Automatic harmonic analysis using the tonal graph

Master's Thesis (non confidential)

Sébastien LI

Supervisors :
Gonzalo ROMERO-GARCIA : gonzalo.romero@ircam.fr
Carlos AGON : carlos.agon@ircam.fr
Gérard ASSAYAG : gerard.assayag@ircam.fr

Host organization :
IRCAM (Équipe Représentations musicales)

01/03/2024 - 09/08/2024

Abstract

English

This master's thesis extends the work of Gonzalo Romero-Garcia [14], with a specific focus on developing a framework for analyzing harmony in musical compositions. Understanding harmony is a crucial part of understanding the complexities of music, and roman numeral analysis is a widely used method among musicologists. This type of annotation gives information about chord's local tonality, root degree, quality, inversion, and any secondary tonalities (or tonicizations). Romero-Garcia's research tackles the task of automatic harmonic analysis, and this work further refines his methodology, enhancing its robustness when applied to complex musical pieces while preserving its high level of interpretability.

Keywords: Music, Harmonic analysis, Roman numeral analysis, Computational musicology, Mathematical morphology, Graph theory

Français

Ce mémoire de master prolonge le travail de Gonzalo Romero-Garcia [14], en se concentrant spécifiquement sur l'élaboration d'un cadre d'analyse de l'harmonie dans les compositions musicales. Comprendre l'harmonie est essentiel pour saisir la complexité de la musique, et l'analyse en chiffrage romain est un outil largement utilisé par les musicologues. Ce type d'annotation fournit des informations sur la tonalité locale d'un accord, le degré de sa fondamentale, sa qualité, son renversement, ainsi que sur les éventuelles tonalités secondaires (ou tonicisations). Le travail de Romero-Garcia aborde la tâche de l'analyse harmonique automatique, et ce travail améliore sa méthodologie en la rendant plus robuste lorsqu'elle est appliquée à des œuvres musicales complexes, tout en préservant la grande interprétabilité de cette méthode.

Mots clés: Musique, Analyse harmonique, Analyse en chiffrage romain, Musicologie computationnelle, Morphologie mathématique, Théorie des graphes

Contents

1 Introduction

Understanding harmony is fundamental in understanding the complexities of music, as it reveals how tonal compositions create expectations, tensions, and resolutions. Despite centuries of musicologists studying tonal harmony, only a fraction of musical pieces have undergone thorough analysis. Therefore, automating harmonic analysis aims to unlock access to annotations for every existing and future composition. By compiling a vast annotated dataset, the applications of such analysis extend widely, such as serving educational purposes, correcting enharmonic spelling errors during MIDI conversions, helping the composition of arrangements, or enhancing music generation techniques.

For example, one could want to learn about the history of chord progressions or explore how composers handled key changes. As these questions improve our understanding of harmonic theory, music learners could benefit from the automation of harmonic analysis. Additionally, in the realm of machine learning, where classification drives inference, the ability to automate harmonic analysis becomes invaluable, especially for conditional music generation tasks.

There are various approaches to analyze the harmony within a musical piece. In jazz and pop genres, chord symbols are the norm, denoting each chord by its root, quality and eventual bass (e.g., E♭maj7/G). While practical for performers sight-reading, this method offers little insight into the chord's functional role with neighboring chords. Another technique, Riemannian functional harmony, assigns each chord to a function relative to a key : tonic (T), dominant (D), or subdominant (S), with a typical progression of $S \rightarrow D \rightarrow T$. While it facilitates comprehension of chord relations, its lack of detail on individual chords often necessitates supplementation with other notations. Roman numeral analysis, prevalent in classical music, emerges as the most common method. Chord symbols can easily be derived from Roman numeral analysis, and, as explained by Tymoczko [17], "[function] labels cannot be translated into Roman numerals or absolute chord labels. However, function labels can often be recovered from Roman numerals. [...] The asymmetry gives us reason to prefer Roman numerals for analytical corpora.". Therefore, Roman numeral analysis serves as the harmonic analysis method for our approach.

In this method, each chord is annotated with five elements:

- Local Tonality: Indicates the primary key of the chord, annotated only when it changes during modulation or at the beginning. For example, "Fm:" signifies that the local tonality is F minor until the next modulation.
- Degree of the Root: Denoted by a Roman numeral ranging from I to VII, representing the relation of the root note to the local tonality. If the root contains an accidental alteration, it is indicated before the Roman numeral. For instance, "♯vi" indicates a raised sixth degree.

- Quality of the Chord: Uppercase numerals represent major chords, while lowercase numerals represent minor chords. Augmented chords are denoted by an uppercase numeral followed by +, diminished chords by a lowercase numeral followed by °, and half-diminished seventh chords by a lowercase numeral followed by ø.

- Inversion of the Chord: Specifies the bass note of the chord relative to the root. For example, root position, first inversion, and second inversion of a triad are annotated as 5, 6, $^{6}_{4}$ respectively.

- Tonicization: Describes the relation of the chord's root to a secondary key other than the primary key, used when borrowing a chord from another key without a modulation. It is indicated by a slash followed by the degree of the secondary key in Roman numerals. For instance, "V/vi" signifies a chord borrowed from the vi of the primary key, where the root is the V of the secondary key. For example, in this case, if the primary key is C major, then the secondary key is A minor and the chord is E major (which contains G♯, not present in the key of C major).

With these elements, we understand the harmonic function of the chord while retaining information about its constituent notes. For instance, a chord progression like E♭:V6/V-V7-I provides the functional understanding of E♭:S-D-T. Meanwhile, it also informs us of the individual chords: F Maj/A - B♭ Maj - E♭ Maj.

2 State of the art

Automating Roman numeral analysis poses a significant challenge due to the contextual dependency in chord notation. Each chord's representation relies on both the local key and its relationship to the root, resulting in numerous potential notations for a single chord, with typically only one being plausible. In certain chord progressions, multiple interpretations may arise, leading to divergent opinions among musicologists. Moreover, choosing the harmonic rhythm (the rate at which chords change) is an essential and complex task. Filtering out passing notes (notes that do not belong to the underlying chord) is also a necessary but sometimes ambiguous task. As this section will show, early approaches often sidestepped the challenges of rhythmic segmentation and key-finding by employing a simplistic, uniform harmonic rhythm or focusing on non-modulating pieces. Previous research on harmonic automation predominantly falls into two categories: rule-based, and machine learning techniques.

2.1 Rule based methods

Winograd's 1968 research [18] marked the pioneering effort in automatic harmonic analysis, using established natural language processing techniques to study Johann Sebastian Bach's chorales. However, his method involved a manual conversion of the score, providing a list of perfect four-note chords, which means that human preprocessing was necessary for rhythmic segmentation and non-harmonic note filtering.

In 1992, Maxwell [9] introduced another approach for analyzing Bach's chorales, yet also relying on manually-instructed rhythm segmentations. His algorithm operated on fifty-five hand-made rules, demonstrating great accuracy for specific examples. However, because of the arbitrary nature of the rules and their subjective associated weights, it struggled with generalization, particularly when encountering arpeggiation, two-note chords, and ornaments. Consequently, despite being pioneers, Winograd's and Maxwell's algorithms faced challenges in addressing basic issues in harmonic analysis.

In his research in 1997, Temperley [15] proposed a method that aims to identify the key and root of chords alongside the harmonic rhythm. This approach relies on the spatial representation of Tonal Pitch Class (TPC), which distinguishes enharmonic notes with different spellings (such as F♯ and G♭), as opposed to the Neutral Pitch Class (NPC), which is more closely tied to perceived frequencies (treating F♯ and G♭ as equivalent). In Temperley's framework, pitches are positioned at varying distances along the line of fifths (for example, C is closer to G than to D). To find an analysis, Temperley employs five distinct rules:

1. The pitch variance rule determines the spelling of notes by selecting the Tonal Pitch Class closer to the center of gravity of recent pitches along the line of fifths.

2. The compatibility rule prioritizes the selection of a root by assigning a priority list based on the pitch's relationship to the root: $1 > 5 > 3 > \flat 3 > \flat 7 >$ other. This implies that for a chord with notes G and B, the selected root is G instead of E because the pitch relationship (1 3) takes precedence over (3 5).
3. The strong beat rule emphasizes segmentation on strong beats.
4. The harmonic variance rule favors roots that are closely positioned along the line of fifths.
5. The ornamental dissonance rule favors non-harmonic pitches when they are separated by seconds.

This approach demonstrates improved generalization compared to prior methods, culminating in the creation of the Melisma Music Analyzer program in 2001.

In 2007, Illescas [4] proposed a method that directly applies Riemann's functional harmony principles. The approach breaks down the task into small steps. First, it analyzes the melody to identify non-harmonic notes, categorizing them into various passing tones based on rules considering beat strength and intervals. Next, each measure is uniformly segmented. Subsequently, chords are selected using intervals, and the key is determined by considering accidental alterations. Then, possible tonal functions of the chord are established. Finally, the best analysis is chosen from an acyclic directed graph, with edge weights determined by hand-made rules.

Those examples of rule based methods highlight the intuitive nature of such algorithms mirroring how musicians approach harmonic analysis. However, they often struggle to generalize on large datasets.

2.2 Machine learning

The data-driven approaches to harmonic analysis usually fall into two type of methods: either probabilistic models or deep learning models, with the latter being the study of most recent researches.

2.2.1 Probabilistic models

In 2004, Raphael [13] introduced a Hidden Markov Model (HMM) to analyze harmony in music pieces divided into uniform segments and encoded with Neutral Pitch Classes. He argued against considering pitch spelling, asserting that listeners could interpret harmony solely based on the frequencies played. The use of a Markov chain was justified musically, as each harmonic label could depend only on the preceding one, expressed as $p(x_{n+1}|x_1, ...x_n) = p(x_{n+1}|x_n)$. Raphael suggested that extending the model could involve retrieving voice leading information.

Temperley [16] extended this suggestion by implementing it into his Bayesian probabilistic model in 2009. His objective was to maximize the expression $P(M, H, S|N)$, where M represents the metrical structure, H stands for harmony, S denotes stream structure (melodic segregation), and N refers to the analyzed note. This method initially focuses on stream structure, applying logical rules to the streams while assuming a flat metrical structure and harmony. Then, it simultaneously searches the optimal metrical structure and harmony. This process involves considering various levels of beat grids to achieve the desired outcome.

2.2.2 Deep learning models

Kröger [6] introduced a machine learning framework in 2009 that tackles the chord labeling task. The study conducts a comparative analysis, contrasting previously mentioned rule-based algorithms with various fundamental architectures of decision trees and neural networks. However, this framework falls short in addressing the challenges associated with rhythm segmentation and key finding.

Building upon Kröger's early work and the growing interest in machine learning for audio tasks [10], Chen [1] introduced a deep learning architecture for Roman numeral analysis. This architecture leverages the Long Short-Term Memory (LSTM) network's ability to capture long-term dependencies in sequential data, commonly utilized in natural language processing tasks. Chen employs a Bidirectional LSTM (BLSTM) block, incorporating both forward and backward LSTMs, enabling the model to consider both past and future chord contexts. The output includes the five elements of Roman numeral analysis. However, the model disregards harmonic rhythm. Chen's work served as a catalyst, inspiring subsequent researchers to explore deep learning techniques for addressing similar challenges in music analysis.

In 2020, Micchi [11] advanced Chen's model by incorporating Gated Recurrent Units (GRUs) instead of LSTMs and integrating convolutional layers to accommodate rhythmic metrics. Expanding the training dataset to include compositions such as Mozart and Beethoven Themes and Variations [2], Bach preludes [17], Beethoven string quartets [12], and various romantic songs enhanced model robustness.

In 2021, Lopez introduced AugmentedNet [8], further enhancing Micchi's architecture by concatenating results from convolutional blocks, enabling simultaneous consideration of all rhythmic metrics. Lopez also introduces additional tasks to the Roman numeral analysis, increasing output redundancy to enhance the model's understanding of the task. Notably, one output classifies the 75 most common Roman numerals, and another output gives the harmonic rhythm. The dataset is expanded to include Haydn's String Quartets [7].

In 2023, Karystinaios [5] introduced a novel approach by encoding music scores as graphs, with each note represented as a node and edges denoting temporal connections. Utilizing a Graph Neural Network (GNN), the model generates of these graphs before passing the resulting arrays to a GRU for Roman numeral prediction, while incorporating additional tasks proposed by Lopez. This innovative method offers a unique perspective on music representation and analysis, potentially giving new insights into the underlying structure and relationships within musical compositions.

2.3 A note on data-driven automatic harmonic analysis

The shift towards machine learning in harmonic analysis brings promising results. However, while these techniques offer powerful tools for automatic analysis, musicologists caution against overlooking inherent ambiguities in harmonic progressions. Unlike in some other fields where ground truth is more clearly defined, harmonic analysis lacks a unique, definitive solution.

The few existing datasets often provide only one or two analysis of a piece, disregarding potential ambiguities. Relying only on these interpretations for algorithm training risks oversimplifying the complexity of musical harmony.

For instance, when faced with conflicting annotations of the same progression, algorithms may struggle to generalize effectively, leading to oscillations between different results without achieving true understanding.

Another significant drawback of deep learning techniques in harmonic analysis is the lack of interpretability of intermediate results. Unlike traditional musicological approaches, where analysts can articulate and justify their interpretations based on theoretical principles and context, deep learning algorithms operate as black boxes, making it challenging to understand how they come to their conclusions. This lack of transparency undermines the essence of analyzing complex tonal pieces, as the algorithm's decisions cannot be interpreted and critiqued in a meaningful way.

Overcoming these limitations requires developing methods to extract and visualize the reasoning behind the algorithm's choices, enabling a more transparent and interpretable approach to automatic harmonic analysis.

3 The tonal graph

The *tonal graph*, developed by Gonzalo Romero-Garcia, is a novel and effective approach to harmonic analysis that also provides high interpretability.

3.1 Presentation of the tonal graph

A tonal graph (V, E) is a chain of bipartite graph where the vertices V represent possible Roman numerals of one piece of music, and the edges E connect nodes that are temporally successive. Its goal is to find the most plausible roman analysis of the piece by building a path $(V_1, V_2, ..., V_n)$ (see Figure 3.1).

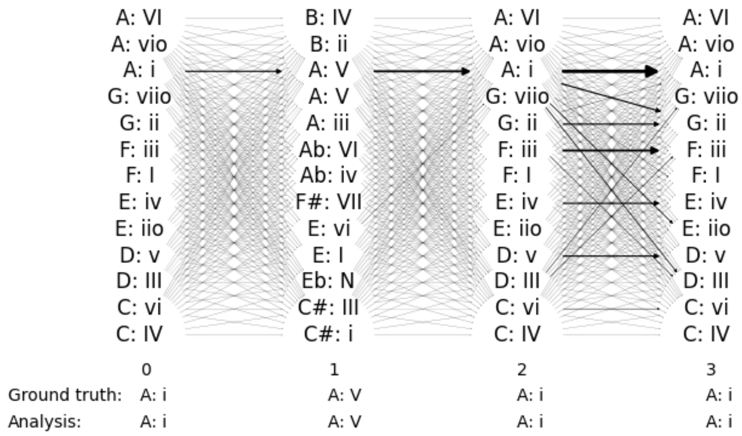

Figure 3.1: Example of a tonal graph for the first measure of the chorale *Ach wie flüchtig, ach wie nichtig* by Johann Sebastian Bach

3.1.1 Construction of the tonal graph

This graph is built upon a representation of the musical score in the form of a piano roll, which is a 2D binary array P of size $128 \times D$ where the first axis represents the MIDI height of the notes and the other axis represents the time. This piano roll is condensed into a collapsed chroma roll S of size $12 \times T$ where $T = \frac{D}{\tau}$ and τ is a temporal collapse factor chosen so that the temporal resolution of the chroma roll is the selected uniform harmonic rhythm of the piece (see Figure 3.2).

$\forall f \in [\![0, 11]\!], \forall t_0 \in [\![0, T]\!],$

$$S[f, t_0] = \begin{cases} 1 & \text{if } \exists m \in \mathbb{N}, \exists t \in [\![t_0\tau, (t_0+1)\tau[\![, P[f + 12m, t] = 1 \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

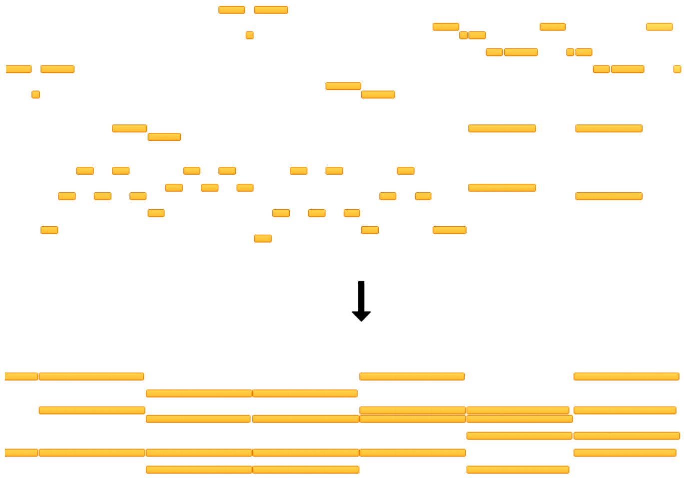

Figure 3.2: The piano roll and the chroma roll of the first measures of the Sonata no. 5 in G major by Wolfgang Amadeus Mozart

Possible Roman numerals are derived from this chroma roll with a morphological binary erosion. This operation acts as a detector of the presence a structuring element.

In our context, a structuring element $R \in \{0,1\}^{12}$ is constructed from a Roman numeral. For instance, the Roman numeral V, which is the major chord built upon the dominant of the key and which contains the notes that are 2, 7 and 11 semi-tones above the tonic, is associated with the structuring element $R_{\text{V}} = [0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1]$.

Thus, if S is the input and R is the structuring element, the erosion of S by R denoted by $S \ominus R$ and belonging to the space $\{0,1\}^{12\times T}$ is defined by

$$(S \ominus R)[f, t] = \begin{cases} 1 & \text{if } R + f \leq S_t \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

where $(R + f)[j] = R[j + f \bmod 12]$ is the circular shift of R by f and S_t is the t-th column of S.

An *activation array* A of size $I \times 12 \times T$ is constructed by stacking the results of binary erosion with the structuring elements $R_1, R_2, \cdots, R_I$ which represent I manually selected and translated Roman numerals shown in table 3.1.

The array is defined as:

$$A[i, f, t] = (S \ominus R_i)[f, t] \quad (3.3)$$

RN	Repr.	Mode	RN	Repr.	Mode
I	$(0, 4, 7)$	M	**v**	$(7, 10, 2)$	m
i	$(0, 3, 7)$	m	**vi**	$(9, 0, 4)$	M
ii	$(2, 5, 9)$	M	**viº**	$(9, 0, 3)$	m
iio	$(2, 5, 8)$	m	**VI**	$(8, 0, 3)$	m
iii	$(4, 7, 11)$	M	**viiº**	$(11, 2, 5)$	M, m
III	$(3, 7, 10)$	m	**VII**	$(10, 2, 5)$	m
III+	$(3, 7, 11)$	m	**N**	$(1, 5, 8)$	m
IV	$(5, 9, 0)$	M	**Fr**	$(8, 0, 2, 6)$	m
iv	$(5, 8, 0)$	m	**Ger**	$(8, 0, 3, 6)$	m
V	$(7, 11, 2)$	M, m	**It**	$(8, 0, 6)$	m

Table 3.1: List of Roman numerals, their integer representation and the mode(s) with which they are associated.

The nodes of the tonal graph correspond to the indexes (i, f, t) where A is nonzero. Here, $f \in [\![0, 11]\!]$ indicates the pitch class of the tonic of the key (0 = C, 1 = C♯, $\cdots$, 11 = B). The index i is the index in the list of Roman numeral candidates. The variable t represents time in the uniform grid. In the graph, a node is denoted as f: L_i, where L_i is the label of the Roman numeral, specifying its figure and quality.

For instance, if a score contains a C major chord at time t_0, the candidates could include C: I, as well as G: IV and F: V. The tonal graph would then include the nodes $(\text{I}, 0, t_0)$, $(\text{IV}, 7, t_0)$, and $(\text{V}, 5, t_0)$.

To create the edges of the graph, a chain of bipartite graphs is constructed by linking all nodes with a time component of t to all nodes with a time component of $t + 1$. The weight of the edges indicates the presence of tonality modulation:

$$e_{(i_1, f_1, t), (i_2, f_2, t+1)} = \begin{cases} 0 & \text{if } f_1 = f_2 \\ 1 & \text{otherwise} \end{cases} \tag{3.4}$$

With this method of calculating edge weights, finding a shortest path in the graph corresponds to identifying an analysis with the minimal amount of modulation. The shortest path algorithm can be written in linear complexity as the tonal graph is constructed as a Directed Acyclic Graph.

3.2 Limitations of the tonal graph

Despite its advantages, using the tonal graph has several shortcomings that need to be addressed.

3.2.1 Enharmonic ambiguities

Describing pitch only by its chromatic information instead of its standard notation is a loss of information, and can sometimes lead to ambiguity. Musicians analyze enharmonically equivalent pitches differently when they are notated distinctly. The most notable example is the fully diminished seventh chord.

For example, when built upon a C, the pitch classes are (0, 3, 6, 9). However, this chord can be notated in numerous different ways: (C, E♭, G♭, B𝄫), (C, E♭, G♭, A♮), (C, E♭, F♯, A), and (C, D♯, F♯, A), and each way has its own specific analysis (the chords are respectively in the key of D♭ minor, B♭ minor, G minor and E minor)

3.2.2 Piano Roll temporal inefficiency

The size of the piano roll P, representing the entire musical piece, is $128 \times D$, where D depends on the piece's duration and the time resolution. This temporal representation can often be highly inefficient. To accurately represent all possible rhythms, the time resolution must be a common divisor of all note lengths. This requirement results in a large array and consequently long computation times (see Figure 3.3).

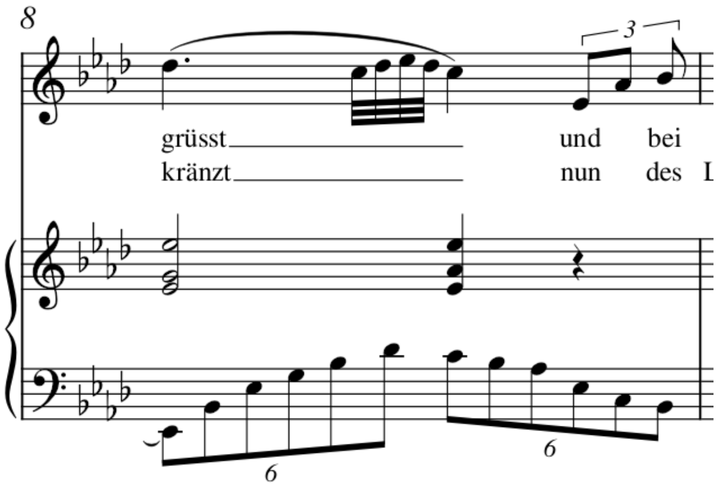

Figure 3.3: Because of the presence of thirty-second notes and tuplets in the 8th measure of *Wenn der Abendstern die Rosen* composed by Emilie Mayer, every measure of the piece must be divided into ninety-sixth notes.

3.2.3 Chords with missing notes

Due to the nature of morphological erosion, this method cannot detect chords with missing notes. For example, if the chord at time t_0 consists (C, E), the Roman numeral **I** does not activate because $S_{t_0} = [1,0,0,0,1,0,0,0,0,0,0,0]$ and $R = [1,0,0,0,1,0,0,1,0,0,0,0]$. No f exists such that $R_f \leq S_{t_0}$.

3.2.4 Non harmonic tones

Musical pieces are not composed of only harmonic notes. When notes outside the intended chord are present, this can lead to the activation of too many unwanted interpretations (see Figure 3.4).

An example would be a melody that rapidly runs through the entire C major scale, which would trigger the activation of all the Roman numerals associated with the C major tonality.

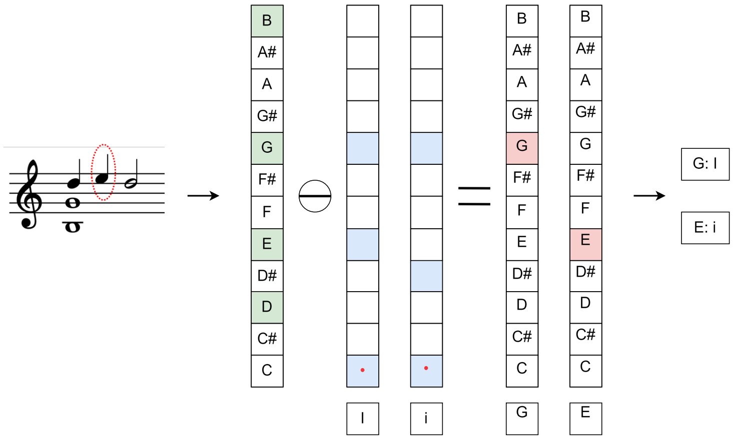

Figure 3.4: The presence of an E in this G major chord adds an unwanted E minor analysis

3.2.5 Harmonic Rhythm

A glaring issue of this method is the lack of automatic harmonic rhythm recognition. This means that a uniform harmonic rhythm is assumed, and shall be individually decided for each piece to analyze.

In the example illustrated by figure 3.5, the first measure necessitates an analysis in eighth notes. However, the second measure requires an analysis in whole notes. Attempting to analyze the first eighth note of the second measure (C B C) would not yield the correct analysis.

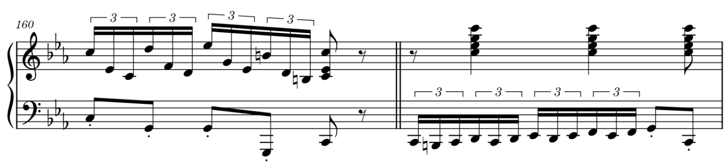

Figure 3.5: Excerpt from *32 Variations on an Original Theme in C minor* by Ludwig van Beethoven

3.2.6 Complex modulations

Choosing the correct analysis by selecting the path with minimum amount of modulation is satisfactory with harmonically simple pieces. Nevertheless, this method requires a more robust comprehension of harmony as it is prone to errors when analyzing some pieces.

Typically, when the modulations are short, there is not enough information about the local tonalities to correctly select when they modulate, and where they modulate to.

In the example illustrated in figure 3.6, an incorrect analysis consisting of (f♯: III → C: ii → I → IV → G♯: VI → C: vi → V → I) could possibly be selected as it has 4 modulations, which is the minimum possible. Here, choosing to modulate to F♯ minor and G♯ major is absurd because the tonalities are harmonically too distant to C major. Because of the lack of information on music theory principles, such analysis fails to grasp the V → I sequence of this excerpt.

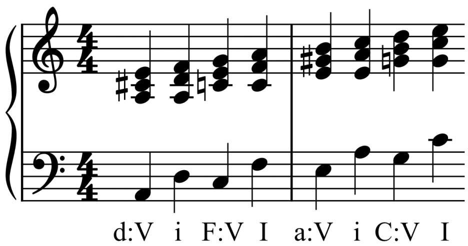

Figure 3.6: A classic modulating sequence with an example of analysis

4 Contributions

The primary objective of this internship was to rectify the limitations of the current methodology. The harmony analysis process has been restructured into three distinct subtasks: chord root detection, rhythm segmentation, and tonal analysis. This approach not only reduces the burden on the shortest path method but also incorporates music theory rules, all while preserving a high level of interpretability (see Figure 4.1).

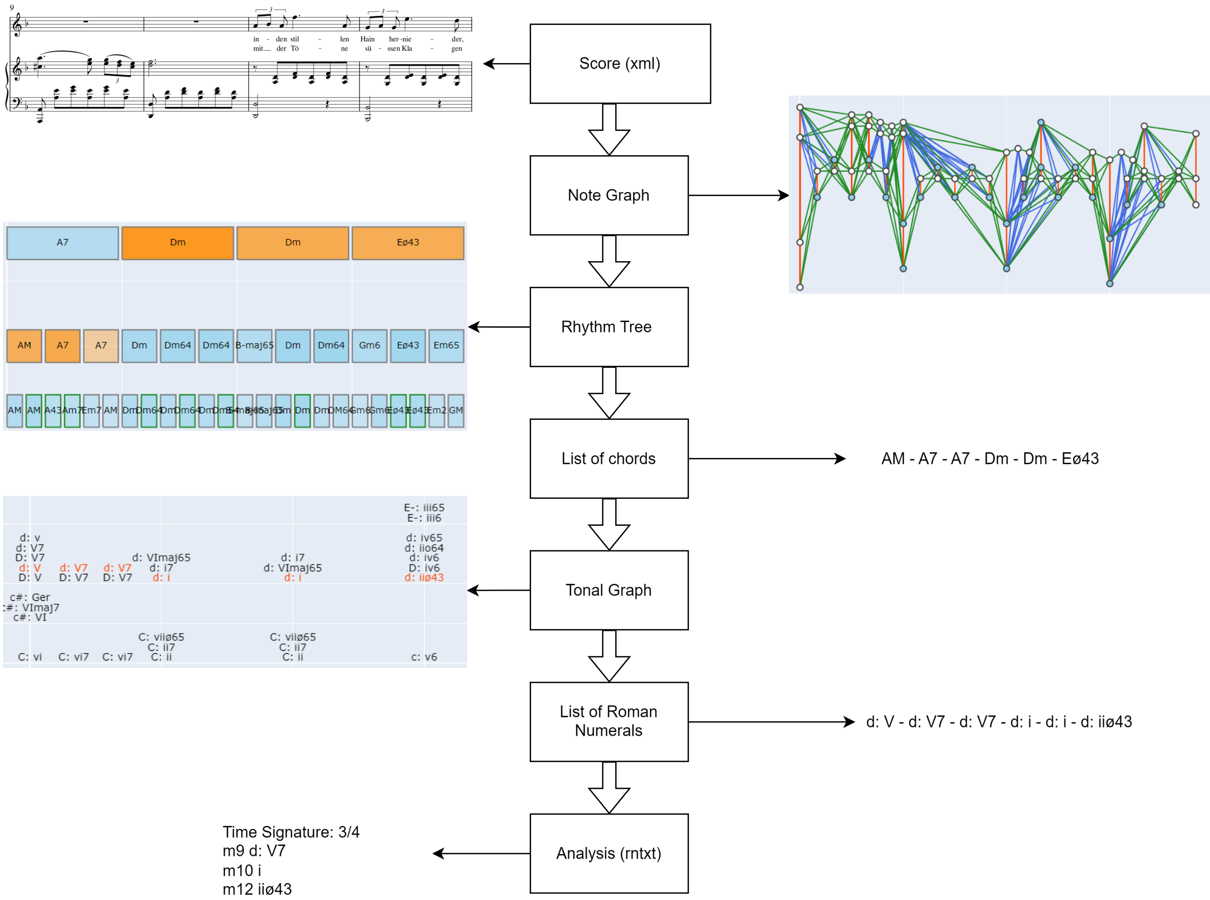

Figure 4.1: Algorithm structure for 4 measures of *Ständchen* by Franz Schubert

4.1 New symbolic representations of the musical score

To accomplish an effective harmonic analysis, a symbolic representation of music was employed to tackle the issues explained in the previous section.

4.1.1 The pitch space

The diatonic/chromatic space $\mathbb{Z}_7 \times \mathbb{Z}_{12}$ can be used to represent note height in order to address the issue of enharmonic ambiguity. By combining the chromatic space with the diatonic space, pitches, as well as intervals, are treated in the same way as in music theory: the pitch / interval name encodes the diatonic space, while the number of semitones encodes the chromatic space (see Tables 4.1 and 4.2).

For example, F♯ is encoded by (3, 6), G♭ is encoded by (4, 6), an augmented sixth is encoded by (5, 10) and a minor seventh is encoded by (6, 10). This allows the distinction between a German chord in C minor (A♭, C, E♭, F♯) and a dominant seventh chord in D♭major (A♭, C, E♭, G♭).

	0	1	2	3	4	5	6	7	8	9	10	11
6	B♯	B𝄪								B𝄫	B♭	**B**
5								A𝄫	A♭	**A**	A♯	A𝄪
4						G𝄫	G♭	**G**	G♯	G𝄪		
3				F𝄫	F♭	**F**	F♯	F𝄪				
2			E𝄫	E♭	**E**	E♯	E𝄪					
1	D𝄫	D♭	**D**	D♯	D𝄪							
0	**C**	C♯	C𝄪								C𝄫	C♭

Table 4.1: The diatonic / chromatic space for notes with 2 or less alterations

	0	1	2	3	4	5	6	7	8	9	10	11
6	A7									d7	m7	M7
5								d6	m6	M6	A6	
4							d5	P5	A5			
3					d4	P4	A4					
2			d3	m3	M3	A3						
1	d2	m2	M2	A2								
0	P1	A1										d1

Table 4.2: The diatonic / chromatic space for intervals

4.1.2 The Note Graph

An interesting representation of the musical score is the graph proposed by Karystinaios [5], which will be referred to as the note graph $N = (V, E)$.

The vertices represent the notes, with all the various types of associated information such as the pitch, the time of onset, the duration, the dynamics, the articulation, the part assignments, the lyrics, and more.

The temporal relationship between notes are described more directly as the directed edge $e_{u,v}$ connects nodes u and v if they meet the following conditions:

- *Onset* edge : u and v start at the same time and u is lower-pitched than v
- *During* edge: v starts while u is sounding
- *Follow* edge: v starts when u ends
- *Silence* edge: v starts after u ends and no notes are in-between

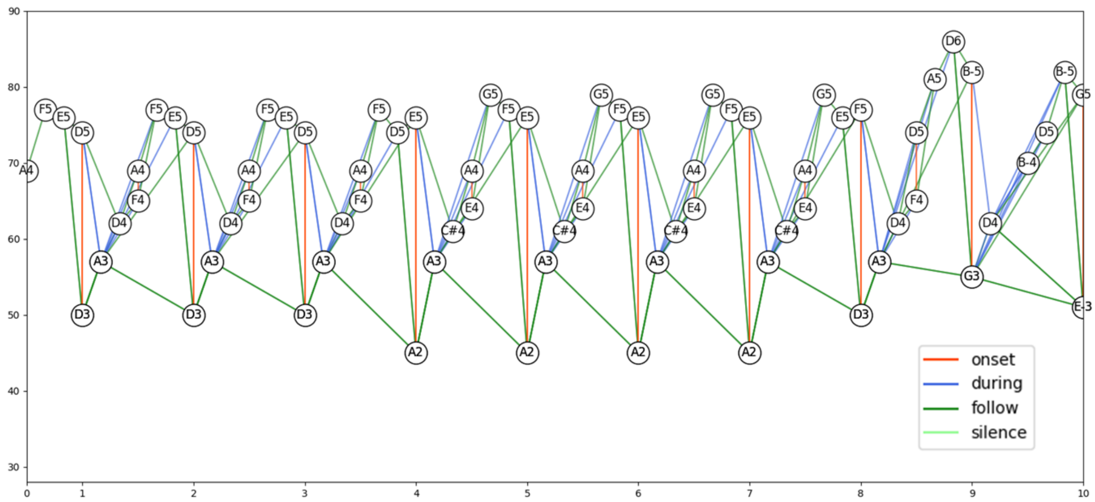

Figure 4.2: Note graph for the first measures of *Sonata N°17, 3rd Mvt., Ludwig van Beethoven*

This graph is a Directed Acyclic Graph, which results to algorithms with low time complexities.

4.2 Chord root detection with similarity levels

Due to the shortcomings of the morphological erosion discussed in section 3.2.3, a different algorithm for chord detection is proposed.

Let W be a time window for the analysis. A subgraph of the note graph $N_W = (V_W, E_W)$ is created so that it corresponds to the notes played during the time window W. Every existing chord is given a score $S_W \in [0, 1]$ that indicates the similarity level between the subgraph and the chord.

A chord is indexed by $(d, c, i_{quality})$ where $d \in [\![0, 6]\!]$ is the diatonic class and $c \in [\![0, 11]\!]$ is chromatic class of the root of the chord, and $i_{quality} \in [\![0, 11]\!]$ is the index of the chord quality in a manually specified family of eleven chord qualities $Q_0, Q_1, \cdots, Q_{11}$ (specified in Table 4.3).

Index	Name	Symbol	Root	Third	Fifth	Seventh
0	Major triad	M	(0,0)	(2,4)	(4,7)	
1	Minor triad	m	(0,0)	(2,3)	(4,7)	
2	Diminished triad	o	(0,0)	(2,3)	(4,6)	
3	Augmented triad	+	(0,0)	(2,4)	(4,8)	
4	Major seventh	maj7	(0,0)	(2,4)	(4,7)	(6,11)
5	Minor seventh	m7	(0,0)	(2,3)	(4,7)	(6,10)
6	Dominant seventh	7	(0,0)	(2,4)	(4,7)	(6,10)
7	Diminished seventh	o7	(0,0)	(2,3)	(4,6)	(6, 9)
8	Half-diminished seventh	ø7	(0,0)	(2,3)	(4,6)	(6,10)
9	Italian augmented sixth [1]	It	(0,0)	(2,2)	(4,6)	
10	French augmented sixth	Fr	(0,0)	(2,4)	(4,6)	(6,10)
11	German augmented sixth	Ger	(0,0)	(2,2)	(4,6)	(6,9)

Table 4.3: Intervals of the selected qualities in the diatonic/chromatic space

Let $C(d, c, i_{quality}) \subseteq \mathbb{Z}_7 \times \mathbb{Z}_{12}$ be the set of notes of the chord indexed by $(d, c, i_{quality})$. For example, $C(2, 3, 1)$ would correspond to the chord D♯minor. Thus, $C(2, 3, 1) = \{(2, 3), (4, 6), (6, 10)\}$

An interesting similarity chord score that handles missing notes and unharmonic notes would be the Intersection over Union (IoU) measure, or formally:

$$S_W^{IoU}(d, c, i_{quality}) = \frac{|C(d, c, i_{quality}) \cap V_W^{pitches}|}{|C(d, c, i_{quality}) \cup V_W^{pitches}|} \tag{4.1}$$

where $V_W^{pitches}$ is the set of pitches of the played notes during the analyzed time window W.

We can rewrite:

$$|C(d, c, i_{quality}) \cap V_W^{pitches}| = \sum_{p \in C(d,c,i_{quality})} \mathbb{1}(p \in V_W^{pitches}) \tag{4.2}$$

By replacing $\mathbb{1}(p \in V_W^{pitches})$ with a weight $w_W(p) \in [0, 1]$ that measures the harmonic importance of the played pitches, we can inject information about music theory. We use the following established principles observed in classical music, which are confirmed in the experiment section 5.1.1:

- Notes that are doubled tend to be more likely harmonic.
- High-pitched notes are generally more likely to be inharmonic.
- Shorter notes are more likely to be inharmonic.

[1] Augmented sixth chords usually do not have a root. For the sake of the example, the root was chosen so that the chord is constructed by a third, a fifth, and an eventual seventh above the root

Thus, for each pitch p, the weight $w_W(p)$ can be divided into three weights:

- $w_{doubling}(doub(V_W, p))$ that depends on the number of times the pitch p is played. This number is defined as:

  $doub(V_W, p) = |\{u \in V_W | u_{pitch} = p\}|$

  This weight function $w_{doubling} \in [0,1]^{\mathbb{N}}$ must be increasing to ensure that the more times a pitch is played, the more it has harmonic relevance. The function must also ensure that $w_{doubling}(0) = 0$, so that the two other weights do not need to be computed if p is not played.

- $w_{octave}(oct(V_W, p))$ that depends on the octave of the lowest played note that has the same pitch as p. This octave number is defined as:

  $oct(V_W, p) = \min(\{u_{octave} | u \in V_W \wedge u_{pitch} = p\})$

  This weight function $w_{octave} \in [0,1]^{\mathbb{N}}$ must be decreasing to ensure that lower pitches have higher harmonic importance.

- $w_{duration}(dur(V_W, p))$ that depends on the relative duration of the longest played note that has the same pitch as p. This relative duration is defined as:

  $dur(V_W, p) = \max(\{u_{duration}/W_{duration} | u \in V_W \wedge u_{pitch} = p\})$

  This weight function $w_{duration} \in [0,1]^{[0,1]}$ must be increasing to ensure that longer pitches have higher harmonic importance.

Empirically satisfying functions that were used are:

$$w_{doubling}(n) = min(1, \sqrt{\frac{n}{3}})$$
$$w_{octave}(n) = \frac{1}{1 + e^{n-6}}$$
$$w_{duration}(x) = \sqrt{x}$$

The complete formula for S_W can be written:

$$S_W(d, c, i_{quality}) = \frac{\sum_{p \in C(d,c,i_{quality})} w_{doubling}(doub(V_W, p)) \cdot w_{octave}(oct(V_W, p)) \cdot w_{duration}(dur(V_W, p))}{|C(d, c, i_{quality}) \cup V_W^{pitches}|} \tag{4.3}$$

Additionally, we have found that more principles of classical music can easily be implemented to create a more robust algorithm [2]:

1. **Leap rule**: Non-harmonic notes are either approached or left by intervals smaller than a third. This can be restated as: any note approached and left by intervals equal to or larger than a third must be harmonic.

2. **Onset rule**: If three or more different pitches are played at the same time and they are pitches of a chord, and if the analysis window is shorter than the played notes, then the underlying chord is likely to be this chord for the analysis window.

The first principle is implemented by examining the edges E of the note graph $N = (V, E)$ and labeling as *leap vertices* all vertices $u \in V$ which all incoming and outgoing edges have an interval greater or equal to a third. Given time window W, for each leap vertex $u_{leap} \in V_W$, the score S_W is modified so that chords that do not contain the pitch of u_{leap} give a zero score.

For example, in the excerpt of Mozart's *Piano Sonata N°1* illustrated in Figure 4.3, the circled E is a leap note because it is both approached and left by thirds. Therefore, time windows W_1 and W_3 must have their score S_W updated so that all chords that do not include the note E have a score of zero.

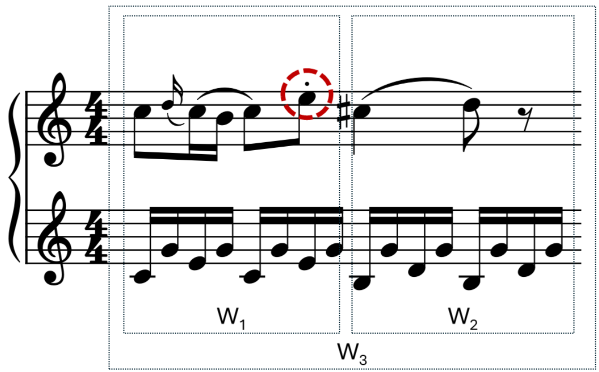

Figure 4.3: In this measure of *Piano Sonata N°1* by Wolfgang Amadeus Mozart, the circled E is a leap note.

It can be noted that all notes of the left hand are considered as leap notes, and thus, time window W_3 should have zero scores for all chords that do not include C, D, E, G and B. Since, no chord includes all these notes, $S_{W_3} = 0$, making any analysis on time window W_3 inadequate. In this example, the implementation of this principle also helps with the harmonic rhythm segmentation since W_3 cannot be a selected analysis time window.

The second principle is implemented by examining the *onset* and *during* edges of the note graph N and thus examining all notes that are played at the same time. If these notes are

[2]While these rules are not absolute and further refinement is needed to address exceptions, their empirical application is generally advantageous

composed of three different pitches or more, we check the list of all the chords that contain all the pitches of the played notes. If this list is empty, we do not update S_W. Otherwise, we update S_W so that only these chords have a non-zero score. This is done only if the analysis time window W is included in the time window of the played notes.

For example, if a C, an E, and a G are played at the same time during the whole time window W, the underlying chord could only be CM, Cmaj7, C7, Am7 or A♯Ger. Thus, every S_W would be zero for other chords.

Therefore, thanks to this framework that implements various principles of music, the algorithm can easily be updated based on the needs and the context of the music genre.

4.3 Harmonic rhythm segmentation with the Rhythm Tree

The challenge of selecting the appropriate harmonic rhythm segmentation is equivalent to determining the correct time windows W for analysis. To address this, a method based on the rhythm tree is proposed.

The rhythm tree of a musical score is a tree that maps possible harmonic rhythm segmentations of the score. Each node of the tree is associated with a time window that starts at time t_{start} and ends at time t_{end} and its children are the time windows derived from analyzing the piece at a lower rhythm subdivision as illustrated in Figure 4.4.

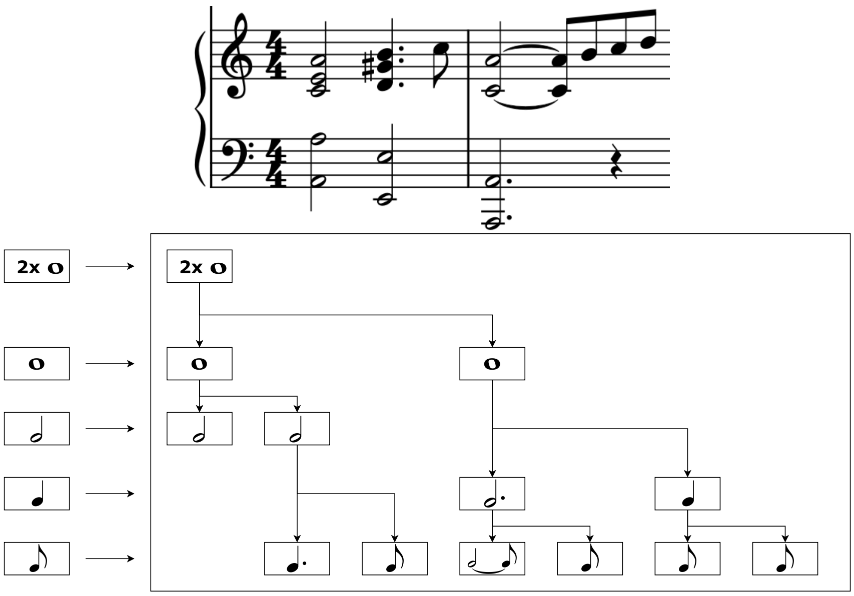

Figure 4.4: The rhythm tree for the first two measures of the *Sonata for Arpeggione and Piano* by Franz Schubert. Nodes at depth 1 represent segmentations on whole-notes, nodes at depth 2 on half-notes, depth 3 on quarter-notes and depth 4 on eighth-notes.

Formally, let W be the time window that represents the note subgraph $N_W = (V_W, E_W)$ at time $[t_{start}, t_{end}[$ and that subdivides the score at rhythm R_i. We define

$$T = \{t \in [t_{start}, t_{end}[| \exists u \in V_W, t = u_{start} \wedge (t - t_{start}) = 0 \text{ mod } R_{i+1}\} \tag{4.4}$$

where R_{i+1} is the rhythm that subdivides R_i (half-notes subdivide whole-notes, quarter-notes subdivide half-notes, etc.).

Given the set $T = \{t_1, ..., t_{n-1}, t_n\}$, the children of a node corresponding to the time window W are constructed as nodes corresponding to time windows W_i that begin at time t_i and end at time t_{i+1}, where i ranges from 1 to $n - 1$.

As discussed in section 4.2, each time window W is analyzed, and for each chord $C(d, c, i_{quality})$, a score S_W is assigned. The challenge lies in selecting which nodes of the rhythm tree should be retained for further analysis.

For any node n associated with time window W that has children $\{n_1, n_2, ..., n_i\}$, we define:

$$best(n) = \max_{d,c,i_{quality}} (S_W(d, c, i_{quality})) \tag{4.5}$$

$$s(n) = \begin{cases} best(n) & \text{if } n \text{ has no child} \\ \max(mean(s(n_1), \cdots, s(n_i)), mean(best(n_1), \cdots, best(n_1))) & \text{otherwise} \end{cases} \tag{4.6}$$

Algorithme 1 : Algorithm for the Rhythm Tree Node Selection

```
Function NodeSelection(n):
    if best(n) ≥ s(n) then
        n_selected ⟵ True;
    else
        n_selected ⟵ False;
        foreach n_i ∈ children of n do
            NodeSelection(n_i)
        end
    end
End Function
```

The intuition behind this method is to recursively select nodes that have a higher score than a mean score of their respective children. We have empirically found that the geometric mean function produces satisfying harmonic segmentation.

We now have a set of selected nodes $\{\hat{n}_1, \hat{n}_2, \cdots, \hat{n}_N\}$, which corresponding time windows $\{\hat{W}_1, \hat{W}_2, \cdots, \hat{W}_N\}$ cover the entire musical piece. Each of these nodes has an associated score $S_{\hat{W}_i}(d, c, i_{quality})$ for each chord $C(d, c, i_{quality})$.

4.4 Incorporation of the Tonal Graph

For each selected time window $\hat{W}_i$, multiple nodes of the tonal graph need to be constructed. This involves considering every chord $C(d, c, i_{quality})$ that has a nonzero score $S_{\hat{W}_i}(d, c, i_{quality})$. However, to optimize computational efficiency, we can focus on only the chords with the highest scores without significantly compromising accuracy. For the experiments, the chords with the top five scores were used.

For each chord $C(d, c, i_{quality})$ that is considered, all the Roman numerals that describe this chord are translated into nodes of the tonal graph. Each node u is assigned a weight $w_u = 1 - S_{\hat{W}_i}(d, c, i_{quality})$. Edges are created to connect all the nodes of time window $\hat{W}_i$ to the nodes of time window $\hat{W}_{i+1}$.

As explained in Section 3.2.6, the edge weights in the tonal graph need to be selected to reflect principles of music theory. A proposed formula for calculating the edge weight $w_{u,v}$ connecting nodes u and v considers the following parameters:

- **Distance between keys:** The distance $\mathscr{D}(u_{key}, v_{key})$ measures the distance between the key associated with node u and the key associated with node v.
- **Chord similarity levels:** The node score w_u represent how well the notes of the musical piece align with the chord for node u.
- **Rarity of Roman numerals:** The coefficient $f_{rn}(u)$ depends on the rarity of the Roman numerals, with more common numerals like I typically having a lower value compared to less common ones like III.
- **Rarity of Roman numeral transitions:**[3] The coefficient $f_{transition}$ depends on the rarity of transitions between two Roman numerals. For instance, the V $\rightarrow$ I transition is common in classical music and would therefore have a lower value.

Formally, we define:

$$w_{u,v} = f_{transition}(u, v) \cdot (c_{key}\mathscr{D}(u_{key}, v_{key}) + c_{chord}\frac{w_u \cdot f_{rn}(u) + w_v \cdot f_{rn}(v)}{2}) \tag{4.7}$$

where c_{key} and c_{chord} are positive real coefficients that indicate the relative importance of their respective parameters in the formula. $f_{transition}$, f_{rn} and $\mathscr{D}$ are functions that are detailed in the folowing sections.

[3] Musicologists consider that first-order transitions are insufficient to fully describe the complexities of music. To achieve a more accurate representation, it may be necessary to incorporate structural information, which may involve analyzing longer sequences of chords and considering hierarchical relationships.

4.4.1 Distance between two keys

To compute the distance $\mathscr{D}$ between two keys, a graph of keys is proposed to accurately represent the geometric space of musical keys. We define a non-oriented, weighted graph $K = (V, E)$ where:

- **Vertices**: Each element of $\mathbb{Z}_7 \times \mathbb{Z}_{12} \times \{major, minor\}$ is associated with a vertex u and is therefore noted as (d_u, c_u, m_u). Here:
  - $d_u \in \mathbb{Z}_7$ denotes the diatonic class of the tonic of the key
  - $c_u \in \mathbb{Z}_{12}$ denotes the chromatic class of the tonic of the key
  - m_u indicates the mode of the key, either "major" or "minor".
- **Edges**: The edges in the set E reflect relationships between keys based on common modulation types. An edge $e_{u,v}$ is created according to the conditions in Table 4.4. The edge weights can be empirically chosen based on the edge types.

Edge type	Diatonic int.	Chromatic int.	Mode of u	Mode of v
Neighbor keys	$d_v - d_u = 4$	$c_v - c_u = 7$	$m_u = m_v$	
Relative keys	$d_v - d_u = 2$	$c_v - c_u = 3$	$m_u = minor$	$m_v = major$
Parallel keys	$d_v = d_u$	$c_v = c_u$	$m_u \neq m_v$	
Enharmonic equivalent	$d_v - d_u = 1$	$c_v = c_u$	$m_u = m_v$	
Minor dominant key	$d_v - d_u = 4$	$c_v - c_u = 7$	$m_u = minor$	$m_v = major$

Table 4.4: Conditions for the construction of edges of the graph of keys

For example, the node associated with the key of C major would have the following neighbors: G major, F major, A minor, C minor, D𝄫 major, B♯major, F minor.

Therefore, we define the key distance $\mathscr{D}(key_1, key_2)$ as the length of the shortest path in this graph of keys between the node u that represents key_1 and the node v that represents key_2. An interactive dashboard was developed to help visualize the graph and the influence of the weights.

In the example illustrated by Figure 4.5, the edge weights that were used are [4]: $w_{neighbor} = 1$, $w_{relative} = 0.7$, $w_{parallel} = 1.3$, $w_{enharmonic} = 0.01$, $w_{dominant} = 1.2$. Selecting the key of E major and G♭minor results in 6 different shortest paths of length 1.71. The one shown in the illustration is composed of the following modulations: E major $\rightarrow$ C♯minor (relative minor) $\rightarrow$ D♭minor (enharmonic) $\rightarrow$ G♭minor (neighbor).

The distance $\mathscr{D}$ is invariant by translation, or formally:

$$\forall (d, c) \in \mathbb{Z}_7 \times \mathbb{Z}_{12}, \mathscr{D}((d_1, c_1, m_1), (d_2, c_2, m_2)) = \mathscr{D}((d_1+d, c_1+c, m_1), (d_2+d, c_2+c, m_2))) \quad (4.8)$$

[4] These weights are also used for the experiments.

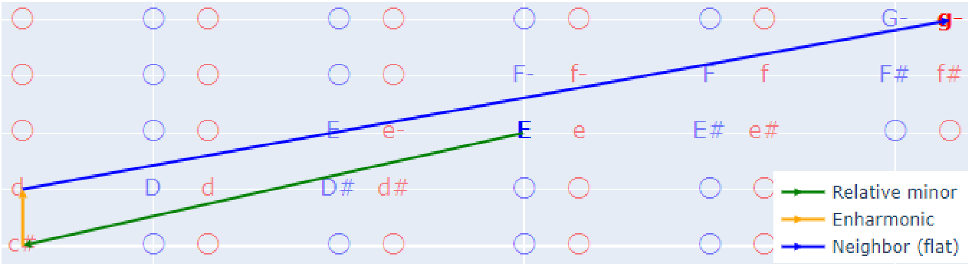

Figure 4.5: One of the shortest path between the key of E major and G♭minor. For the sake of visibility, keys that have more than 2 alterations in their name are represented by circles.

Therefore,

$$\mathscr{D}((d_1, c_1, m_1), (d_2, c_2, m_2)) = \hat{\mathscr{D}}((d2 - d1, c2 - c1, m1, m2)) \tag{4.9}$$

where $\hat{\mathscr{D}}((d, c, m1, m2)) = \mathscr{D}((0, 0, m1), (d, c, m2))$ is a distance that can be pre-computed and stored in a $7 \times 12 \times 2 \times 2$ array (size 336). This substitution prevents the storage of an array of shape $7 \times 12 \times 2 \times 7 \times 12 \times 2$ (size 28224).

4.4.2 Transition weight and Roman numeral weight

Determining the correct $f_{transition}$ and f_{rn} can be approached using data-driven methods. This involves analyzing pieces from the dataset with initially set parameters (either random or manually chosen) and then adjusting $f_{transition}$ and f_{rn} based on a learning rate to minimize discrepancies between the analysis and the dataset (see results in Figure 4.6).

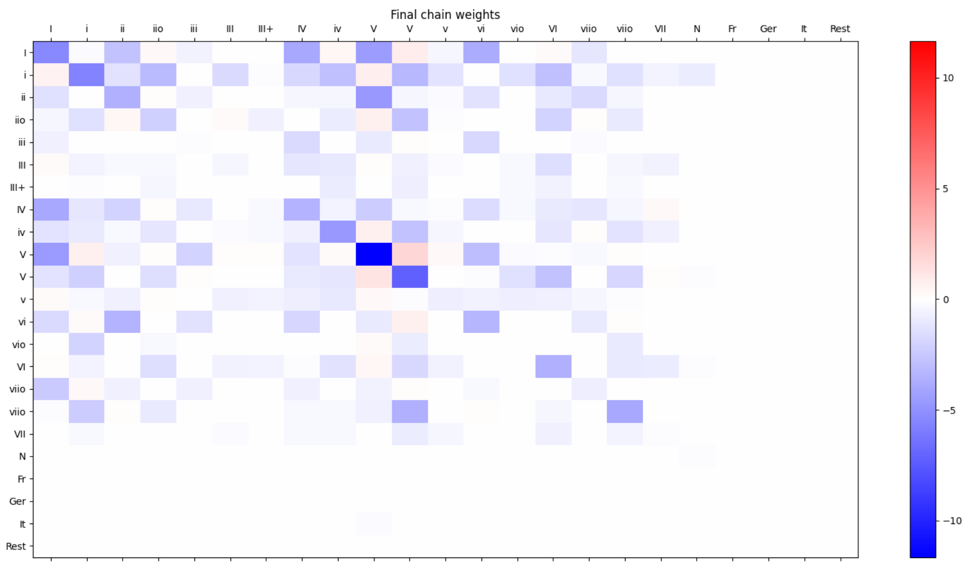

Figure 4.6: $f_{transition}$ after training on a dataset of Bach's chorales. As expected, the weights are low for transitions V → V and V → I

This method was applied to a dataset of Bach's chorales with some success, achieving an

Mode	Roman numeral	f_{rn}	Mode	Roman numeral	f_{rn}
Major	I	1	Minor	i	1
Major	Imaj7	0.5	Minor	i7	0.8
Major	ii	0.99	Minor	N	0.95
Major	ii7	0.99	Minor	iio	0.99
Major	iii	0.8	Minor	iiø7	0.99
Major	iii7	0.5	Minor	III+	0.9
Major	iv	0.9	Minor	III	0.6
Major	IV	0.99	Minor	IIImaj7	0.4
Major	IVmaj7	0.8	Minor	iv	0.99
Major	V	0.99	Minor	iv7	0.8
Major	V7	0.99	Minor	V	0.99
Major	vi	0.99	Minor	V7	0.99
Major	vi7	0.8	Minor	v	0.8
Major	viio	0.99	Minor	VI	0.99
Major	viiø7	0.99	Minor	VImaj7	0.6
Minor	It	0.9	Minor	viio	0.99
Minor	Ger	0.9	Minor	viio7	0.99
Minor	Fr	0.9	Minor	VII	0.8

Table 4.5: Suggested values for f_{rn}

Chord 1	Chord 2	$f_{transition}$
V7	I	0.9
V	I	0.94
V7	i	0.9
V	i	0.94
viiø7	I	0.95
viio	I	0.94
viio7	i	0.95
viio	i	0.94
other	other	1

Table 4.6: Suggested values for $f_{transition}$

accuracy of 78.64%. However, due to difficulties in generalizing to a broader dataset, as well as the challenges discussed in sections 2.3 and 5, we have found that manually selected coefficients based on music theory knowledge are preferable. A suggestion of these values are presented in tables 4.5 and 4.6. These values can be adjusted according to the needs and music genres.

With this revised definition of the edge weights of $e_{u,v}$, the correct analysis is still obtained by applying a shortest path algorithm to the tonal graph.

5 Experiments

The Challenge of Evaluating Harmonic Analyses

Harmonic analysis presents inherent challenges in evaluation due to the abundance of ambiguities. Multiple valid analyses can arise from the same musical piece, depending on the hierarchical perspective adopted during the analysis. This variability highlights the complexity of assessing harmonic interpretations and the influence of different analytical frameworks.

For example, in the excerpt in B♭major illustrated in figure 5.1, one approach to analyzing the first measure might involve interpreting it as a modulation to G minor with a Roman numeral every eighth-note (g: i6 V43 i V6). Alternatively, another perspective could be to view the passage as a two-measure sequence that supports a five-measure cycle of fifths (D - g - C - F - B♭). This approach would maintain the key of B♭major and assign one Roman numeral per measure (B♭: V6/vi vi V6/V V I).

Figure 5.1: Excerpt from "Das Wandern", *Die schöne Müllerin* by Franz Schubert with the two valid analyses.

The first analysis represents a more localized approach, where the focus is on immediate harmonic relationships within a smaller segment of the music. In contrast, the second analysis provides a more global perspective, interpreting the passage in terms of broader harmonic cycles. Choosing one type of analysis over the other (which could be labeled as a "ground truth" in a dataset) should not be equated with an error in the same way as a clearly incorrect analysis.

Another common example of harmonic ambiguity arises in cases involving very long modulations. In such scenarios, the analysis can diverge significantly based on where one chooses to identify the points of modulation.

This issue is a common critique of using Roman numerals as a harmonic analysis tool, as it highlights the lack of a universally agreed-upon approach for resolving such ambiguities. The choice between local and global perspectives, and the inherent flexibility in interpretation, reflects the complexity and subjectivity involved in harmonic analysis. There is no clear-cut solution to this problem, as the effectiveness of Roman numerals in capturing harmonic relationships often depends on the specific context and analytical goals.

Nevertheless, the advantage of our proposed framework lies in its flexibility, which is achieved through a comprehensive set of interpretable and customizable rules. This allows a user to not only operate the framework but also to finely adjust its parameters and approach. By tweaking these rules, the user could tailor the framework to better suit specific analytical needs or preferences, enabling a more precise and personalized analysis of musical pieces.

5.1 The When-in-Rome Corpus

For the experiments, a subset of the When-in-Rome dataset [3] was used.

5.1.1 Presentation of the dataset

The dataset consists of around 2 000 harmonic analyses of around 1 500 distinct works, spanning from Baroque to contemporary classical music. Due to the stylistic diversity, we have chosen to focus exclusively on pieces composed during the Classical era (see Figure 5.2). Each piece is categorized by genre (Orchestral, Quartets, Piano Sonatas, etc.), and includes information on the composer, the piece's title, and its movements. The analyses are provided in the RomanText format [17], while the musical scores are available in .xml, .mxl, .krn, or .mscz formats. For implementation purposes, only pieces available in .xml or .mxl formats were selected for the experiments.

To support the validity of equation 4.2, a verification was performed to assess the claim that harmonic notes are generally lower-pitched and shorter in duration. The figures 5.3 and 5.4 appear to confirm these assertions.

5.1.2 Limitations of the dataset

In addition to the issues discussed in section 5, which are common to all harmonic analysis datasets, there are instances where the analyses given in the When-in-Rome dataset deviate from standard practices. For instance, in the theme exposition of Beethoven's *9 Variations on a March by Dressler* (see Figure 5.5) which begins in C minor, the When-in-Rome dataset's analysis does not follow the expected modulation to E♭major in the fourth measure. Instead, it employs a prolonged tonicization of III.

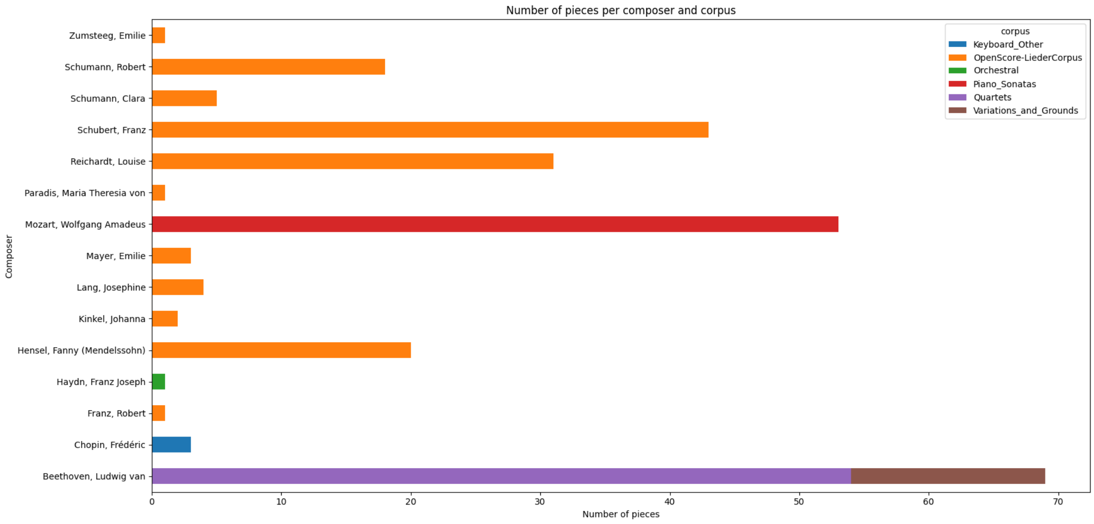

Figure 5.2: Distribution of composers and genre of the selected pieces in the When-in-Rome dataset

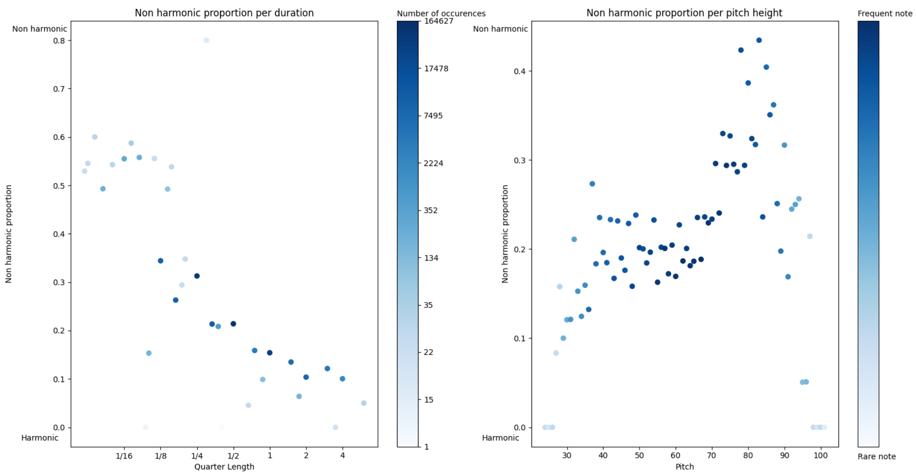

Figure 5.3: **Left**: The proportion of non-harmonic notes relative to note duration, with darker points indicating more frequent rhythms. **Right**: The proportion of non-harmonic notes relative to pitch height, with darker points representing more frequent pitches.

Moreover, some analyses do not align with their associated score, notably due to incomplete measures, which is a common occurrence in repeat bars with anacruses.

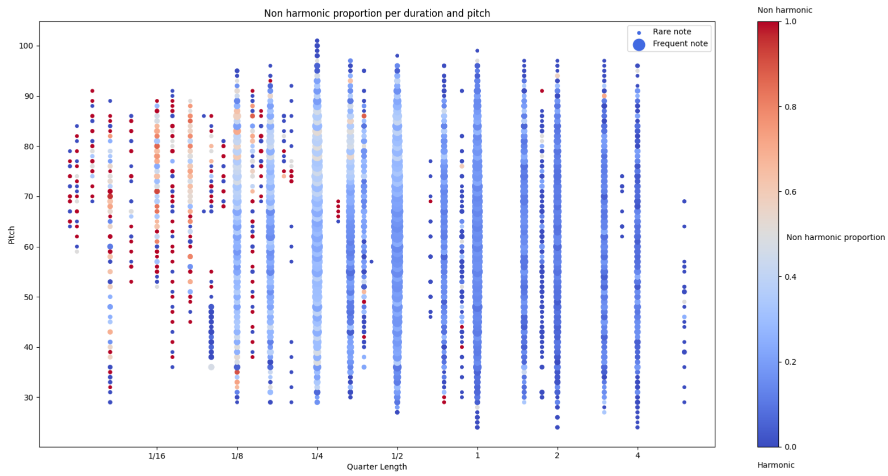

Figure 5.4: Points are positioned according to their pitch and duration. The size of each point indicates the frequency of occurrence. The color represents the proportion of non-harmonic notes: red indicates a high proportion of non-harmonic notes, and blue indicates a high proportion of harmonic notes.

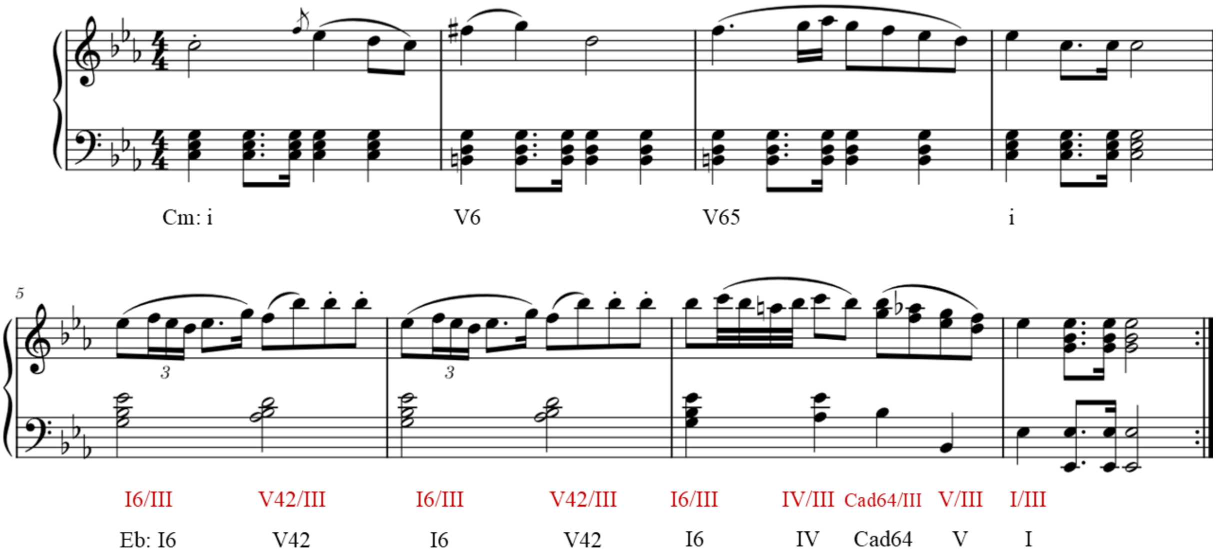

Figure 5.5: First measures of *9 Variations on a March by Dressler* by Ludwig van Beethoven. For the second half, the analysis given by the When-in-Rome dataset is above a standard analysis.

5.2 Comparison of algorithms on the Dataset

To address these limitations, effort was invested in selecting a subset of pieces and manually verifying and curating their analyses. The "curated" dataset, used for algorithm comparison, includes 34 pieces: 11 themes and variations by Ludwig van Beethoven, 3 piano pieces by Frédéric Chopin, and 20 lieder by Fanny Mendelssohn.

Table 5.1: Algorithm comparison between AugmentedNet (AugNet) and our algorithm (TonGraph)

Composer	Key accuracy		Degree accuracy		Degree and quality accuracy	
	AugNet	TonGraph	AugNet	TonGraph	AugNet	TonGraph
Beethoven	89.5	74.7	73.9	57.1	69.0	52.4
Chopin	81.9	61.6	64.6	52.1	58.6	47.0
Mendelssohn	84.6	69.5	69.8	57.7	63.5	52.9
Average	85.9	70.5	70.7	57.0	64.9	52.2

Our algorithm that we name TonalGraph can be found at https://github.com/Sebastien-li/Tonal-Graph. Its output is compared to the output of the state of the art algorithm *AugmentedNet* [8] by computing the relative duration of Roman numerals that are identical to the provided ground truth. More specifically, we verify the accuracy of the Roman numerals's key, its degree and its degree with quality. For example, a Roman numeral E: V has the same key and degree as E: V7, but differs in quality. This distinction means that the error can be categorized differently from a straightforward mistake. The accuracies are shown in table 5.1.

It appears that our algorithm is underperforming compared to AugmentedNet, with discrepancies of around fifteen percentage points in accuracy. It is worth noting that both AugmentedNet and our algorithm struggle more with pieces that have highly ambiguous harmony. The piece with the worst accuracies is Chopin's Revolutionary Étude (Op. 10, No. 12), which features complex harmony and a large number of non-harmonic notes, with a degree accuracy of 59.2% for AugmentedNet and 33.8% for our algorithm.

An interactive dashboard was also developed to compare and evaluate the strengths and limitations of both algorithms using toy examples. Let us consider an excerpt from Mozart's Sonata No. 16 (Figure 5.6) where both algorithms showed poor accuracy.

Measure 10 is globally a C:IV chord with a passing V65/V chord at the last eighth-note. Measure 11 could be analyzed as either in C major or G major, as it is a transition between the two keys. The ground truth analysis does not modulate yet, while AugmentedNet suggests a modulation to G major at the third beat of measure 11. Both analysis are justifiable. However, TonalGraph proposes a modulation to G major at the third beat of measure 10, which is be slighly premature; the earliest justifiable point for modulation would be the last eighth-note of measure 10.

AugmentedNet tends to adopt a more global perspective, treating the F♯ as a passing chord and not assigning a Roman numeral, whereas the ground truth analysis specifies the IV6 and the passing V65/V. This global approach results in an analysis of measure 11 that overlooks the C: I64 chords, which may be harder to justify (but not entirely unjustifiable). Conversely, TonalGraph appears to take a more local approach, attempting to assign a Roman numeral to each quarter note, and failing due to the abundance of non-harmonic notes. Nevertheless, TonalGraph's analysis of measure 11 is closer to the ground truth.

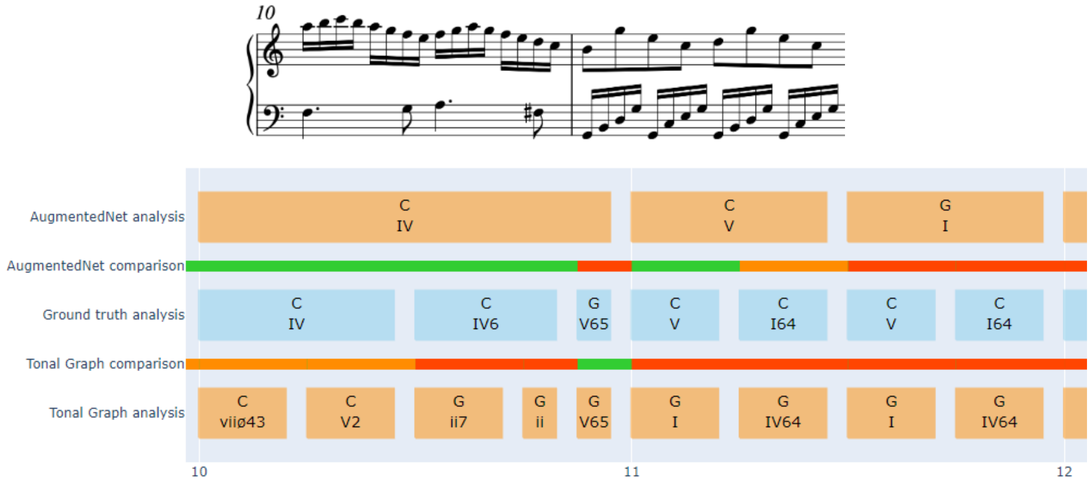

Figure 5.6: Measure 10 and 11 of *Sonata No. 16* by Wolfgang Amadeus Mozart. The analysis of the When-in-Rome dataset (middle), and the analysis created by the algorithms AugmentedNet (Top side) and TonalGraph (Bottom side) are shown in orange and blue boxes. The orange bar shows a difference in Roman numeral and the red bar shows a difference in key.

By checking the constructed rhythm tree with the chord with the highest score of each node (see Figure 5.7), it can be noted that the failure in analyzing measure stems from the harmonic rhythm selection algorithm (algorithm 1). A more convincing analysis could have been derived if it selected the chord of a higher subdivision of half-notes instead of quarter-notes.

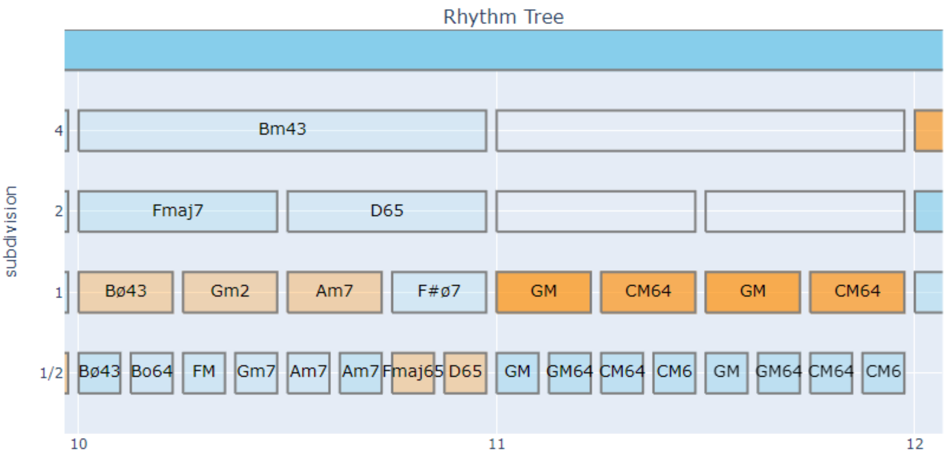

Figure 5.7: Rhythm tree with the chord of highest score for each node. Orange nodes indicates the selected nodes.

It is also important to note that TonalGraph can yield very satisfying results when applied to harmonically simpler pieces. In the initial measures of the theme exposition of Beethoven's Variations on "God Save the King" (see Figure 5.8), both AugmentedNet and TonalGraph exhibit very high accuracy. The 100% accuracy of AugmentedNet can be attributed to the fact that this piece is included in its training dataset.

In this example, all the errors arise from differing interpretations of a chord as a passing chord, with both interpretations being valid. In each case, the ground truth analysis is present in the rhythm tree at a higher subdivision. It can be noted that AugmentedNet's result labels the Am7 chord in measure 2 as a passing G chord, which is slightly incoherent. A more consistent analysis would be either "m2 V b2.5 vi7 b3 V6" or "m2 V," rather than "m2 V b2.5 V b3 V6."

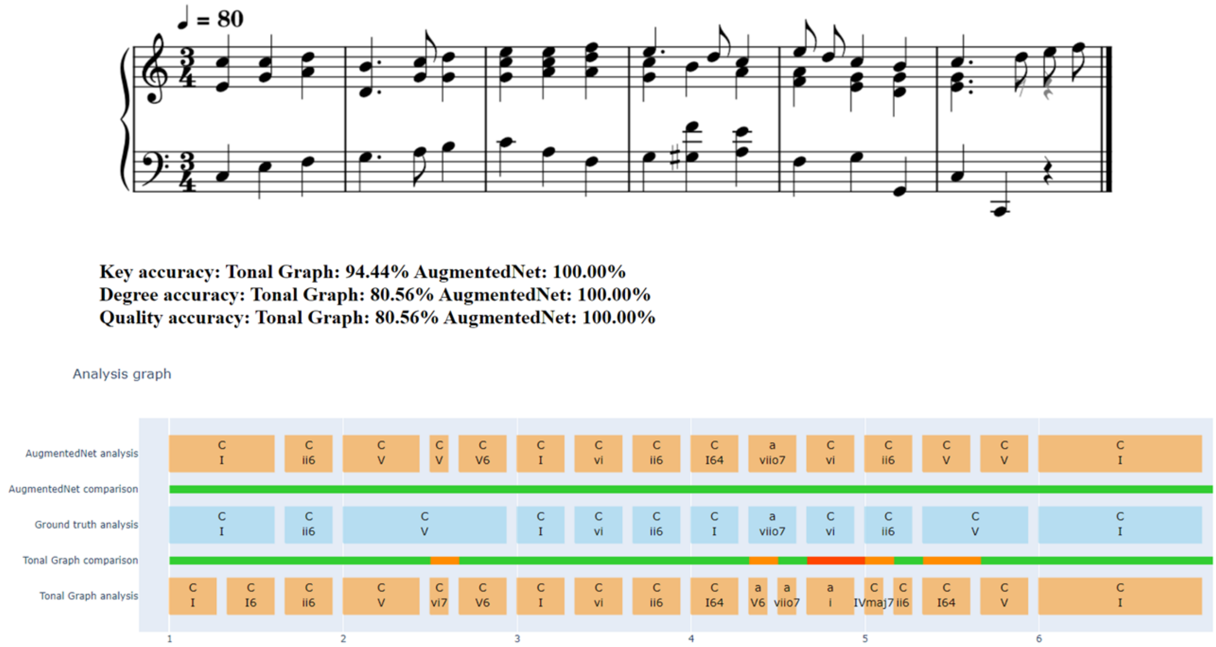

Figure 5.8: The analysis comparison of the first measures of *7 Variations on "God Save the King"* by Ludwig van Beethoven

6 Conclusion and Perspectives

The presented method appears to have strong potential, not only achieving satisfying analyses but also offering a level of interpretability and interactivity that could make it more appealing than state-of-the-art methods, especially in applications that requires not only the resulting analysis but also how it was obtained.

By exploring the algorithm's results and fine-tuning the highly interpretable parameters such as Roman numeral weights, transition weights, or the formula for harmonic segmentation in the rhythm tree, further refinement of the method could be achieved by musicians without complete comprehension of the algorithm.

Therefore, a natural extension of this work would be the development of a harmonic notation tool that uses interactive graphs for enhanced visualization.

Moreover, the graph theory approach could be expanded to broader types of musical analysis at higher hierarchical levels. By employing spatio-temporal graphs that capture more complex musical information, such as textural, structural, or hierarchical elements, and by applying tools like mathematical morphology on the graph lattice, we could explore the geometric relationships of similarities in music. This area of study will be the focus of my upcoming thesis.

Bibliography

[1] Tsung-Ping Chen and Li Su. “Functional Harmony Recognition of Symbolic Music Data with Multi-Task Recurrent Neural Networks.” In: *Proceedings of the 19th International Society for Music Information Retrieval Conference*. Paris, France, 2018, pp. 90–97.

[2] J. Devaney et al. “Theme And Variation Encodings with Roman Numerals (TAVERN): A New Data Set for Symbolic Music Analysis”. In: *Proceedings of the 16th International Society for Music Information Retrieval Conference*. Málaga, Spain, 2015, pp. 728–734. URL: https://www.semanticscholar.org/paper/Theme-And-Variation-Encodings-with-Roman-Numerals-A-Devaney-Arthur/574734ed2599a06e64150595b88cbf176a673945.

[3] Mark Gotham et al. “When in Rome: a meta-corpus of functional harmony”. In: *Transactions of the International Society for Music Information Retrieval* (2023).

[4] P. Illescas, D. Rizo, and J. Quereda. “Harmonic, Melodic, and Functional Automatic Analysis”. In: *Proceedings of the 2007 International Computer Music Conference*. 2007, pp. 165–168.

[5] Emmanouil Karystinaios and Gerhard Widmer. “Roman Numeral Analysis With Graph Neural Networks: Onset-Wise Predictions From Note-Wise Features”. In: Milan, Italy: ISMIR, 2023, pp. 597–604. DOI: 10.5281/zenodo.10265357.

[6] Pedro Kröger, Alexandre Passos, and Marcos Sampaio. “Rameau: A System for Automatic Harmonic Analysis”. In: *International Conference on Mathematics and Computing*. 2009, pp. 1–8.

[7] Néstor Nápoles López. “Automatic Harmonic Analysis of Classical String Quartets from Symbolic Score”. MA thesis. Universitat Pompeu Fabra, 2017.

[8] Néstor Nápoles López, Mark Gotham, and Ichiro Fujinaga. “AugmentedNet: A Roman Numeral Analysis Network with Synthetic Training Examples and Additional Tonal Tasks”. In: *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. 2021, pp. 404–411.

[9] H. John Maxwell. “An Expert System for Harmonic Analysis of Tonal Music”. In: *Understanding Music with AI*. AAAI Press, 1992, pp. 147–160.

[10] Matt McVicar et al. “Automatic Chord Estimation from Audio: A Review of the State of the Art”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.2 (Feb. 2014). Conference Name: IEEE/ACM Transactions on Audio, Speech, and Language Processing, pp. 556–575. ISSN: 2329-9304. DOI: 10.1109/TASLP.2013.2294580.

[11] Gianluca Micchi, Mark Gotham, and Mathieu Giraud. “Not All Roads Lead to Rome: Pitch Representation and Model Architecture for Automatic Harmonic Analysis”. In: *Transactions of the International Society for Music Information Retrieval* 3.1 (2020), pp. 42–54. ISSN: 2514-3298. DOI: 10.5334/tismir.45.

[12] Markus Neuwirth et al. "The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets". In: *Frontiers in Digital Humanities* 5 (2018). ISSN: 2297-2668. DOI: 10.3389/fdigh.2018.00016.

[13] Christopher Raphael and Joshua Stoddard. "Functional Harmonic Analysis Using Probabilistic Models". In: *Computer Music Journal* 28.3 (2004), pp. 45–52. ISSN: 0148-9267. JSTOR: 3681508.

[14] Gonzalo Romero-García. "Mathematical Morphology for the Analysis and Generation of Time-Frequency Representations of Music". 2023SORUS554. PhD thesis. 2023. URL: http://www.theses.fr/2023SORUS554.

[15] David Temperley. *The Cognition of Basic Musical Structures*. Cambridge, Mass.: MIT Press, 2004. ISBN: 978-0-262-70105-1.

[16] David Temperley. "A Unified Probabilistic Model for Polyphonic Music Analysis". In: *Journal of New Music Research* 38.1 (2009), pp. 3–18. ISSN: 0929-8215. DOI: 10.1080/09298210902928495.

[17] Dmitri Tymoczko et al. "The Romantext Format: A Flexible and Standard Method for Representing Roman Numeral Analyses". In: *Proceedings of the 20th International Society for Music Information Retrieval Conference*. Delft, The Netherlands, 2019, pp. 123–129.

[18] Terry Winograd. "Linguistics and the Computer Analysis of Tonal Harmony". In: *Journal of Music Theory* 12.1 (1968), pp. 2–49. ISSN: 0022-2909. DOI: 10.2307/842885. JSTOR: 842885.