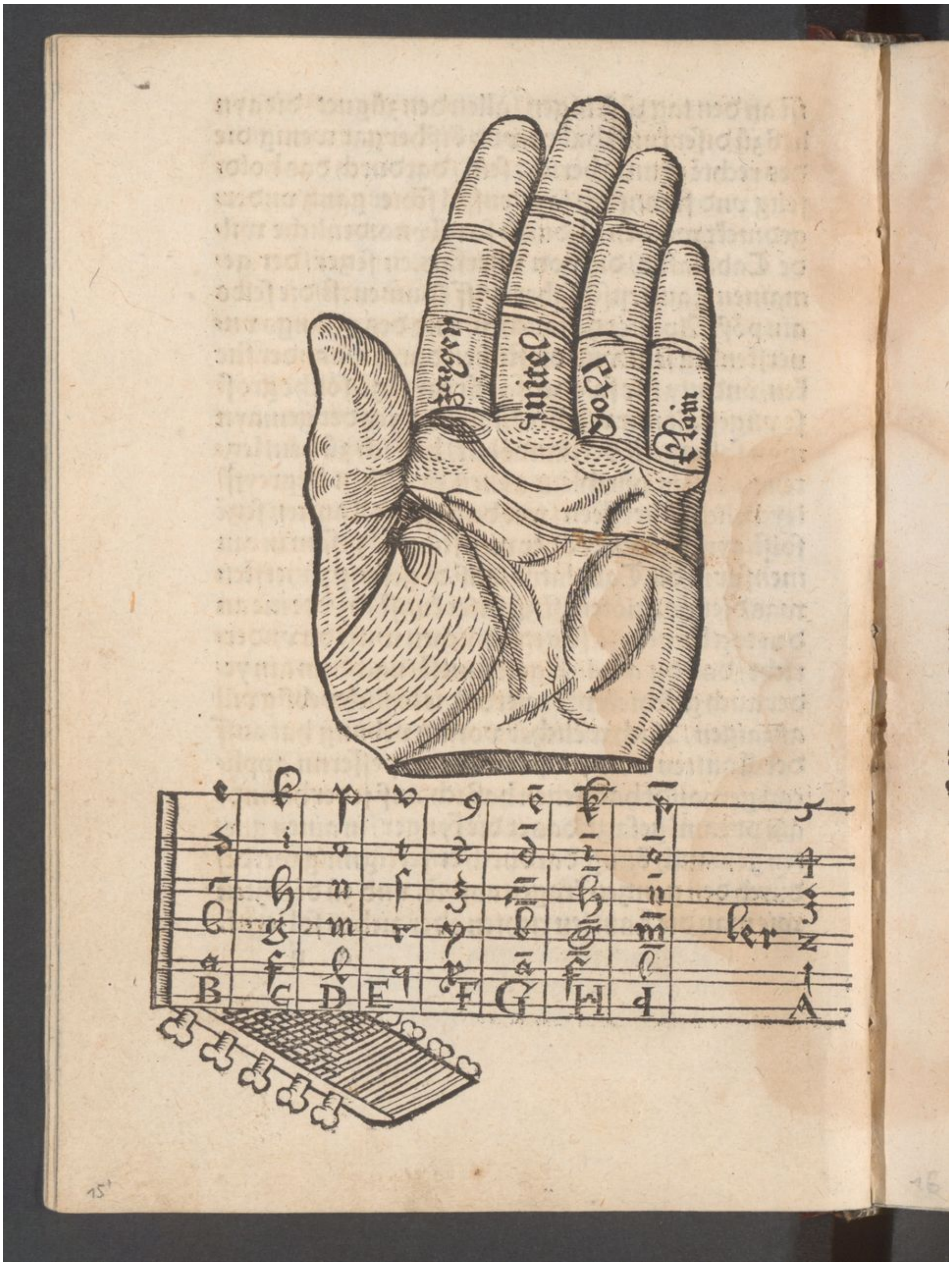

Figure 2.8: Lautenkragen (lute collar) from *Ain schone kunstliche vnderweisung*, Hans Judenkünig, 1523, folio 15v. The Lautenkragen shows the symbols used in German lute tablature and their corresponding location on the fingerboard. See Table 2.1 for more details about the symbols. The symbols for the empty strings are shown on the very right, indicated with the text "ler", in modern german "leer", for "empty". The lowest notes on the frets start on the leftmost end of the fretboard. The lowest course is visually on the bottom and it can be observed that the naming system is different than for the rest of the positions. This course was added later onto the already existing system. There is also a hand with names on the fingers: "zaiger" (pointer finger), "mitel" (middle finger), "gold" (ring finder), "klain" (little finger).

2.1.3 Intabulations

In the sixteenth century, lute music can be categorized into three types: original compositions in varying degrees of counterpoint (e.g., fantasias, ricercars or preludes), settings of dance tunes, and intabulations. An intabulation is a form of musical transcription that adapts polyphonic vocal music for solo instruments, in this case the lute. This process often involves the addition of ornaments, which are decorative musical elements that enhance the expressiveness of the performance. Intabulations make up a significant portion of the lute repertoire that has been preserved from the Renaissance period. [5]

Before the invention of tablature notation, instrumental performances of vocal pieces likely relied on the printed vocal parts from part-books, which required familiarity with mensural notation as well as skills to abstract a playable version for the lute and adding ornamentation where appropriate. Simultaneously reading from (at least) four different voices from the separate part-books required lots of practice and knowledge. To simplify playing and also enabling lesser skilled players, instrument-specific tablatures (e.g., for organ or lute) were developed; these show the polyphonic texture at a glance and often include ornamentations. Such intabulations are among the earliest existing sources of written down instrumental music. The term "intabulation" refers both to the act of adapting vocal music and to its notated form. Due to the popularity of the lute as an instrument for both amateur and professional musicians, books were published that provided guidance on how to play the lute as well as how to create intabulations. [6]

One of the earliest sources in German that includes instructions on how to make such an intabulation is Hans Judenkünigs "Ain schone kunstliche Underweisung", Vienna 1523, Part II (from which also Figure 2.2 and Figure 2.8 were taken). An Italian source is Vincenzo Galilei's "Fronimo Dialogo" from 1584, which contains a section dedicated to the rules and practices of ornamentation in lute music. These are just two of several examples of treatises that discuss the topic of intabulation and playing techniques for the lute.

2.1.4 Vincenzo Galilei's Rules for Ornamentation

In his 1988 dissertation, Hiroyuki Minamino summarizes the rules for ornamentation as described by Vincenzo Galilei in his 1584 treatise *Fronimo Dialogo.* [7]

Ornaments are formulas like mordents, trills and turns used by lutenists to embellish a melody. Due to the physical functionality of a lute, small, quick notes are added to elongate the main notes and to create a more interesting melodic line. A singers voice can sustain notes for a rather long time, but a plucked string instrument like the lute produces sounds that decay quickly. Galilei states that ornaments should be added according to the rules of counterpoint. [7, 105]

The following points[3] outline key aspects of Galilei's rules as interpreted by Minamino:

1. Ornamental figures cannot overlap: "In most cases, Galilei limits the ornaments to one voice in three- or four-voice counterpoint." ([7, II/2., pp. 105-107, §2])
2. Step-wise motion of 4 notes in the range of a fifth: "Most of the leaps consist of a fifth so that the step-wise motion of four notes can fill the interval." ([7, §2])
3. Original voice should not be changed: "Original voice should not be changed. [...], the counterpoint remains the same as the original, since the ornaments start on the main note." ([7, §2])
4. Changing the pace too abruptly: "It is wrong, Galilei writes, to use ornaments which move from fusae or semifusae to minims or semibreves." ([7, §4])
5. First note of an ornamentation is consonant: "The first note of ornaments should not be dissonant with other voices." ([7, §3])
6. Every fourth semiminima is consonant: "Every fourth semiminima of every ornamental figure must be consonant with the other part." ([7, §3])
7. No successive dissonances in disjunct movement: "Embellishment with successive dissonances in disjunct movement is forbidden." ([7, §3])
8. No parallel fifths: "Creating parallel fifths by applying ornaments should be avoided." ([7, §3])
9. No embellishment in succession of chords: "[T]he cadence in which there is a succession of chords in small equal rhythm should not be embellished." ([7, §5])

These rules provide a rough framework for understanding how ornamentations could be applied in lute intabulations. They are only one perspective on the topic, and other sources may present different or additional guidelines or rules. For this thesis, I chose to implement a subset of these rules for the search and classification of ornamentations in lute tablature. The specifics of the implementation of the chosen rules will be discussed in chapter 3.

2.2 Similarity in Music

Musical similarity is a fundamental concept in music information retrieval, serving as the foundation for tasks such as finding themes, detecting plagiarism, analyzing musical styles, searching music databases, and, of course, searching for ornamentations. Intuitively, two musical passages are similar when they share important features in a given context, including melodic, rhythmic, harmonic, timbral, or structural characteristics. However, modeling similarity formally presents significant challenges because music has different aspects, is organized in layers, and depends on context. What counts as "similarity"

[3] This summary has been taken from a poster presented by my colleagues Olja Janjuš, Reinier de Valk and me at the ICCCM25, which can be found here: `https://www.titanmusic.com/icccm2025/slidesandposters/JanjusPoster.pdf`

in one task (such as recognizing a melody) may be quite different in another task (such as analyzing a performance) [8]. Therefore, practical systems must use specific representations and measures that are tailored to their particular objectives.

Computational approaches to musical similarity typically follow two different methodological paths. The first strategy works directly on the musical surface, using approximate matching techniques that compare sequences of notes or events with, for example, edit distances (a measure of the similarity of two strings) or dynamic programming algorithms. This approach can handle insertions, deletions, substitutions, and small timing differences. While this flexibility is advantageous, it comes with costs: the computation required can be enormous, and choosing appropriate thresholds or normalization parameters requires care to prevent irrelevant matches [8].

The second strategy first abstracts the musical score into simpler representations such as intervals, contours, or chord labels – and then applies exact matching algorithms on this reduced space. This approach can be very efficient and inherently handles transformations like transposition when interval-based representations are used. However, abstraction necessarily involves discarding some surface-level musical details. In practice, hybrid approaches work best: using coarse abstraction for fast filtering followed by more detailed comparison for re-ranking results. [8]

2.2.1 Challenges in polyphonic music

Polyphonic music presents significant challenges for tokenization (transformation to reduced information content) and pattern matching because voices overlap and their timing relationships are often ambiguous. The following representational strategies have been developed to address these complexities. The first approach encodes music as one or more monophonic sequences of letters (strings), typically one string per voice, which allows the use of well-established string algorithms and information retrieval techniques. This method works well when reliable voice separation is available, but it may miss patterns that span multiple voices and depends on accurate part extraction [9].

The second approach uses a geometric point-set model where each note becomes a point in time-pitch space, possibly with additional weights such as duration values or other additional features. Pattern matching becomes a geometric problem and can handle transformations flexibly without requiring explicit voice labels [10, 11, 12]. The cost includes increased computational complexity and the tendency to generate many musically irrelevant patterns without appropriate constraints.

A third approach abstracts the musical surface into harmonic or functional descriptors - including chord labels, chroma-like vectors (similar to pitch-classes), or tonal-space functions - and performs comparisons on these derived sequences. This strategy is often the most natural choice for harmony-focused analytical tasks [8].

Polyphonic music is difficult to split into simple tokens. Unlike a single melody or a line of text, it does not naturally become a single, clear stream of tokens. There are different strategies to work with polyphonic music, some of which will be discussed in the following.

Approximate matching on sequences

Edit-distance and sequence-alignment methods work directly on symbolic sequences, including pitches, intervals, durations, or small feature groups. In their basic form, these techniques count the number of edit operations needed to transform one sequence into another. Dynamic programming variants achieve transposition-invariance by working on intervals rather than absolute pitch values. Dynamic programming approaches, such as those by Laitinen and Lemström [13], represent both query patterns and database content as points in 2D space (time, pitch) and use state tables to track matching progress while allowing for note skipping and tempo variations. Timing flexibility is achieved by comparing relative timing patterns or by allowing gaps within limited windows. These methods are intuitive and can handle local variations, but they depend on good parameter settings and can become computationally expensive for long, complex polyphonic inputs without some preliminary reduction. A related practical approach encodes overlapping n-grams (sequences of n adjacent symbols in a particular order) of musical events to enable efficient indexing. The n-gram approach represents musical content as event sequences and uses sliding window techniques to extract single-voice sequences from polyphonic textures. Rather than using absolute pitch values, intervals between consecutive notes and ratios of rhythmic durations are encoded to achieve transposition and tempo invariance. These intervals and ratios are then mapped to text characters using encoding schemes that reflect musical patterns, which enables the use of standard text retrieval methods [14].

Geometric matching and transportation distances

Geometric methods treat musical scores and queries as weighted point sets in time-pitch space and search for transformations that align patterns with passages in target pieces. An important approach uses transportation distances such as Earth Mover's Distance and similar variants, which measure the minimal "work" needed to transform one point set into another. When musical content is divided into short, overlapping sections with normalized time axes per section, these distance measures become practical for large datasets while remaining robust against tempo changes and small pitch variations [11]. A related line of research focuses specifically on transformed pattern matching within classes of musically meaningful transformations—including transposition, inversion, retrograde, and time scaling. Recent parallel algorithms can efficiently find maximal transformed matches without requiring voice labels, making them particularly suitable for dense polyphonic textures where voice leading relationships are unclear [10]. These algorithms build on earlier foundational work including (e.g., SIATEC) and related pattern discovery methods [15]. Hausdorff-style distances provide another geometric approach, addressing two key problems: measuring dissimilarity based on Euclidean distances between point sets, and maximal subset matching where similarity relates to the number of points in corresponding locations [12]. These methods, combined with geometric hashing techniques, offer efficient ways to locate exact or near-exact query occurrences, though they trade some matching flexibility for enhanced speed and precision.

Graph based representations

Recent work in music information retrieval has explored representing symbolic music as graphs where notes and other musical symbols are connected through relationships such as onset synchrony and temporal succession, with subsequent learning of graph embeddings using Graph Neural Networks (GNNs). Specialized software tools designed for symbolic music analysis have made it possible to learn representations that capture both local contextual information and cross-voice relationships [16]. These learned representations can support various analytical tasks including clustering, content-based retrieval, and specialized analyses such as cadence detection. For historical musical notations or repertoires with ambiguous voice alignment, it is often preferable to use simplified graph models with minimal structural assumptions, allowing the learning algorithms to capture patterns from contextual data. This approach offers a data-driven complement to manually designed similarity measures, provided that evaluation procedures remain careful, robust and task-specific. See [17] for an example with music from the Notre Dame repertoire.

2.2.2 Limitations

No single representation can capture all aspects of musical similarity completely. String-based methods can have difficulties with dense polyphonic textures; geometric methods may generate too many pattern matches without appropriate musical constraints; harmonic abstractions necessarily lose melodic surface information; learned models depend heavily on training data quality and graph design.

Musical similarity can be implemented through sequence alignment techniques, geometric matching algorithms, and harmonic abstraction methods, with increasing support from graph-based learning approaches. For polyphonic musical content, representations that either respect voice structure or bypass it through geometric or harmonic summary techniques work most effectively. Method selection should be guided by the specific analytical task, desired invariance properties (including transposition, tempo/temporal warping, and inversion), and available computational resources.

With the data set available for this thesis, only limited options for implementing classification of ornamental figures were feasible. The data set consists of lute tablature intabulations notated in TabCode (see Chapter 3.1 for more details). The most straightforward way to implement the classification of ornamentations in this format is a rule-based approach, as described in chapter 3.

CHAPTER 3

Ornamentation Search and Classification

In this chapter, the implementation of the rules for ornamentation by Vincenco Galilei (see Section 2.1.4) will be translated into computational logic.

3.1 Data Set

The dataset chosen for testing the implementation is taken from the JOSQUINTAB dataset, also partially digitized in the framework of the E-LAUTE project. The data is available on GitHub[1]. It contains 64 intabulations of vocal music for lute of vocal compositions by Josquin des Prez (ca. 1450-1521) saved in TabCode (`.tbp`), along with their corresponding vocal models in `MIDI` format. The intabulations were created by different people. Additionally, the data set contains metadata about the pieces, and the output of the alignment of the intabulations with their vocal models using the tool `abtab`[2]. For more details about the origins of the dataset, see Section 3 in the original paper [5]. To understand the following process, it is necessary to first explain how `abtab` works and what kind of output it provides.

3.1.1 abtab

The toolbox `abtab` provides computational processing of music written in lute tablature. One of its modules, `tabmapper`, is designed to align lute tablature with a vocal model in `MIDI` format and make a best-guess on the re-separation of voices of the lute tablature. It needs a tablature encoding (in `.tbp` format) and a `MIDI` file of the vocal model (in separate voices) as input. After processing, the output consists of a `.csv` file with

[1] https://github.com/reinierdevalk/data/tree/master/tabmapper/josquin-int

[2] https://github.com/reinierdevalk/abtab

information about each note in the tablature with each note in a new row, a MIDI file of the lute tablature with the separated voices, an MEI file[3] with separated voices with added information from the .csv file embedded as an additional attribute (<color>), and an additional .csv file with preliminary information about the ornaments found in the tablature. To view the rendered MEI-file, the web-app *mei-friend* can be used[4].

During the alignment process, each tablature note is assigned to one of five categories that describe its relationship to the vocal model: matches, ornamentations, repetitions, adaptations, and ficta. The last of these categories relates to a historical notational practice: in Renaissance vocal music, accidentals were often not written in the score but were expected to be supplied by the singers during performance. Such unwritten accidentals are referred to as *musica ficta*[5] After categorizing the notes, they are assigned to specific voices based on how they align with the corresponding notes in the vocal model.

The different categories are defined as follows:

- match (normal; black):
  - there is a note at that onset time in the vocal model with the same pitch
- ornamentation (blue):
  - no note at that onset time in the vocal model
  - only this note at that onset time
  - duration less than or equal to ornamentation threshold
- repetition (green):
  - the previously played note in the vocal model in that voice
- adaptation (red):
  - there is a note at that onset time in the vocal model, but it is a different pitch
- ficta (orange):
  - this note's accidental is notated in the tablature that is not notated in the vocal model

More information about the tabmapper algorithm can be found in chapter 4 of the JOSQUINTAB paper [5].

3.1.2 Processing the output of abtab

As mentioned before, the tabmapper-output includes two .csv files per piece. Only the .csv file with information about each note in the tablature is used for the ornament identification process. In the following, this file is referred to as "mapping-csv". The mapping-csv contains (excluding some unused columns) the following information for each note in the tablature per row:

[3] The Music Encoding Initiative (MEI) defines an XML-based system for encoding musical documents in a machine-readable structure. See https://music-encoding.org/.

[4] https://mei-friend.mdw.ac.at/

[5] For more information, see for example, https://en.wikipedia.org/wiki/Musica_ficta.

- note index (zero-based)
- pitch (MIDI number)
- duration (as a fraction of a whole note, encoded as a string)
- onset time (zero-based, absolute, as a fraction of a whole note)
- voice (zero-based, 0 is the highest voice)
- categorization flag (match, ornamentation, repetition, adaptation, ficta)

First, the mapping-csv is preprocessed to prepare the data for the ornament identification process. Some columns contain data that are not reliable (e.g., bar numbers) or not needed for the process (e.g., cost). In the voice column it can happen that a note is assigned to two different voices simultaneously, because there is a unison in the vocal model (i.e. the same note in two (or more) different voices). These special cases are handled by duplication, and each copy is assigned to one of the two voices in two separate rows.

Additionally, an alternative voice assignment ("squished") is created separately by assigning all notes into the highest possible voice. Notes at the same onset time are then assigned, into the other voices (starting from the top), which is more similar to how the lute tablature is written in one staff. This second assignment allows better performance evaluation of the algorithms presented later. Figures 3.1 and 3.2 show the difference between the two voice assignments for an example piece.

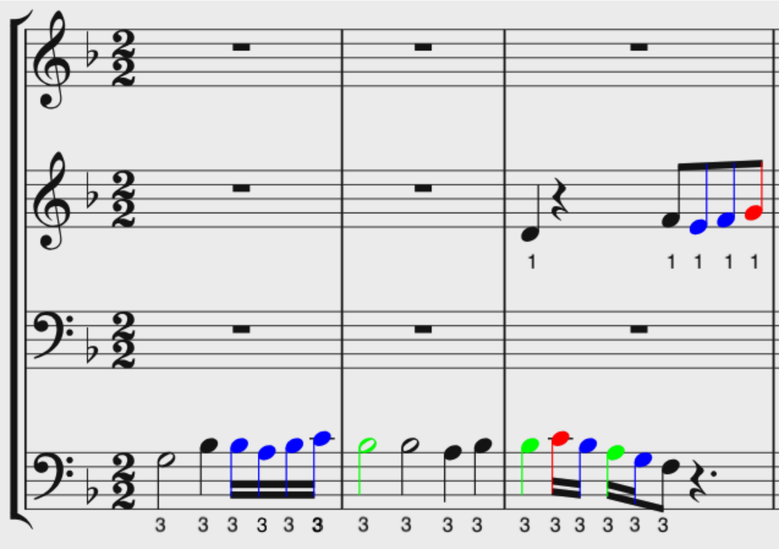

Figure 3.1: Voice assignments from the `abtab` output, visualized in the voice separated CMN notation. Notes in color mark annotated notes found by `abtab`. Blue notes are annotated with 'ornamentation', green notes with 'repetition', red notes with 'adaptation' and orange notes with 'ficta'.

In the next step, the data from the preprocessed mapping-csv is used to identify ornamental figures that follow Galilei's rules.

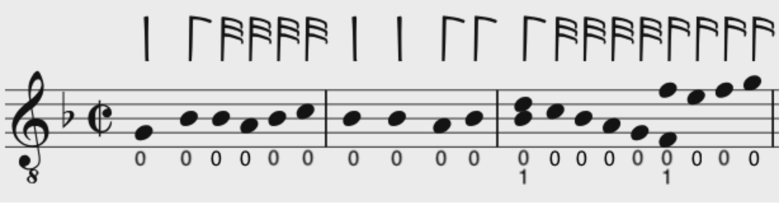

Figure 3.2: Voice assignments from the alternative representation, where each note is assigned to the highest possible voice, visualized in the diplomatic transcription.

3.2 Implementation of the rules

The idea of classifying ornaments by historical rules was initially proposed by my colleague Olja Janjuš and first presented as a poster by Olja Janjuš, Reinier de Valk and me at the ICCCM25 in Aalborg, Denmark in October 2025.[6] This section will discuss the further development of the presented ideas and algorithm and focus on comparing the voice separated approach described on the poster with an approach closer to how lute tablature works.

The code, implemented in Python 3.13.3, and data used for evaluation is available on GitHub[7]. Instructions for setting up the environment and running the code can be found in the repository's `README.md` file.

When implementing the rules for ornamentation described in Section 2.1.4, three different approaches were taken for identifying ornamental figures in the data. The first approach ("abtab") used the annotations from `abtab`'s `tabmapper` as a starting point, and then merged consecutive notes with the same rhythmical value into larger ornamental figures. The second approach ("raw") did only rely on finding sequences of notes with the same rhythmical value below the ornamentation threshold given by `abtab`, without considering the annotations provided in the output of `abtab`. For both approaches, context notes before and after the sequences were included to be able to analyze the rules that require context information (like the motion in range of a fifth). Finally, a third approach ("voices_separated") used the voice separation proposed by `abtab` to identify ornamental figures in the individual voices and then processed them similarly to the "abtab" approach, but kept each voice separate. Then, several filter steps based on the rules were applied to each set of identified ornamental figures.

3.2.1 Ornamental figures cannot overlap

"In most cases, Galilei limits the ornaments to one voice in three- or four-voice counterpoint." ([7, II/2., pp. 105-107, §2])

[6] The poster presented at the ICCCM25 can be found here: `https://www.titanmusic.com/icccm2025/slidesandposters/JanjusPoster.pdf`

[7] `https://github.com/ohwjd/ornament_classification`

This rule was chosen as a basic constraint for identifying ornamental figures. The "ornamentation" flag from `abtab`'s `tabmapper`-module also uses this as an underlying assumption, as a note can only be marked as ornamentation if it is the only one at that onset time. For the approach that did not rely on `abtab` annotations, sequences of notes of the same rhythmical value were flagged as such when they contained chords. Because the piece is analyzed chord by chord, ornamental figures in different voices that could overlap cannot be detected in this approach.

3.2.2 Step-wise motion of four notes in the range of a fifth

"Most of the leaps consist of a fifth so that the step-wise motion of four notes can fill the interval." ([7, §2])

This rule was implemented as a filter on the identified ornamental sequences. It first checks the number of core unique onset times of the ornamental figure (excluding the context notes), and only considers sequences with exactly four onset times and step-wise motion (only half or whole steps between consecutive notes). At this point, sequences can still contain onset times with more than one note (so a chord), but those get flagged as such in the process. The next step is checking if the notes run in a monotonic manner or if there are changes in direction. Then, the monotonic sequences get checked if the range between the note on the first onset (or one of the notes on the first onset) of the sequence and one of the post-context notes forms a perfect fifth.

Monotonic ascending sequences will thus lead up +7 half steps (perfect fifth). Because of octave equivalence, though, the final note may drop one octave below leading to an ascending sequence with the final tone being 5 half steps lower than the first (perfect fourth down). The same holds true of course vice-versa for the descending sequence.

3.2.3 Changing the pace too abruptly

"It is wrong, Galilei writes, to use ornaments which move from fusae or semifusae to minims or semibreves." ([7, §4])

To check for this rule, the rhythmic value of the first onset time of the ornamental figure is compared to the rhythmic value of the pre-context note. If the difference in rhythmic value exceeds the threshold defined by Galilei, which is a change from fusae (eighth notes) to semi-breves (whole notes) or semifusae (sixteenth notes) to minims (half notes), the sequence is filtered out as invalid ornamentation. Additionally, to cover cases where there are thirty-second notes involved, the algorithm checks for thirty-second notes moving to quarter notes. To generalize this, the algorithm checks for changes in rhythmic value that exceed two levels of note duration (e.g., from thirty-second notes to quarter notes, or from sixteenth notes to half notes).

3.2.4 First note of an ornamentation is consonant

"The first note of ornaments should not be dissonant with other voices." ([7, §3])

This rule was implemented by checking the interval between the first note of the ornamental figure and the notes sounding in other voices at the same time. The function counts the number of unique ornament sequences that begin with a consonant interval or with a single note. A consonant interval is defined as a unison, minor third, major third, perfect fourth, perfect fifth, minor sixth, major sixth, or octave, evaluated modulo twelve semitones to account for octave equivalence.

3.2.5 No embellishment in succession of chords

"[T]he cadence in which there is a succession of chords in small equal rhythm should not be embellished." ([7, §5])

To implement this rule, a filter was implemented that checks if a sequence of ornament notes contains chords at each onset time of the sequence.

3.3 Evaluation

Due to the lack of a ground truth dataset for ornamentation in lute tablature, the evaluation of the developed algorithm relies on manual evaluation and analysis of the results in two examples. For this, two intabulations of the same vocal piece by Ludwig Senfl (ca. 1490-1543) were chosen from the JOSQUINTAB dataset (see Chapter 3.1 for more details). The goal is to compare the ornamentations identified by the algorithm in both intabulations and see how the intabulations differ in their use of ornamentation and whether the results are consistent between the two pieces. The validated analysis algorithm is then performed on the whole dataset. The two chosen pieces are both found in the same original source published by Hans Newsiedler:[8]

1. *Ach vnfall wes zeihest du mich* (4400_45_ach_unfall_was), pp. 111-114.
2. *Ach vnfal wes zeigstu mich* (4481_49_ach_unfal_wes_zeigst_du_mich)), pp. 396-400.

The song that both intabulations are based on is a contrafactum of Josquin Desprez's (ca. 1450-1521) *Qui belles amours*, which means that Senfl took the original music of Josquin and set a new text to it. Unfortunately, the vocal source by Ludwig Senfl is not preserved fully, so Josquin Desprez's older version was used as the vocal model to compare to in `abtab` (see Chapter 3.1 for more details). Therefore, some minor adjustments were needed in the input files, including the addition of a beat in the second measure so that it is now 3 beats long instead of two. These adaptations were done by the author of `abtab`, Reinier de Valk.

When looking at the beginning of both intabulations (see Figures 3.3 and 3.4), one can see that the first two bars are the same in both pieces, with the same ornamentation

[8] Newsidler, Hans, *Ein neugeordnetes künstliches Lautenbuch.*, Nuremberg: Johan Petreius 1536. `https://opac.rism.info/id/rismid/rism993104150`, `https://opac.rism.info/id/rismid/rism993104151`

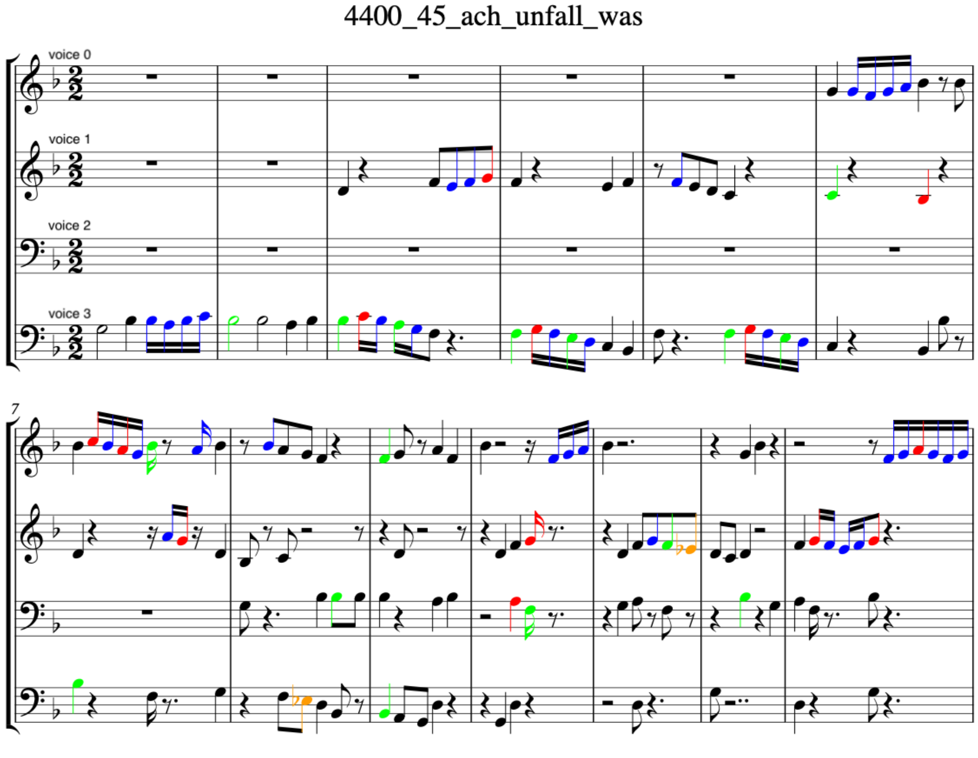

Figure 3.3: The first bars of the first piece in the output of `abtab` with a best guess of voice separation. One can see that there were some ornamentations added by the intabulator. For example, in the lowest voice, there are monotonic falling lines in the third, fourth, and fifth bar. Not all ornamentations were perfectly identified by `abtab` as described in the text.

added. But after that, the pieces diverge in their use of ornamentation with the second one adding smaller notes and quite a bit more ornamentation in general, than the first one. Comparing the pieces throughout, the main melody notes are consistently the same, with the differences mainly being in the added ornamentations, both in frequency and complexity. Whereas the ornamentations in the first intabulation stays relatively simple, the second one adds more elaborate ornamentations. Both pieces are in the same tuning (G), have the same meter (2/2) and they have the same number of bars (56).

Counting the absolute number of identified ornaments in both pieces shows a clear difference in the amount of ornamentation used. In Tables 3.1 and 3.2, the results for both pieces are summarized for the three different approaches described above. When counting the number of identified ornaments, the column "abtab" shows the number of ornaments identified using the annotations from `abtab`'s `tabmapper`. The "raw" column shows the number of ornaments identified using the approach that only relies on finding sequences of notes with the same rhythmical value below a certain threshold. Finally, the "voices_merged" column shows the number of ornaments identified using the

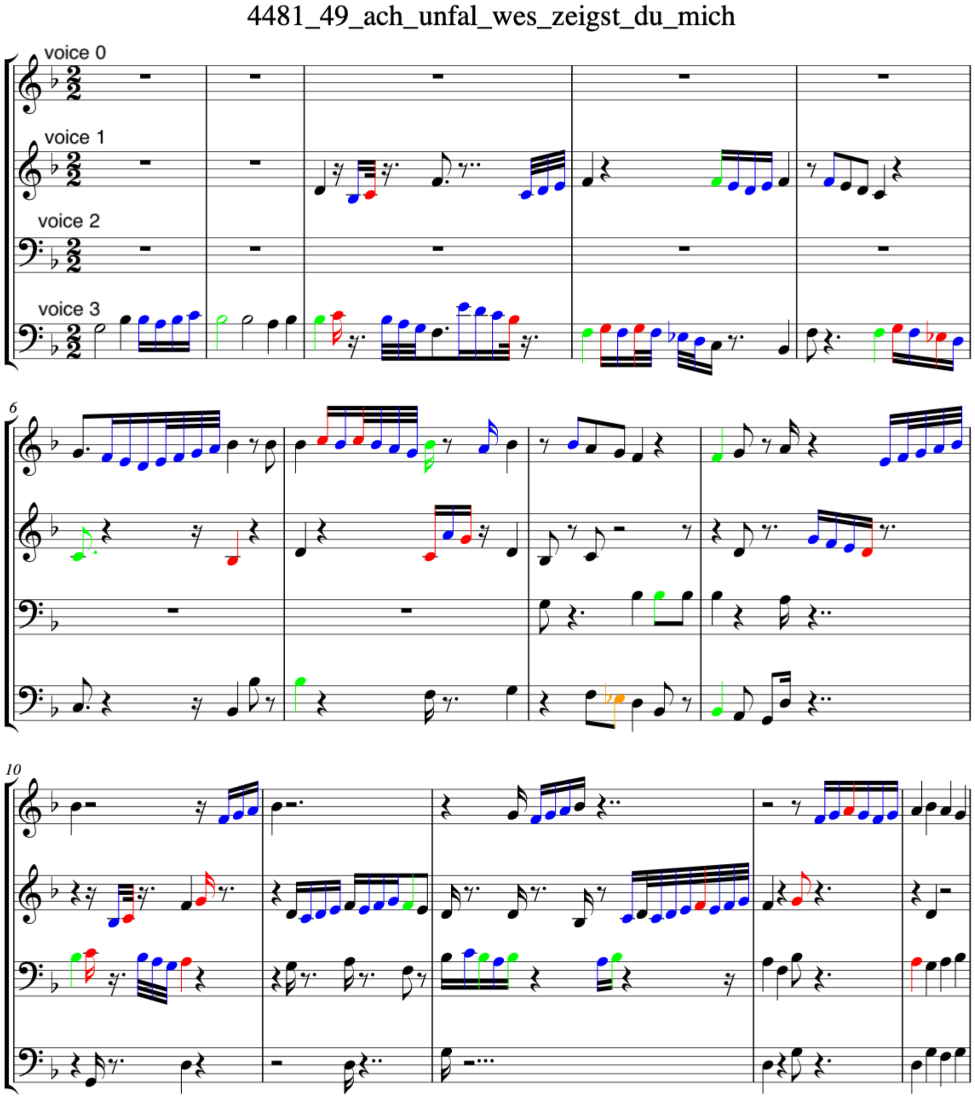

Figure 3.4: The first bars of the second piece in the output of abtab with a best guess of voice separation. One can see that there were some ornamentations added by the intabulator, there are a lot more and smaller notes, compared to the first piece. There are longer lines with movements going up and down, e.g., in measure 6 in the highest voice. Not all ornamentations were perfectly identified by abtab as described in the text.

	count	abrupt dur changes	consonant start	fourstep	non-chord
abtab	45	0	43	16	23
raw	43	0	41	18	19
voices_merged	54	0	–	14	–

Table 3.1: Summary of ornamentation counts, abrupt duration changes and consonant starts for 4400_45_ach_unfall_was.

	count	abrupt dur changes	consonant start	fourstep	non-chord
abtab	80	1	80	31	53
raw	64	1	64	32	35
voices_merged	93	1	–	21	–

Table 3.2: Summary of ornamentation counts, abrupt duration changes and consonant starts for 4481_49_ach_unfal_wes_zeigst_du_mich.

voice-separated approach and summed up the results per voice.

When looking at the summarized results from the first piece in Table 3.1, one can see that the total number of identified ornaments is different for each of the approaches. The "abtab" approach identified 45 ornaments, the "raw" approach found 43, and the "voices_merged" approach detected 54 ornaments. The differences in the number of identified ornaments especially in the "voices_merged" approach can be attributed to the fact that some ornaments were split up into different voices, leading to a higher total count. This may happen when the best guess estimation of abtab's voice separation sorts notes to different voices.

In the second piece, summarized in Table 3.2, the differences between the approaches are even more pronounced due to the higher amount of ornamentation present in the piece. Here, the "abtab" approach identified 80 ornaments, the "raw" approach found 64, and the "voices_merged" approach detected 93 ornaments.

The difference between the "abtab" and "raw" approaches is more significant in the second piece. When looking at the score that has the `abtab`-output marked[9], it can be observed that there are several instances where `abtab` marked single notes in a voice as ornamentation (e.g., measure 3, voice 1 (the second highest); measure 7, voice 1 (the second highest)) and only one or two other notes are part of that ornamentation figure in the same voice. The "raw" approach, on the other hand, relies on identifying sequences of notes with the same rhythmical value, so these single-note ornamentations cannot happen in that approach, leading to a lower count of identified ornaments.

When looking at the results for "abrupt duration changes", corresponding to rule 3.2.3 the second piece has one example of this rule being violated, whereas the first piece has none. Looking at the whole data set, this rule is rarely violated in general, which could indicate that lute players and arrangers were generally careful about not changing the pace too abruptly when adding ornamentation. The "consonant start" column (referring to rule 3.2.4) shows, that almost all identified ornaments in the first piece start with a consonant note, with only two exceptions in both the "abtab" and "raw" approaches. This rule was not evaluated for the "voices_merged" approach, because with a single voice it is not applicable.

Rule 3.2.2 (step-wise motion of four notes in the range of a fifth, "fourstep") is only satisfied by a rather small portion of the identified ornaments in both pieces. This rule is very restrictive, as only sequences with exactly four onset times and step-wise motion are considered valid. Additionally, if the range between the first note of the sequence and one of the post-context notes forms a perfect fifth, the sequence is registered as "perfect fifth". For the first piece, 16 out of 45 identified ornaments in the "abtab" approach satisfy this rule, and 18 out of 43 in the "raw" approach. In the second piece, 31 out of 80 identified ornaments in the "abtab" approach satisfy this rule, and 32 out of 64 in the "raw" approach. The number for the "raw" approach is higher even though the total number of identified ornaments is lower.

A sequence of four notes was categorized as "perfect_fifth" if the range between the first note of the sequence and one of the post-context notes forms a perfect fifth and the sequence is monotonic (either ascending or descending). If no perfect fifth is present, but the sequence is still monotonic, it is categorized as "monotonic". Sequences with changes in direction were categorized as "non-monotonic". The "_chords" suffix indicates that one or more of the core onset times is a chord. This violates the assumption of rule 3.2.5, but these cases were still counted to evaluate how often this occurs in the data.

When looking at the breakdown of different fourstep categories in Figures 3.5, 3.6, 3.7, and 3.8, one can see that in both pieces, the majority of identified fourstep ornaments are monotonic and have a perfect fifth between the first note of the sequence and one of the post-context notes. The different approaches yield quite similar results in all categories for the first piece and similar results for the second piece.

[9] See file in GitHub: `https://github.com/ohwjd/ornament_classification/blob/main/data/4481_49_ach_unfal_wes_zeigst_du_mich-polyphonic.pdf`.

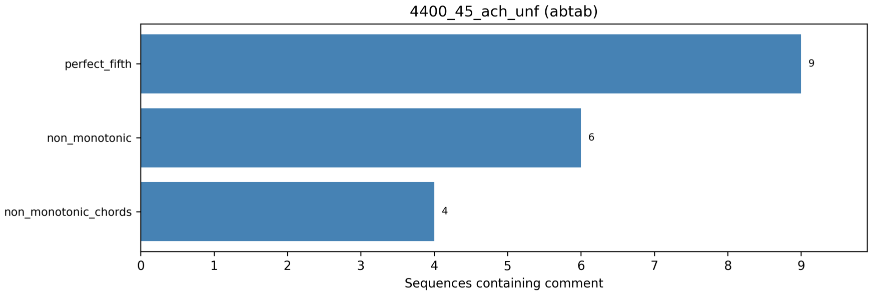

Figure 3.5: Categories of "fourstep"-sequences for 4400__45 ("abtab" approach).

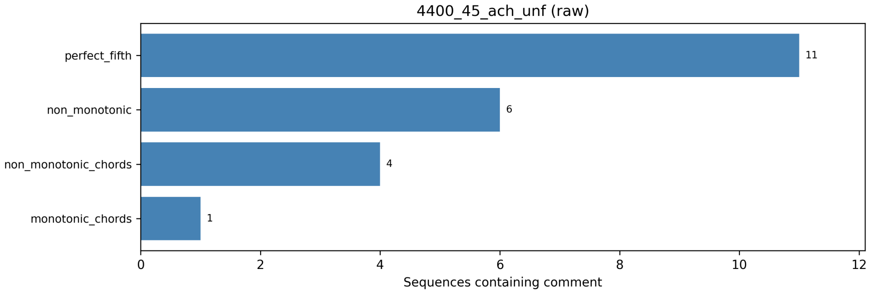

Figure 3.6: Categories of "fourstep"-sequences for 4400__45 ("raw" approach).

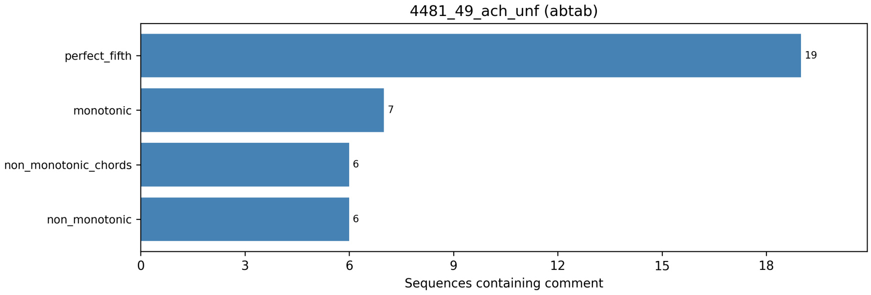

Figure 3.7: Categories of "fourstep"-sequences for 4481__49 ("abtab" approach).

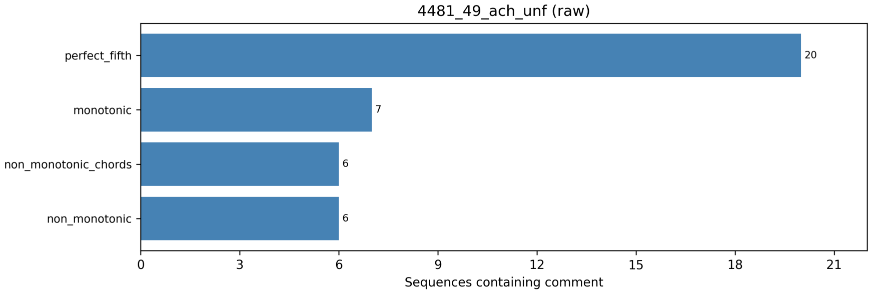

Figure 3.8: Categories of "fourstep"-sequences for 4481__49 ("raw" approach).

In Figure 3.9, a summary of the identified ornaments over all pieces in the data set is shown for both the "abtab" and the "raw" approach. The number of total identified ornaments is higher for the "abtab" approach, which is consistent with the observations made in the two example pieces. The "duration change" rule and the "consonant start" rule are rarely violated in general, indicating that these aspects were generally well adhered to during ornamentation. Of all found ornaments, about 25% of the "abtab" ornaments are a kind of fourstep-variant and about 61% contain no chords. In the "raw" approach, about 38% of found ornaments are fourstep-variants and about 60% do not contain chords.

At this point, I think, it is not possible to decide which approach gives the best results. The evaluation needs to be done with properly annotated test data to decide on an approach.

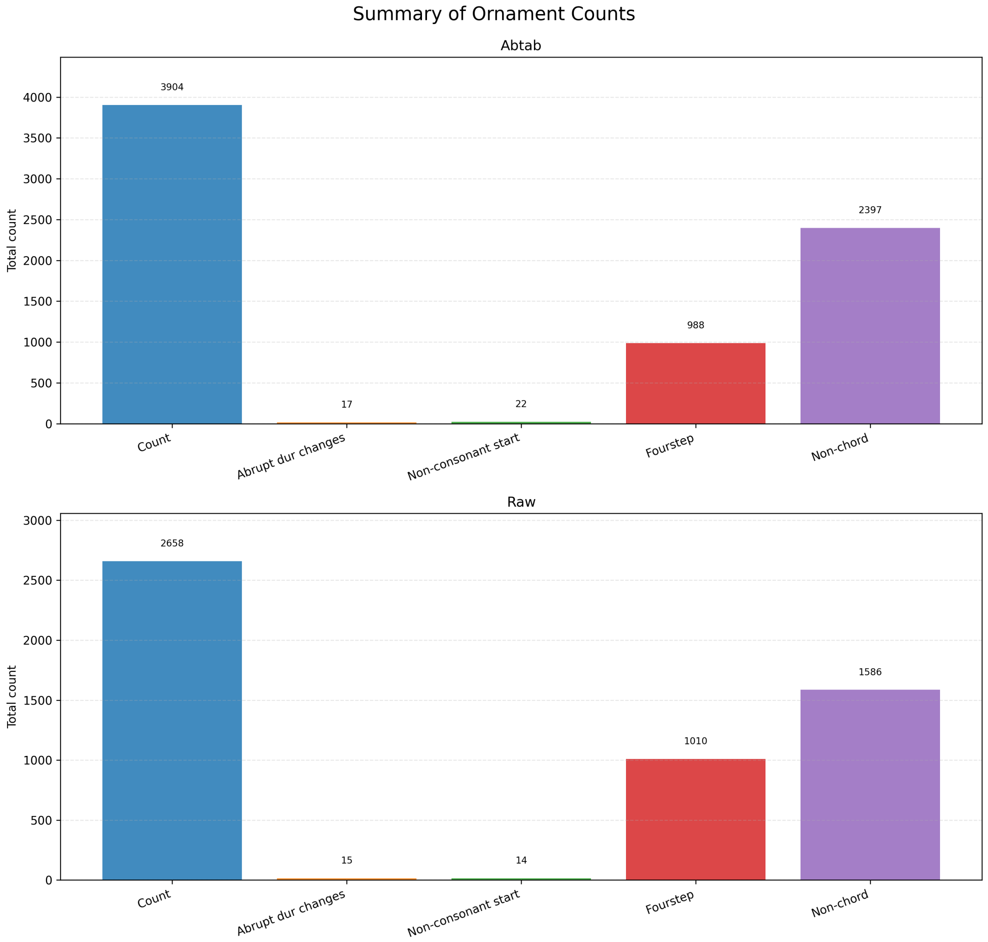

Figure 3.9: Summary of ornaments over all pieces of the data set for the "abtab" and the "raw" approach. Compared to the overall number of identified ornaments, the number of "abrupt duration changes" and "non-consonant start" is very low.

CHAPTER 4

Conclusion & Future Work

As outlined in the introduction (1), music information retrieval has seen considerable progress in recent years. The present work adopted a rule-based approach to experiment with ornament classification, due to the absence of properly annotated corpora as well as ground truths.

Despite the existence of several digitization projects in early music, annotated data is not really available. There is some machine-readable tablature available, also due to the efforts of the E-LAUTE project. Ornament labels are not available, and the material is confined to formats such as tablature MEI or `.tpb` with limited software support. Attempts to translate the corpus into Humdrum or other formats failed for exactly these reasons. Any rigorous assessment of the current system will therefore depend on future data sets that combine consistent encodings with verified annotations of ornaments.

Within this constrained landscape, the Galilei rules (see Section 2.1.4) serve as an exploration rather than a definitive description of ornamentation found in lute tablature. They capture Italian lute practice of the late Renaissance, yet the pieces investigated here may only partially follow those prescriptions, also maybe because the repertoire is mostly from German sources in German lute tablature.

Several open tasks remain: first of all, a ground truth data set needs to be created to enable proper evaluation of the implemented rules. Meter changes such as 2/2 to 3/1 that may disturb the diminution factor need to be accommodated, and confronting the implementation with additional treatises, including Judenkünig (see Section 2.1.3), especially when extending the E-LAUTE corpus toward German-speaking sources, would need to be considered. Furthermore, easier and better visualization of the output in the score would also be useful for musicologists. Graph-based representations promise more contextual information once sufficient material is available, and graph neural networks could leverage those structures effectively.

These observations point to a deeper, familiar gap. Digital humanities projects still struggle with patchy corpora, and computer science projects often miss the musical detail that scholars expect. Working in both areas convinced me that collaboration can be very rewarding, but it also reminded me how much effort it takes to agree on shared language, methods, and tools.

The algorithm developed here is therefore positioned as an aid rather than a replacement for the musicologist. Ornamentations are identified and counted and help to tackle repetitive tasks, but the interpretive authority remains with musicological experts. A visualization of the results in the score would be a next step to facilitate expert assessment and further refinement of the rules. Strengthening interdisciplinary dialogue, improving data availability, and refining the rule base step by step will be essential to deepen computational support for historical music research.

Overview of Generative AI Tools Used

During the work in this thesis, some AI tools were used to support various aspects. GitHub Copilot in Visual Studio Code with the model GPT-5-Codex (Preview) was used to assist in the implementation of the python code. For writing the thesis, Visual Studio Code with LaTeX inline-suggestions turned on was used, to support writing LaTeX commands. ChatGPT, Claude AI and Deep-L were used to translate single words or phrases from German to English.

List of Figures

List of Tables

Bibliography

[1] J. Rodin, “Searching the Notes—All of Them: The Josquin Research Project,” in *The Oxford Handbook of Music and Corpus Studies* (D. Shanahan, J. A. Burgoyne, and I. Quinn, eds.), Oxford University Press, 2022. 2

[2] K. Dorfmüller, “Issues in Transcribing German Lute Tablature,” *Soundboard Scholar*, vol. 8, no. 1, pp. 1–14, 2022. 3, 5, 6, 8

[3] H. Minamino, “An Invisible Notation: On the Invention of German Lute Tablature,” *Discoveries: South-Central Renaissance Conference News and Notes*, vol. 17, no. 2, pp. 3,13, 2000. 3

[4] W. Apel, *The Notation of Polyphonic Music, 900-1600.* Cambridge, Mass., Mediaeval Academy of America, 1961. 6

[5] R. de Valk, R. Ahmed, and T. Crawford, “JosquIntab: A Dataset for Content-based Computational Analysis of Music in Lute Tablature,” in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, (Delft, The Netherlands), pp. 431–438, ISMIR, Nov. 2019. 10, 15, 16

[6] H. M. Brown, “Intabulation,” in *Grove Music*, Oxford University Press, 2001. 10

[7] H. Minamino, *Sixteenth-Century Lute Treatises with Emphasis on Process and Techniques of Intabulation.* PhD thesis, University of Chicago, 1988. 10, 11, 18, 19, 20

[8] E. Cambouropoulos and M. Kaliakatsos-Papakostas, “Symbolic Approaches and Methods for Analyzing Musical Similarity: Representation and Pattern Processing in Harmony,” in *The Oxford Handbook of Music and Corpus Studies* (D. Shanahan, J. A. Burgoyne, and I. Quinn, eds.), Oxford University Press, 2022. 12

[9] K. Lemström, “String Matching Techniques for Music Retrieval,” Dec. 2000. 12

[10] D. Meredith, “A parallel, geometric algorithm for transformed pattern matching in polyphonic music,” *15th International Workshop on Machine Learning and Music*, Sept. 2024. 12, 13

[11] R. Typke, F. Wiering, and R. C. Veltkamp, “A Search Method for Notated Polyphonic Music with Pitch and Tempo Fluctuations,” in *ISMIR 2004 Conference Proceedings*, pp. 281–288, 2004. 12, 13

[12] C. Romming and E. Selfridge-Field, *Algorithms for Polyphonic Music Retrieval: The Hausdorff Metric and Geometric Hashing.* Jan. 2007. 12, 13

[13] M. Laitinen and K. Lemström, “Dynamic Programming in Transposition and Time-Warp Invariant Polyphonic Content-Based Music Retrieval,” in *12th International Society for Music Information Retrieval Conference*, 2011. 13

[14] S. Doraisamy and S. Rüger, “Robust Polyphonic Music Retrieval with N-grams,” *Journal of Intelligent Information Systems*, vol. 21, pp. 53–70, July 2003. 13

[15] G. A. Wiggins, K. Lemström, and D. Meredith, “SIA(M)ESE: An Algorithm for Transposition Invariant, Polyphonic Content-Based Music Retrieval,” 2002. 13

[16] E. Karystinaios and G. Widmer, “GraphMuse: A Library for Symbolic Music Graph Processing,” July 2024. 14

[17] J. J. Stutter, “Musical reuse in the Notre Dame repertory: Historiography and new computational directions,” 2025. 14