Mathematical Morphology for the Analysis and Generation of Time-Frequency Representations of Music

Gonzalo Romero-García

Sorbonne Université

École doctorale : Informatique, Télécommunications et Électronique de Paris
Laboratoire : Sciences et Technologies de la Musique et du Son, UMR 9912

PhD Thesis in Computer Science

Mathematical Morphology for the Analysis and Generation of Time-Frequency Representations of Music

---

Presented by:
M. Gonzalo Romero-García

Supervised by:
Pr. Carlos Agón
Pr. Isabelle Bloch

The defense took place on the 15th of November 2023, before a jury composed of:

Julien Tierny	**DR** at CNRS	*President*
Florent Jacquemard	**CR** at Inria	*Reviewer*
Laurent Najman	**Pr.** at Université Gustave Eiffel	*Reviewer*
Thierry Géraud	**Pr.** at Epita	*Examiner*
Dmitri Tymoczko	**Pr.** at Princeton University	*Examiner*


Abstract

This thesis explores the application of Mathematical Morphology to the analysis and generation of music, focusing on two time-frequency representations: spectrograms and piano rolls. Mathematical Morphology is a nonlinear image processing tool that serves to consider topological notions of the image. We present three applications. The first is to analyze spectrograms with morphological tools to obtain parameters with which to synthesize a musical instrument sound. The second is to generate piano rolls with two musical parameters, texture and harmony, by arranging them through morphological dilation. The third is to apply morphological operators to analyze piano rolls using graph theory. The thesis thus proposes new approaches for problems in sound analysis and computational musicology.

Résumé

Cette thèse explore l'application de la Morphologie Mathématique à l'analyse et à la génération de musique, en se concentrant sur deux représentations temps-fréquence : les spectrogrammes et les piano roll. La Morphologie Mathématique est un outil de traitement d'images non linéaire qui sert à exploiter des notions topologiques de l'image. Nous présentons ici trois applications : la première consiste à analyser les spectrogrammes avec des outils morphologiques pour obtenir des paramètres avec lesquels synthétiser un son d'instrument musical. La seconde est de générer des piano roll avec deux paramètres musicaux, la texture et l'harmonie, en les agençant avec la dilatation morphologique. La troisième consiste à appliquer des opérateurs morphologiques pour analyser les rouleaux de piano en utilisant la théorie des graphes. Ainsi, la thèse propose de nouvelles approches pour les problèmes d'analyse sonore et de musicologie computationnelle.

Acknowledgements

The completion of this PhD represents the culmination of years of dedication and study. Although the journey of writing a PhD is often a lonely one, it would not have been possible without the support of numerous individuals who stood by my side. Here is the list of those I wish to thank for their unwavering support.

But before that, I want to thank the institutions that made possible me becoming doctor: Sorbonne Université, which I consider my *alma mater*, and IRCAM, which is the place I want to be. It is marvelous that these two institutions work together and that I was able to be part of them through the ATIAM master and the EDITE doctoral school.

I would like to thank now all the people that helped me along the way.

First of all, I want to thank my two supervisors, Carlos Agón and Isabelle Bloch for believing in me and supporting me during all this journey. I could not have dreamed with better supervisors; together, they form the necessary Yin and Yang.

Secondly, I want to thank Moreno Andreatta and Gérard Assayag for having introduced me into IRCAM's RepMus team. It has been a real pleasure to be part of such an amazing team and to finally found people that explore the very specific domain I dreamed all my life: computational musicology.

This PhD lies at the intersection of three domains: mathematics, computation and music. In the process of learning these topics, I was helped by several professors that marked me. I would like to thank all of them.

In particular, I want to thank my teachers from the Real Conservatorio Superior de Música de Madrid; special thanks to Alicia Díaz de la Fuente, who taught me analysis, Alejandro Román, who encouraged me to visualize music, Zulema de la Cruz, who introduced me to sound synthesis, and Enrique Rueda, who explained to me what harmony means.

Also, I want to thank those from Universidad Complutense de Madrid, in particular Daniel Azagra Rueda, the supervisor of my first research work, and the one who encouraged me to come to Paris and to integrate Sorbonne Université.

And finally, the teachers of ATIAM's Master, from whom I learned computing and signal processing. In particular, I want to thank Roland Badeau, the one who taught me all I know about Signal Processing, and Philippe Esling, the one who taught me all I know about Machine Learning.

And now is the time to thank those that did contribute to my PhD in non-academic ways. I prefer to express my gratitude to them in the language I speak with them, so I will use either Spanish of French.

Me gustaría empezar por mi familia. Esas personas que me apoyaron indefectible-

mente y tendieron la oreja cada vez que quise contarles mis extravagancias matemusicales. Me siento muy afortunado de tenerlos a mi lado y de todo lo que han hecho por mí. Hablo, por supuesto, de mis tías (Ofelia y Laura), de mis tíos (Vicente, Quino y Ramón), de mis abuelos (Enrique y Maribel), de mis primos, ... Una mención especial merecen mis hermanos: Ana, la mayor, que con el paso del tiempo ha pasado de madre a hermana, y que ha traído al mundo a mis dos sobrinas preciosas, Laura y Carmen. Enrique, el pequeño, la persona en el mundo que mejor me conoce, aquél con el que he pasado infancia, adolescencia y juventud, mi *nakama*, mi amigo, mi hermano. Y para terminar, me gustaría agracer a mis padres: Anabel y Jesús. Nada de esto hubiera sido posible sin ellos. Su apoyo, tanto financiero como moral, ha sido el porqué y el cómo de que yo sea hoy quien soy. Siempre confiaron en mí y nunca dudaron de que fuera a conseguirlo. Gracias Papá. Gracias Mamá.

Hay otra familia que también es mi familia: los Hevia-Aranguren. Esas personas joviales y cultivadas que hacen que cualquier momento sea disfrute. Sea poesía o sea ciencia, son ellos los que, con más sapiencia, entonan la melodía. Redi Padre y María, hiceron un buen día, y con mucha diligencia, dos retoños destinados a llegar a mi presencia. Clara, nuestra fiel compañera, siempre alegre y cariñosa, divertida y deleitosa. Redi: la roca, impasible, calculador. Un amigo que conservo desde mi más tierna infancia y que siempre ha creído en mí y me ha animado y apoyado para que haga realidad mi sueño. Muchas gracias a todos.

Pero, como dice el refrán, *no hay dos sin tres*. Mes remerciements se dirigent cette fois-ci à une troisième famille, celle qui m'a accueilli en France comme si j'étais un des leurs : la famille Denizeau. Gérard, un puits de science, Cécile, un puits d'amour, Aurélien, la parfait mélange entre intelligence et humour, Sylvestre, le parfait mélange entre intelligence et sarcasme. Grâce à eux je me sens en France comme si j'étais à la maison.

Continuons sur cette belle lancée pour remercier les amis que j'ai connus à Paris. Les premiers sont ceux avec lesquels je fus amené à habiter dans la résidence JVD[1] à la Cité internationale universitaire de Paris, cet endroit merveilleux où je passai mes trois années de thèse. Ce fut un moment extraordinaire, d'échange et de multiculturalité. Quelques personnages notables sont Sergi, Carlos, Miguel ou Jesus, les espagnols à Paris grâce auxquels je n'oubliais pas mes origines. Je voudrais aussi remercier le directeur de la résidence, Bertrand Cosson, qui fut mon salvateur quand je croyais tout perdu, en m'octroyant trois ans de plus dans ce magnifique campus.

Après JVD ce fut l'ATIAM. Je garde un merveilleux souvenir de ces moments passés à partager notre envie de savoir. De très belles amitiés en sont nées ; André, Colette, Victor, Lenny, Lydia, ... Avec un en particulier, Olivier Birot, je continue à

[1] Julie-Victorie Daubié.

partager des soirées entières à parler informatique et musique.

Et puis le temps passe et tu rencontres des gens qui rentrent dans ta vie avec une puissance particulière : Daniel Bedoya, camarade de COSMOS, Thomas Borsoni, camarade de maths, Christophe Weis, camarade d'analyse musicale, Pierre Ludmann, camarade d'informatique, Timothée Chambery, camarade de discussions, Mohsen Mechichi, camarade de jeux. Je suis très heureux de vous avoir connus.

Et, avant de finir avec le français, je voudrais remercier les nouveux amis que je me suis fait ici pour l'accueil chaleureux qu'ils m'ont réservé. Merci de m'avoir intégré dans votre clan. De plus, je ne voudrais pas oublier Victoria, avec qui j'ai passé une très bonne semaine à l'IRCAM et qui m'a invité à son séminaire à Strasbourg, ni Paul Lascabettes, avec qui j'ai partagé cette aventure de Morphologie Mathématique Musicale.

Y ya va siendo hora de volver a las raíces: Madrid y Valencia. Y a aquellos que llevan conmigo desde la *uni* y el *liceo*. Empecemos por los de Madrid.

En aquellos tiempos, acababa de llegar a Madrid y estaba ansioso de saber qué me depararía la vida. Y fue en ese momento que conocí a personas que llevo y llevaré siempre en mi corazón. En el orden cronológico, el primero fue Jorge, que me recibió con un « hola, me llamo Jorge », seguido de Alberto, uno de esos amigos que siempre te alegras de ver y con el que sientes que la conversación podría durar para siempre; pocos hay en el mundo tan listos y tan cultos. Un gran hallazgo fue Adrián Pineda; desde ese día en el ascensor y esa conversación en los cien montaditos, me abrió las puertas de sus círculos de par en par. Tardes enteras de conversaciones de música y noches enteras de juerga madrileña. Me gustaría también mencionar a los que llegaron al final: Xabi, con el que terminé la carrera (aún recuerdo ese trabajo de topología algebraica), y Darío, aquel que ha pasado de alumno a maestro.

Y para terminar de enrraizar, volvamos a Valencia, esa tierra tórrida de la que vengo, lo quiera o no. Allí pasé mi infancia y adolescencia, y guardo de esos tiempos muchos amigos. Pero de todos ellos, me gustaría resaltar cuatro: Jorge, Pedro, Louis y Cristóbal. Cada uno diferente al otro. Cada cual en lo suyo. Y, a día de hoy, cada uno en un sitio. Del último, me gustaría resaltar hasta qué punto ha sido central en mi desarrollo; tanto hemos hablado que podría decirse que hemos formado una mente común. Gran visionario, mejor persona. Pero no quisiera olvidarme de dos personas de Valencia que han contribuido *directamente* a esta tesis: Marina Delicado, con la que grabé los *samples* de piano que utilicé al principio de mi tesis, y Carlos García Pagán, que ha diseñado la figura 2.8d.

Hay una persona que he dejado deliberadamente para el final: Bérengère Denizeau. No voy a mencionar los diferentes apelativos que utilizo para dirigirme a ella, pues podrían llenar una tesis entera. Ma compagne d'aventures, mi media

naranja, aquella que hace que cada día sea un regalo. La personne qui m'a soutenu du début jusqu'à la fin, en passant par le milieu, et encore après. Siempre ahí; intelligente, subtile, mais surtout drôle. Sagace et stratégique, douce et ferme, avec un œil d'aigle et un cœur de lion. Es músico y escritora, baila y canta, et jongle avec le langage haciendo trucos de ilusionista. Heureusement que je t'ai connu. C'est à toi que je dédie cette thèse.

Notations

Set theory

1. $\mathcal{P}(X)$: the power set of X.
2. B^A: the set of functions from A to B, i.e., $B^A = \{f \in \mathcal{P}(A \times B) : a = b \Rightarrow f(a) = f(b)\}$
3. $|X|$: if X is a set, the cardinal of X, i.e., the number of elements of X. If the number of elements of X is infinite, we can notate $|X| = \infty$ and, if needed, use the usual $\aleph_0, \aleph_1, ...$ for different cardinals.
4. $\mathcal{P}_F(X)$: the finite subsets of X. We can write $\mathcal{P}_F(X) = \{A \in \mathcal{P}(X) : \exists n \in \mathbb{N} \text{ such that } |A| = n\}$.

Arithmetic

We will use several common number sets. We will use the dot symbol for decimals since the comma symbol is already used to enumerate elements.

1. $\mathbb{N}$: the set of natural numbers with the zero included[2]. We will call them the *natural numbers*.
2. $\mathbb{N}^*$: the set of natural numbers without the zero. We will call them the *positive natural numbers*.
3. $\mathbb{Z}$: the set of integer numbers.
4. $\mathbb{Q}$: the set of rational numbers.

[2]We follow ISO standards (see (ISO, 2019b))

5. $\mathbb{Q}^{+*}$: the set of positive rational numbers.
6. $\mathbb{Q}^{+}$: the set of non-negative rational numbers.
7. $\mathbb{R}$: the set of real numbers.
8. $\mathbb{R}^{*}$: the set of real numbers without the zero.
9. $\mathbb{R}^{+*}$: the set of positive real numbers.
10. $\mathbb{R}^{+}$: the set of non-negative real numbers.
11. $\imath$: the imaginary unit, i.e., the number such that $\imath^2 = -1$
12. $\mathbb{C}$: the set of complex numbers.
13. $\mathbb{T}$: the one dimensional torus, i.e., $\mathbb{R}/\mathbb{Z}$.
14. $\mathbb{K}$: a field.
15. $\overline{z}$: the complex conjugate of $z \in \mathbb{C}$, i.e., if $z = a + b\imath$ then $\overline{z} = a - b\imath$
16. $\mathbb{Z}_n$: the set of integer numbers modulo n.
17. $\overline{\mathbb{N}}$: an abbreviation for $\mathbb{N} \cup \{\infty\}$.
18. $\overline{\mathbb{Z}}$: an abbreviation for $\mathbb{Z} \cup \{-\infty, \infty\}$.
19. $\overline{\mathbb{Q}}$: an abbreviation for $\mathbb{Q} \cup \{-\infty, \infty\}$.
20. $\overline{\mathbb{R}}$: an abbreviation for $\mathbb{R} \cup \{-\infty, \infty\}$.

Functions

1. $\mathbb{1}_A \in \{0,1\}^X$: the characteristic function of subset A of X, i.e.,

$$\begin{array}{rccl} \mathbb{1}_A : & X & \to & \{0,1\} \\ & x & \mapsto & \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \end{array} .$$

2. We will use often the canonical bijection between the subsets of a set X and its characteristic function, i.e.,

$$\begin{array}{rccc} \mathbb{1}: & \mathcal{P}(X) & \to & \{0,1\}^X \\ & A & \mapsto & \mathbb{1}_A \end{array} .$$

3. $\text{supp}(f)$: the support of the function f, i.e., if $f : X \to \mathbb{K}$, $\text{supp}(f) = \{x \in X : f(x) \neq 0\}$, where $\mathbb{K}$ is a field.

Functional spaces

1. $\mathcal{L}_1(\mathbb{R};\mathbb{C})$: the space of functions from $\mathbb{R}$ to $\mathbb{C}$ that are integrable i.e., such that $||f||_1 = \int_{\mathbb{R}} |f(t)|\, dt < \infty$.
2. $\mathcal{L}_2(\mathbb{R};\mathbb{C})$: the space of functions from $\mathbb{R}$ to $\mathbb{C}$ that are square-integrable i.e., such that $||f||_2 = \int_{\mathbb{R}} |f(t)|^2\, dt < \infty$.
3. $\mathcal{L}_\infty(\mathbb{R};\mathbb{C})$: the space of functions from $\mathbb{R}$ to $\mathbb{C}$ that are bounded i.e., such that $||f||_\infty = \sup\{|f(t)| : t \in \mathbb{R}\} < \infty$.
4. $\mathcal{C}^\infty(\mathbb{R};\mathbb{R})$: The space of smooth functions from $\mathbb{R}$ to $\mathbb{R}$, i.e., the functions $f : \mathbb{R} \to \mathbb{R}$ such that $f^{(k)}$ exists and is continuous $\forall k \in \mathbb{N}$.

Order theory

1. $\vee$: the supremum symbol as a binary operation (for instance, $a \vee b$).
2. $\wedge$: the infimum symbol as a binary operation (for instance, $a \wedge b$).
3. $\bigvee$: the supremum symbol as a unary operation (for instance, $\bigvee A$).
4. $\bigwedge$: the infimum symbol as a unary operation (for instance, $\bigwedge A$).

Spaces for music

These spaces are presented in Chapter 2.

1. $\mathcal{T}$: a time space.

2. $\mathcal{T}_{\mathrm{s}}$: the time space measured in seconds.
3. $\mathcal{T}_1$: the time space measured in samples. It is equal to $\mathbb{Z}$.
4. $\mathcal{T}_{\mathbf{o}}^{\frac{p}{q}}$: the time space measured in wholes with a $\frac{p}{q}$ time signature.
5. $\mathcal{F}$: a frequency space.
6. $\mathcal{F}_{\mathrm{Hz}}$: the frequency space measured in Hertz.
7. $\mathcal{N}$: the space of pitches.
8. $\mathcal{N}_{12}$: the space of chromas.
9. $\mathcal{F}_{\mathrm{st}}$: the frequency space measured in semitones.
10. $\mathcal{A}$: an amplitude range.
11. $\mathcal{A}_2$: the Boolean lattice, i.e., $\{0, 1\}$.
12. $\mathcal{A}_3$: the rhythmic lattice, i.e., $\{\perp, \cdot, \times\}$.
13. $\mathcal{D}_{\boldsymbol{pf}}$: the score dynamics.
14. $\mathcal{D}_{128}$: the MIDI dynamics.
15. $\mathcal{A}_{\mathcal{D}}^{P}$: the pianistic dynamics.
16. $\mathcal{A}_{\mathcal{D}}$: the sustained dynamics.
17. $\mathcal{T} \times \mathcal{F}$: a time-frequency plane.
18. $\mathcal{A}^{\mathcal{T}\times\mathcal{F}}$: a musical space. Also notated as $\mathcal{M}$.

Contents

Introduction

Music encompasses a variety of activities, among which two stand out: the generation of music and the analysis of music. These two activities are the focus of this thesis. What sets this work apart lies in our approach to the generation and analysis of music through a specific mathematical and computational discipline: Mathematical Morphology.

Mathematical Morphology (MM) is a domain that lies at the intersection of Mathematics and Computer Science. It was originally formulated in the mid-1960s by Matheron, 1967, 1965, 1975 and J. Serra, 1982. Historically, this was the first consistent nonlinear image analysis theory (Najman & Talbot, 2010), and it remains an indispensable tool in today's field of image analysis.

Although originally developed for image analysis, particularly in the context of porous media, mathematical morphology has transcended its initial scope. This evolution into a broader discipline is highlighted by the development of a mathematical formalism based on lattice theory.

The main challenge of this thesis is to explore the applications of MM to music. Our chosen methodology involves examining the different musical representations to which MM can be applied, and determining their musical relevance. To achieve this, we focus on a specific aspect of MM that employs structuring elements. This requires a space of the form

$$(T^E, \preceq)$$

where $(T, \leq)$ is a complete lattice and E is a set endowed with a notion of neighborhood, that is, for each point $p \in E$, there is a neighborhood $\mathcal{V}(p)$ associated. The order $\preceq$ is the pointwise order induced by $\leq$, i.e., $\forall f, g \in T^E$, $f \preceq g \Leftrightarrow \forall p \in E$, $f(p) \leq g(p)$.

Multiple methods exist for associating a neighborhood with a point, but in MM, a prevalent approach is to use what is called a structuring element B. The neighborhood of p is then defined as the translation $p + B$. One way to accomplish this is by having a group $(G, +)$ acting on E, and consider B as a subset of G.

Group theory has proven extremely useful in musical analysis and generation (Lewin, 1987; Andreatta, 2004; Papadopoulos, 2015). Given this, we posit that MM can be meaningfully applied to music from a group-theoretical perspective. More precisely, we propose a particular space that we call a *musical space* and that is of the form

$$\mathcal{M} = \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$$

where $\mathcal{T}$ represents time, $\mathcal{F}$ represents frequency, together constituting the time-frequency plane $\mathcal{T} \times \mathcal{F}$, and $\mathcal{A}$ represents the set of possible amplitudes. The time-frequency plane, $\mathcal{T} \times \mathcal{F}$, is accompanied by a group $(G_{\mathcal{T}\times\mathcal{F}}, +)$ that acts on it. Additionally, the amplitude set is endowed with a complete lattice structure $(\mathcal{A}, \leq)$.

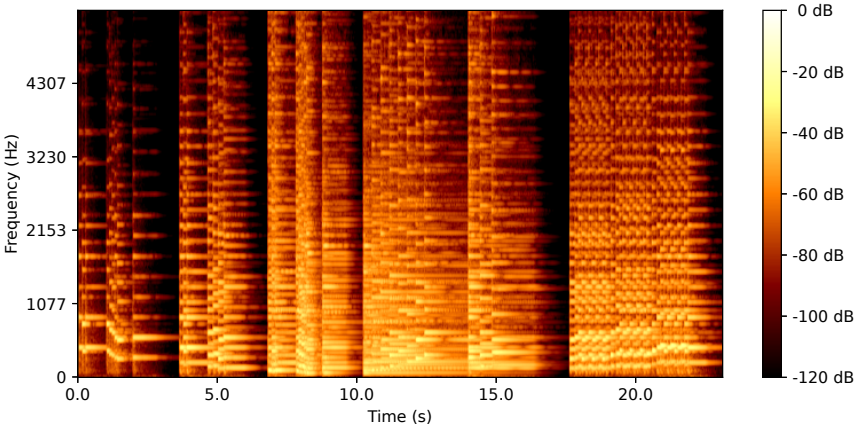
These musical spaces capture what are commonly referred to as time-frequency representations of music. Traditionally, musical representations are categorized into two major families: signal-based and symbolic. Within both of these families, multiple time-frequency representations exist. Our objective is to establish a unified framework within which MM can be applied. Due to its algebraic nature, we believe that MM can be adapted to accommodate both families of representations, despite their seemingly distinct natures.

Among the various time-frequency representations available, we have selected one from each major family: spectrograms for the signal-based family and piano rolls for the symbolic family. These chosen representations are well-studied and provide a robust foundation for the application of MM. Figure 1 showcases an example of each representation, featuring the initial five bars of Bach's Toccata and Fugue in D minor, BWV 565.

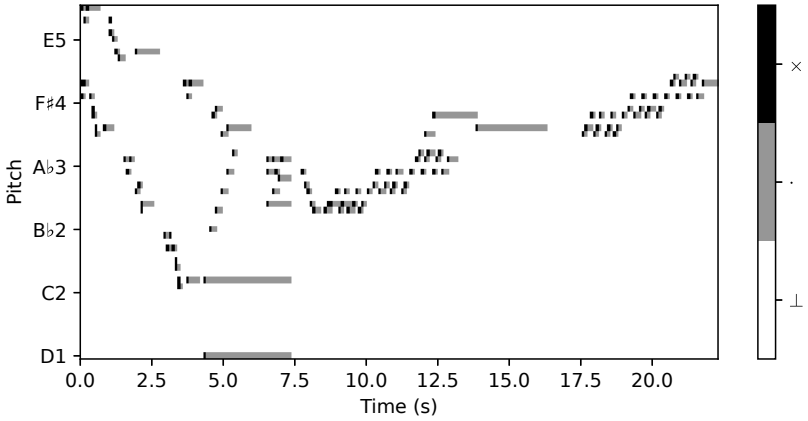
Several questions naturally arise within this context: what types of lattices should we employ for spectrograms and piano rolls? Which groups are most suitable? What is the musical significance of the structuring elements? Addressing these questions constitutes one of the primary focuses of this thesis. Throughout this work, we aim to answer these questions while ensuring musical relevance.

Although the primary goal of this thesis is to offer tools for both generating and analyzing music, considerable effort is devoted to establishing a rigorous mathematical framework. To this end, this thesis is replete with definitions, propositions, examples, and other standard mathematical terminology. Aside this mathematical rigor, every theoretical contribution has a computational counterpart. We have made the code available in a public repository at https://github.com/Manza12/MMM where one can find the figures, algorithms, and sounds presented throughout this thesis, together with the code that was used to generate them.

This thesis is organized as follows: Chapter 1 is devoted to providing a comprehensive introduction to mathematical morphology. In this chapter, we present



(a) Spectrogram



(b) Piano roll

Figure 1: Time-frequency representations of Bach's Toccata and Fugue in D minor, BWV 565 bars 1-5.

the key operators that will be used throughout the thesis and define the specific mathematical constructs essential for applying MM.

For example, we introduce a new formalism for greyscale MM, known as the residuated triplet, which enables a clear understanding of the roles of each space involved in the process, namely, the input, the structuring element, and the output. This distinction between spaces and precision in the use of operations is crucial in our framework. Unlike in image applications of MM, where the input and the output

are often of the same nature (i.e., an image), in our case this assumption does not hold true and lacks musical significance.

In Chapter 2, we explore the diverse options for time, frequency, and amplitude to construct spaces suitable for applying MM. We examine a range of choices, both discrete and continuous, as well as algebraic and analytical. We then formalize the structures that characterize our main representations: spectrograms and piano rolls.

We place particular emphasis on distinguishing between a *space* (a set of points) and a *group* acting on this space (a set of transformations, which in our case are conceived as shifts). This distinction enables us to assign a musical meaning to these actions, specifically the concept of translating a timestamp or a pitch by a certain amount (e.g., shifting A up by 3 semitones results in C).

Moreover, we also address amplitudes in a musically meaningful way. We explore various possible amplitude ranges and associate them with corresponding musical phenomena. This effort led to the concept of a residuated triplet; in traditional image processing, the amplitude range is often limited to grayscale values, but in the case of piano rolls, this may not be the case.

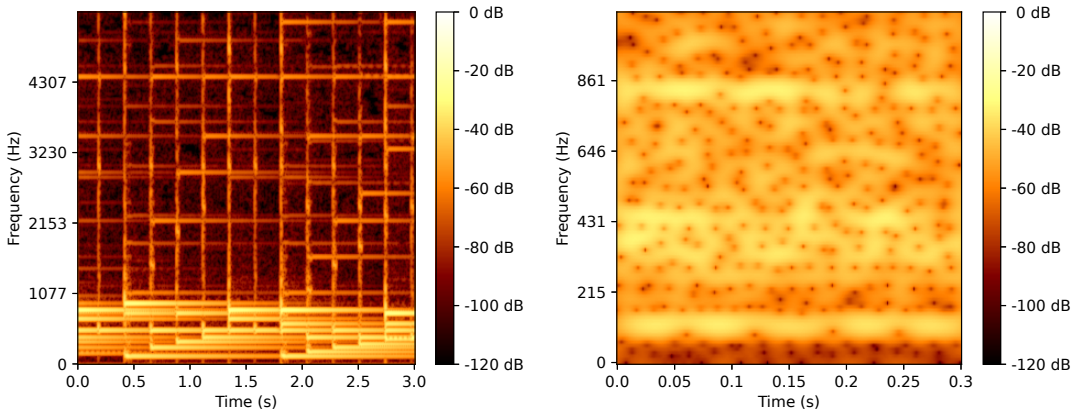
Finally, this chapter introduces a formal definition of a piano roll as well as a derived representation known as the chroma roll. While these concepts are standard in computational musicology, formal mathematical definitions have been lacking. We aim to provide one that is both flexible and rigorous.

After establishing the fundamental theoretical framework in Chapters 1 and 2, we turn to our primary contributions in the subsequent chapters. The remaining three chapters delve into the potential applications of MM for addressing specific Music Information Retrieval (MIR) tasks.

Chapter 3 explores the use of MM for analyzing spectrograms of sounds produced by musical instruments. Such sounds possess specific characteristics (an attack, a sinusoidal component, and a noise element) that make them well-suited for MM analysis. Specifically, the attack usually manifests as a vertical line in the spectrogram, while the sinusoidal component appears as a series of horizontal lines (see Figure 2a). In this context, we lean towards the conventional usage of MM, as it is typically applied in image analysis. Rather than endowing our structuring elements with musical significance, we design them to alter the image in a manner conducive to isolating specific details (the vertical and horizontal lines).

The noise component exhibits distinct characteristics compared to the other elements; it manifests as a density of energy punctuated by holes (see Figure 2b). In this case, traditional MM techniques prove to be particularly effective, enabling us to isolate this noise component by filtering out the lines, and thus the other components.

These methods are integrated into a unified morphological processing pipeline.



(a) Glockenspiel (b) Gong

Figure 2: Spectrograms of music instruments.

The processed information is subsequently utilized to synthesize a sound that closely resembles the original input.

Chapters 4 and 5 shift our focus to the use of MM for both generating (Chapter 4) and analyzing (Chapter 5) piano rolls. In these chapters, structuring elements change from being mere parameters to becoming central actors. We endow them with musical significance and employ them for both the generation and analysis of pieces represented as piano rolls.

One of the key contributions emerges in Chapter 4: the formalization of music generation through two musical (yet mathematical) parameters: texture and harmony. While these terms can be contextually ambiguous, we give them precise mathematical definitions within our framework. This chapter draws upon tools from other algebraic disciplines, such as the tensor product, and utilizes the full arsenal of techniques previously exposed, including residuated triplets.

To organize textures and harmonies coherently, we introduce an implementation model based on Python objects and XML documents. This model is intended for integration into Computer Assisted Composition (CAC) software, offering an alternative to traditional score editors for generating music that aligns with classical harmonic procedures.

The final chapter, Chapter 5, is dedicated to applying both MM and graph theory to address complex tasks in Music Information Retrieval (MIR). Specifically, we tackle two major challenges: chord segmentation and harmonic analysis. MM proves to be exceptionally well suited for these tasks, particularly for Roman numeral anal-

ysis. In this context, the operation of erosion takes on significant meaning and serves as the foundation for constructing what we term a *tonal graph*.

The *tonal graph* represents another major contribution of this thesis. We believe it addresses the challenges of harmonic analysis in a notably elegant manner, offering high levels of customization while still delivering strong results with minimal configuration. Although this concept is introduced for the first time in this work, ongoing research is being conducted and is expected to be published in subsequent studies.

While the core of our work is firmly anchored in Mathematical Morphology, our scope broadens to include a diverse array of fields within Mathematics and Computer Science, such as Group Theory, Abstract Algebra, Fourier Analysis, and Graph Theory. Central to our methodology is the role of MM tools as key elements in a holistic problem-solving pipeline. This approach is vividly demonstrated in Chapters 4 and 5, where we employ algebraic methods and graph-based strategies to address complex challenges, ranging from the generation of musical compositions to the analysis of harmony and texture.

Chapter 1

Mathematical Morphology

Mathematical Morphology (MM) is a theory and technique used for the analysis and processing of geometrical structures, initially developed for image analysis in the mid-1960s by two researches at the *École des Mines* in Paris: Matheron, 1975 and J. Serra, 1982. Its mathematical foundation draws upon set theory, lattice theory, topology, and random functions. While MM is predominantly applied to digital images, its versatility allows for its use on various other mathematical structures, including graphs, surface meshes, or solids.

In this chapter, we present the standard framework of deterministic Mathematical Morphology (MM) based on lattice theory. Specifically, we develop MM based on structuring elements with our input being sets (leading to binary MM) and functions (leading to greyscale MM).

As a roadmap, we are guided by the works of Najman and Talbot, 2010 and Bloch et al., 2007, adopting their notations with few exceptions. Additionally, we aim to enhance the mathematical formalism of the framework in which we will operate throughout the thesis. To achieve this, we will use the notion of a group acting on a set[1] for the application of MM with structuring elements. Moreover, we will incorporate the concept of a residuated lattice triplet for the codomains of functions, ensuring that the operations performed on them are well defined.

The organization of the chapter is as follows: we begin with the algebraic foundations of MM in Section 1.1, where we present the standard operators in the framework of complete lattices. Next, in Section 1.2, we delve into the common framework of MM with structuring elements, introducing the necessity of a group acting on a set and a residuated lattice triplet. Finally, in Section 1.3, we discuss some essential considerations for implementing the operators. In particular, we will present our

[1]This notion already underlies in previous works, but we will make it explicit and central.

contributions in the Python/PyTorch framework, achieved through the creation of the MM library `nnMorpho`.

1.1 Mathematical Morphology on Lattices

The commonly accepted framework for MM in the deterministic setting is the theory of complete lattices (Ronse, 1990). In this section, we provide a review of the definitions of complete lattices and basic morphological operators. For proofs of propositions and more in-depth development about this topic, we refer to (Heijmans & Ronse, 1990) and (Ronse & Heijmans, 1991).

1.1.1 Complete Lattices

The definition of a lattice is based on the concept of *partially ordered set* or, abbreviated, *poset*; this algebraic structure is extensively presented in (Birkhoff, 1948) and its definition is recalled in Appendix A (Definition A.2).

In order to define a lattice, we use the notions of *supremum* and *infimum*. These notions are defined formally in (Birkhoff, 1948) and their definitions are recalled in Appendix A (Definitions A.3 and A.4). We recall that the supremum of a set is the lowest upper bound of the set and the infimum of a set is the greatest lower bound of the set. We use the notations exposed in the preamble. These elements may not exist in general lattices.

Definition 1.1 (Lattice)**.** Let $(L, \leq)$ be a partially ordered set, and $\vee$ and $\wedge$ the supremum and infimum associated with $\leq$.

1. If $\forall a, b \in L,\ \exists c \in L :\ c = a \vee b$ then $(L, \leq)$ is an **upper semilattice**.
2. If $\forall a, b \in L,\ \exists c \in L :\ c = a \wedge b$ then $(L, \leq)$ is a **lower semilattice**.
3. If $(L, \leq)$ is both an upper semilattice and a lower semilattice then $(L, \leq)$ is a **lattice**.

We then use the notation $(L, \leq, \vee, \wedge)$ for lattices. Let us present the two main examples of lattices that we are using in this thesis.

Examples 1.2.

1. *Let E be a set. Then the set of subsets of E equipped with the set inclusion, union and intersection, $(\mathcal{P}(E), \subseteq, \cup, \cap)$ is a lattice.*

2. *Let E be a set and $(T, \leq)$ be a lattice. We define the pointwise order $\preceq$ by*

$$\forall f, g \in T^E,\ f \preceq g \Leftrightarrow \forall p \in E,\ f(p) \leq g(p)\,.$$

Then, $(T^E, \preceq)$ is a lattice. In this case, the supremum $\vee$ and the infimum $\wedge$ are given by:

$$\begin{array}{rccl} \vee: & T^E \times T^E & \to & T^E \\ & (f,g) & \mapsto & f \vee g: \begin{array}{ccc} E & \to & T \\ p & \mapsto & f(p) \vee g(p) \end{array} \\ \wedge: & T^E \times T^E & \to & T^E \\ & (f,g) & \mapsto & f \wedge g: \begin{array}{ccc} E & \to & T \\ p & \mapsto & f(p) \wedge g(p) \end{array} \end{array}\,.$$

These two examples are the ones that we will work with throughout the thesis; the first one is called the *binary* case and the second one the *functional* case.

When working with lattices, one can take the supremum and the infimum of any finite subset by considering each element one by one[2]. However, we cannot guarantee that the supremum of an infinite subset of the lattice exists. Lattices that satisfy this property are called *complete*. They are presented now and will be utilized throughout the thesis.

Definition 1.3 (Complete lattice)**.** Let $(L, \leq, \vee, \wedge)$ be a lattice. We say that $(L, \leq, \vee, \wedge)$ is a **complete lattice** if $\forall A \subseteq L$, $\exists \bigvee A \in L$ and $\exists \bigwedge A \in L$.

In the case of complete lattices, supremum and infimum can also be considered as unary operators defined on the power set of the lattice:

$$\begin{array}{rccl} \bigvee: & \mathcal{P}(L) & \to & L \\ & A & \mapsto & \bigvee A \end{array} \quad \text{and} \quad \begin{array}{rccl} \bigwedge: & \mathcal{P}(L) & \to & L \\ & A & \mapsto & \bigwedge A \end{array}\,.$$

We substitute then $\vee$ and $\wedge$ by $\bigvee$ and $\bigwedge$ in the notation.

A direct consequence of $(L, \leq, \bigvee, \bigwedge)$ being a complete lattice is that there exist two particular elements:

- the *top* element, denoted by $\top$, which is the supremum of L, i.e., $\top = \bigvee L$,
- the *bottom* element, denoted by $\bot$, which is the infimum of L, i.e., $\bot = \bigwedge L$.

We then use the notation $(L, \leq, \bigvee, \bigwedge, \top, \bot)$ for complete lattices.

[2]It is important to note that supremum and infimum are associative and commutative.

Remark 1.4. Let $(L, \leq, \bigvee, \bigwedge, \top, \bot)$ be a complete lattice. Then,

$$\top = \bigwedge \emptyset \qquad\qquad \bot = \bigvee \emptyset \,.$$

Let us discuss the Examples 1.2 from the perspective of complete lattices.

Examples 1.5.

1. *Let E be a set. Then, the lattice $(\mathcal{P}(E), \subseteq)$ is complete. The top element is E and the bottom element is $\emptyset$.*

2. *Let E be a set and $(T, \leq, \bigvee, \bigwedge, \top, \bot)$ be a complete lattice. Then, $(T^E, \preceq, \bigvee, \bigwedge, \top, \bot)$ is a complete lattice. The unary operators are then written*

$$\begin{array}{rccl} \bigvee: & \mathcal{P}(T^E) & \to & T^E \\ & F & \mapsto & \bigvee F: \begin{array}{rcl} E & \to & T \\ p & \mapsto & \bigvee\{f(p) \in T : f \in F\} \end{array} \end{array}$$

$$\begin{array}{rccl} \bigwedge: & \mathcal{P}(T^E) & \to & T^E \\ & F & \mapsto & \bigwedge F: \begin{array}{rcl} E & \to & T \\ p & \mapsto & \bigwedge\{f(p) \in T : f \in F\} \end{array} \end{array}$$

and the top element and bottom element are

$$\begin{array}{rccl} \top: & E & \to & T \\ & p & \mapsto & \top \end{array} \qquad \textit{and} \qquad \begin{array}{rccl} \bot: & E & \to & T \\ & p & \mapsto & \bot \end{array}$$

respectively.

Finally, let us recall the notion of a *complemented lattice*. Even if it is not needed for the main tools of mathematical morphology, it is useful for some particular cases and comes handy in some proofs.

Definition 1.6 (Complemented lattice)**.** Let $(L, \leq, \vee, \wedge)$ be a lattice. We say that $(L, \leq, \vee, \wedge)$ is a **complemented lattice** if there exists a function

$$\begin{array}{rccl} \cdot^c: & L & \to & L \\ & a & \mapsto & a^c \end{array}$$

called complementation, that satisfies: $\forall a \in L$,

$$a \vee a^c = \top \qquad\qquad a \wedge a^c = \bot \,.$$

In this case, we notate $(L, \leq, \vee, \wedge, \cdot^c)$.

1.1.2 Dilation and Erosion

Let us now present the most basic morphological operators: *dilation* and *erosion*.

In the following, we omit the tedious notation $(L, \leq, \bigvee, \bigwedge, \top, \bot)$ for complete lattices, and use simply L or, eventually, $(L, \leq)$. In order to be consistent, every time that L is involved, it comes with its usual order, operations and elements ($\leq$, $\vee$, $\wedge$, $\bigvee$, $\bigwedge$, $\top$ and $\bot$), and if several lattices are presented, we assign a subscript to each of them and propagate it through the order, operations and elements to keep the notations consistent (for instance, L_1, $\leq_1$, $\vee_1$, $\wedge_1$, $\bigvee_1$, $\bigwedge_1$, $\top_1$ and $\bot_1$).

Definition 1.7 (Dilation)**.** Let L_1 and L_2 be two complete lattices. We say that an operation $\delta : L_1 \to L_2$ is a **dilation** if it commutes with the supremum, i.e.,

$$\forall A_1 \subseteq L_1, \quad \delta\left(\bigvee\nolimits_1 A_1\right) = \bigvee\nolimits_2 \delta(A_1)\,, \tag{1.1}$$

where $\delta(A_1) = \{\delta(a_1) \in L_2 : a_1 \in A_1\}$.

Definition 1.8 (Erosion)**.** Let L_2 and L_1 be two complete lattices. We say that an operation $\varepsilon : L_2 \to L_1$ is an **erosion** if it commutes with the infimum, i.e.,

$$\forall A_2 \subseteq L_2, \quad \varepsilon\left(\bigwedge\nolimits_2 A_2\right) = \bigwedge\nolimits_1 \varepsilon(A_2)\,, \tag{1.2}$$

where $\varepsilon(A_2) = \{\varepsilon(a_2) \in L_1 : a_2 \in A_2\}$.

A direct consequence of Definitions 1.7 and 1.8 is that dilation and erosion are increasing operators.

Erosions and dilations usually come in pairs; if chosen properly, they form an adjunction.

Definition 1.9 (Adjunction)**.** Let $(P_1, \leq_1), (P_2, \leq_2)$ be two partially ordered sets. Let $\alpha : P_1 \to P_2$ and $\beta : P_2 \to P_1$ be two operators. We say that (β, α) is an adjunction if $\forall a_1 \in P_1, \forall a_2 \in P_2$,

$$\alpha(a_1) \leq_2 a_2 \Leftrightarrow a_1 \leq_1 \beta(a_2)\,. \tag{1.3}$$

We say that α is lower adjoint of β, and β is upper adjoint of α.

It is important to remark that, since α and β lay at opposite directions of the order symbol $\leq$, they do not play similar roles. This is why the adjunction has an order (β, α).

The next theorem expresses the equivalence between adjunctions and pairs of erosion-dilation in the case of complete lattices.

Theorem 1.10. *Let L_1 and L_2 be two complete lattices. Then,*

1. *Given $\delta : L_1 \to L_2$ and $\varepsilon : L_2 \to L_1$ such that (ε, δ) is an adjunction, δ is a dilation and ε is an erosion.*

2. *Conversely,*

   (a) *given a dilation $\delta : L_1 \to L_2$, there is a unique erosion ε such that (ε, δ) is an adjunction, given by:*

$$\begin{array}{rccl} \varepsilon : & L_2 & \to & L_1 \\ & a_2 & \mapsto & \bigvee\{a_1 \in L_1 : \delta(a_1) \leq a_2\} \end{array} , \tag{1.4}$$

   (b) *given an erosion $\varepsilon : L_2 \to L_1$, there is a unique dilation δ such that (ε, δ) is an adjunction, given by:*

$$\begin{array}{rccl} \delta : & L_1 & \to & L_2 \\ & a_1 & \mapsto & \bigwedge\{a_2 \in L_2 : a_1 \leq \varepsilon(a_2)\} \end{array} . \tag{1.5}$$

1.1.3 Opening and Closing

We first recall the general algebraic definitions of *opening* and *closing*, as particular morphological filters. Forms of these operators can be built by composition of adjoint erosion and dilation.

Definition 1.11 (Opening)**.** Let L be a complete lattice. Let $\gamma : L \to L$ be an operator such that

1. $\forall x, y \in L,\ x \leq y \Rightarrow \gamma(x) \leq \gamma(y)\,,$ (Increasing)
2. $\forall x \in L,\ \gamma(x) \leq x\,,$ (Anti-extensive)
3. $\gamma^2 := \gamma \circ \gamma = \gamma\,.$ (Idempotent)

Then γ is an **opening**.

Definition 1.12 (Closing)**.** Let L be a complete lattice. Let $\varphi : L \to L$ be an operator such that

1. $\forall x, y \in L,\ x \leq y \Rightarrow \varphi(x) \leq \varphi(y)\,,$ (Increasing)
2. $\forall x \in L,\ x \leq \varphi(x)\,,$ (Extensive)

3. $\varphi^2 := \varphi \circ \varphi = \varphi$. (Idempotent)

Then φ is an **closing**.

Openings and closings are particular cases of morphological filters. Opening, being anti-extensive, eliminates (or reduces) elements from the input, while closing, being extensive, add (or increases) elements. Furthermore, the third condition, idempotence, guarantees that all removals or additions take place in the initial iteration. This differs from conventional linear filters, where applying the same filter multiple lead to different results.

A common method of creating openings and closings is by combining dilations and erosions.

Proposition 1.13. *Let L_1 and L_2 be two complete lattices and (ε, δ) be be an adjunction with $\delta : L_1 \to L_2$ and $\varepsilon : L_2 \to L_1$. Then,*

1. *$\varphi := \varepsilon \circ \delta : L_1 \to L_1$ is a closing,*
2. *$\gamma := \delta \circ \varepsilon : L_2 \to L_2$ is an opening.*

1.2 Mathematical Morphology with Structuring Elements

Common specific forms of morphological operators are defined based on *structuring elements*. A structuring element can be viewed as a pattern that we seek to find (in the case of erosion) or replicate (in the case of dilation) throughout the input of the operator.

The most abstract approach to achieve this involves having a binary relation between elements of the domain[3]. However, in this work, we are going to introduce a group action, that naturally leads to a binary relation.

1.2.1 Action of a Group on a Set

In this section, we introduce the action of an additive[4] group $(G, +)$ on a set E. This provides a framework for utilizing the notion of a structuring element. We use (Rotman, 1994) as reference for group theory.

[3]The domain is E if the lattice is either $(\mathcal{P}(E), \subseteq)$ or $(T^E, \preceq)$.

[4]When using the additive notation for a group we assume that this group is commutative.

Definition 1.14. Let $(G, +)$ be a group with identity element 0 and E be a set. We say that a function $+ : E \times G \to E$ is an **action** if

1. $\forall p \in E,\ p + 0 = p,$ (Identity)
2. $\forall p \in E,\ \forall x, y \in G,\ (p + x) + y = p + (x + y).$ (Compatibility)

It should be noted that we deliberately performed an abuse of notation by using $+$ for both the group operation and the group action. This abuse of notation simplifies the expressions significantly and the confusion is impossible since the context makes it clear to which elements $+$ is being applied.

Intuitively, the group G *shifts* the elements of E. This concept can be compared with the definition of an affine space, where elements of the affine space (called points) are shifted by the vectors of the associated vector space. Similarly, in our case, we also refer to the elements of the set E as *points*, thus calling E the *space*, while the elements of $(G, +)$ are referred to as *shifts*.

The most basic action is the translation action of a group into itself.

Definition 1.15 (Translation action)**.** Let $(G, +)$ be a group. Then, the **translation action** of a group into itself is the function

$$\begin{array}{rccc} + : & G \times G & \to & G \\ & (x, y) & \mapsto & y + x \end{array} . \tag{1.6}$$

A straightforward approach to create an action when we have a group $(G, +)$ and a set E that are in bijective relation is by combining the translation action with the bijection.

Proposition 1.16. *Let $(G, +)$ be a group, E be a set and $\iota : E \to G$ a bijection between E and G. Then,*

$$\begin{array}{rccc} + : & E \times G & \to & E \\ & (p, x) & \mapsto & \iota^{-1}(\iota(p) + x) \end{array} \tag{1.7}$$

is a group action.

We will employ this technique to establish associations between groups and sets. In the following examples, we present the main groups that will be utilized extensively in this thesis.

Examples 1.17.

1. $(\mathbb{R}, +)$ *is an additive group. It will represent the continuous translations.*

2. $(\mathbb{Q}, +)$ *is an additive group. It will represent the fractional translations.*

3. $(\mathbb{Z}, +)$ *is an additive group. It will represent the integer translations.*

4. $(\mathbb{Z}_{12}, +)$ *is an additive torsion group. It will represent the integer translations modulo 12.*

5. *Let* $(G_1, +_1)$ *and* $(G_2, +_2)$ *be two additive groups. Then,* $(G_1 \times G_2, +)$ *is an additive group, where the sum is defined by*

$$\begin{array}{rccc} + : & (G_1 \times G_2) \times (G_1 \times G_2) & \to & G_1 \times G_2 \\ & \big((x_1, x_2), (y_1, y_2)\big) & \mapsto & (x_1 +_1 y_1, x_2 +_2 y_2) \end{array} . \tag{1.8}$$

We use these examples to illustrate translation in Figure 1.1. The figure illustrates a significant observation: in this case, the element of the set, denoted by p, represents a point, while the element of the group, denoted by x, represents a shift. Translating a point by a shift results in another point.

Whereas translation actions are defined for particular elements of a set, we can also apply them to a subset of the set; this is done by the following definition: $\forall x \in G$, $\forall A \subseteq E$,

$$A + x := \{p + x \in E : p \in A\} . \tag{1.9}$$

Similarly, $\forall B \subseteq G$, $\forall p \in E$,

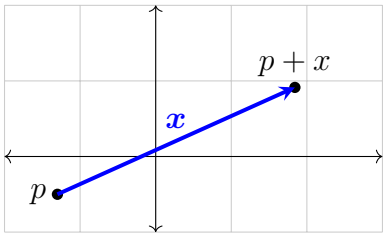
$$p + B := \{p + x \in E : x \in B\} . \tag{1.10}$$

These equations are illustrated in Figure 1.2.

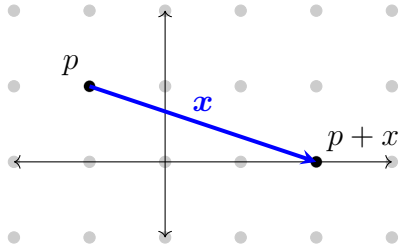
We outline that, even if the addition is commutative, the group action is not. To maintain consistency, we always place the point-like element first and the shift-like element after when performing the group action.

The case $p+B$ will be used extensively in the following: B will be the structuring element (a subset of G) and p will be each of the points of the input. Note also the importance of the 0 in this case: arrows are depicted going from 0 to the coordinates of each of the elements of B.

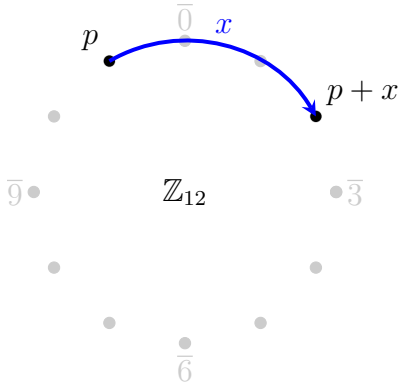
Furthermore, the fact that G is a group makes possible the reflection of B. We denote by $\check{B}$ the reflection of B given by the formula $\check{B} = \{-b \in G : b \in B\}$. This operation is illustrated in Figure 1.3.



(a) Example of a translation action on $\mathbb{R}^2$; $p = (-1.3, -0.5)$ and $x = (\pi, \sqrt{2})$.

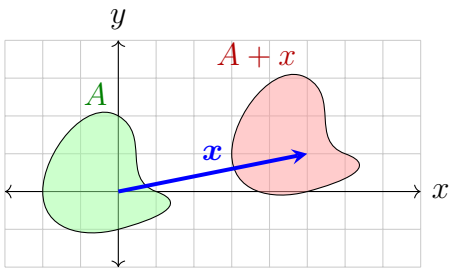


(b) Example of a translation action on $\mathbb{Z}^2$; $p = (-1, 1)$ and $x = (3, -1)$.

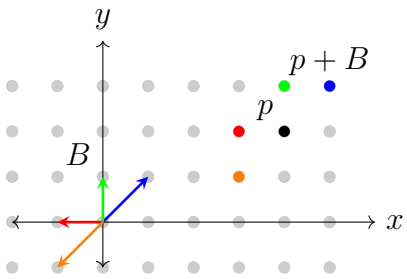


(c) Example of a translation action on $\mathbb{Z}_{12}$; $p = \overline{11}$ and $x = \overline{3}$.

Figure 1.1: Examples of translation actions.



(a) Example of the translation of a set of points A by a group element x in $\mathbb{R}^2$.



(b) Example of the translation of a point p by a set of group elements B.

Figure 1.2: Translation of sets.

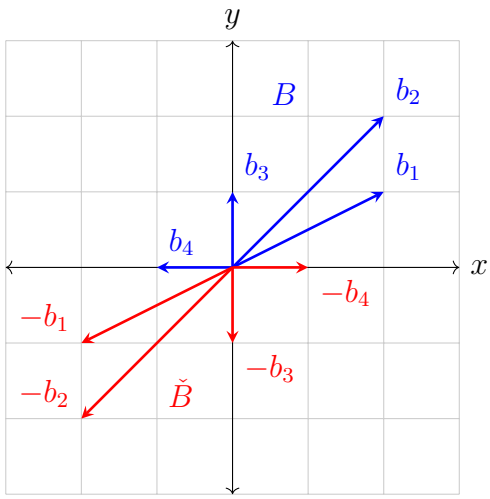


Figure 1.3: Example of the reflection $\check{B}$ of B.

1.2.2 Binary Mathematical Morphology

Binary morphology is often referred to as morphology on sets because it uses the lattice $(\mathcal{P}(E), \subseteq)$ with E as the base space. In order to apply operators that rely on structuring elements, the action of a group $(G, +)$ on E serves as an appropriate framework, and we employ it throughout this section.

Definition 1.18 (Binary dilation and erosion)**.** Let $A \subseteq E$ and $B \subseteq G$.

The **binary dilation** $A \oplus B$ of A by B is defined by:

$$A \oplus B = \{p \in E : (p + \check{B}) \cap A \neq \emptyset\} \tag{1.11}$$
$$= \{a + b \in E : a \in A,\, b \in B\}\,. \tag{1.12}$$

The **binary erosion** $A \ominus B$ of A by B is defined by:

$$A \ominus B = \{p \in E : p + B \subseteq A\}\,. \tag{1.13}$$

B is called *structuring element*. If we fix a structuring element $B \subseteq G$, we can define the binary dilation and erosion operators

$$\begin{array}{rccc} \delta_B : & \mathcal{P}(E) & \to & \mathcal{P}(E) \\ & A & \mapsto & \delta_B[A] = A \oplus B \end{array} \tag{1.14}$$

$$\begin{array}{rccc} \varepsilon_B : & \mathcal{P}(E) & \to & \mathcal{P}(E) \\ & A & \mapsto & \varepsilon_B[A] = A \ominus B \end{array} \tag{1.15}$$

that form an adjunction.

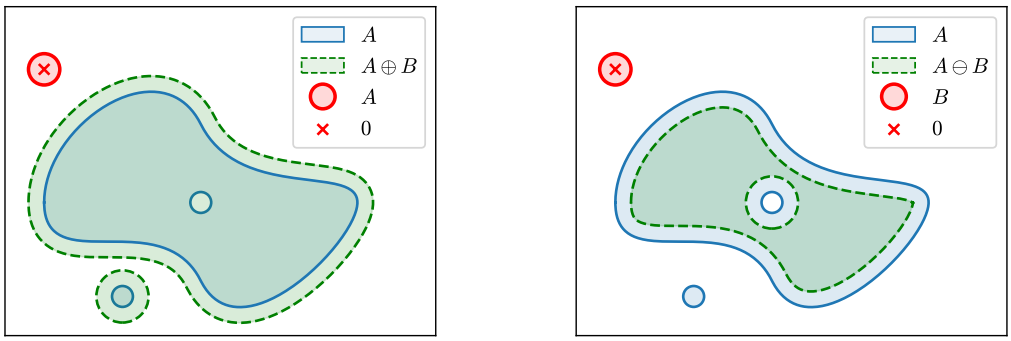
The fact that $(\mathcal{P}(E), \subseteq)$ is a complemented lattice (where the complement of an element $A \subseteq E$ is $A^c := \{p \in E : p \notin A\}$), gives an interesting duality property that link both operations.

Proposition 1.19. *Let $A \subseteq E$ and $B \subseteq G$. The following equations hold*

$$(A \ominus B)^c = A^c \oplus \check{B}^c \tag{1.16}$$

$$(A \oplus B)^c = A^c \ominus \check{B}^c\,. \tag{1.17}$$

An intuitive illustration of how these operators transform a shape is given in Figure 1.4.



(a) Dilation (b) Erosion

Figure 1.4: Illustration of binary dilation and erosion in $\mathbb{R}^2$.

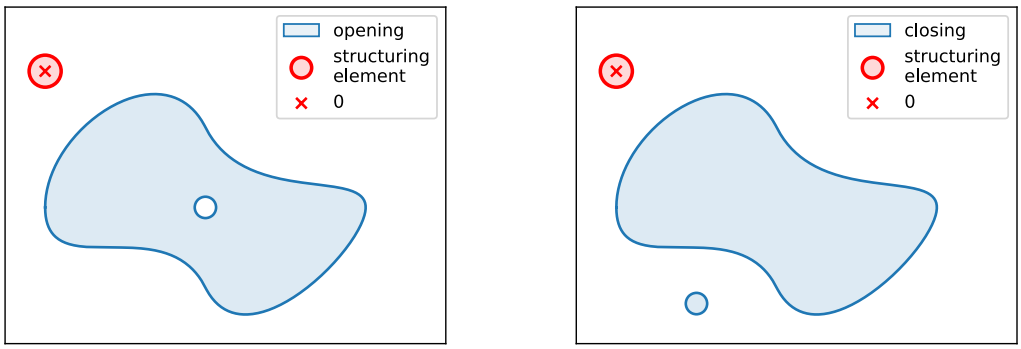
The composition of a dilation with an erosion gives either an opening or a closing, depending on the order of composition.

Proposition 1.20. *Let $A \subseteq E$ and $B \subseteq G$. Then,*

1. $\gamma_B := \delta_B \circ \varepsilon_B$ *is an opening,*
2. $\varphi_B := \varepsilon_B \circ \delta_B$ *is a closing.*

Figure 1.5 provides an intuitive interpretation of opening and closing operations. During the opening, the little circle outside the shape from Figure 1.4 has been *filtered* out, while during the closing, the little hole inside the shape has been *filled* in. This occurs because the circle and the hole are smaller (in terms of inclusion) than the structuring element. Despite these changes, the main shape of the object

remains unchanged. This preservation of the general shape is one of the most interesting features of morphological filters: they filter out some parts of the object or its complement that are smaller than the structuring element, yet the overall shape remains unaffected.



(a) Opening (b) Closing

Figure 1.5: Illustration of binary opening and closing of the shape from Figure 1.4.

1.2.3 Functional Mathematical Morphology

Functional mathematical morphology, or mathematical morphology on functions, is based on the lattice $(T^E, \preceq)$ (see Examples 1.5 2) of functions defined on a space E and with codomain T, where $(T, \leq)$ is a complete lattice. Let $(G, +)$ be a group that acts on E.

1.2.3.1 Flat morphology

Definition 1.21 (Flat dilation and erosion)**.** Let $f \in T^E$ and $B \subseteq G$.

The **flat dilation** $f \oplus B$ of f by B is defined as:

$$\begin{array}{rccc} f \oplus B: & E & \to & T \\ & p & \mapsto & \bigvee\limits_{b \in \check{B}} f(p+b) \end{array} . \tag{1.18}$$

The **flat erosion** $f \ominus B$ of f by B is defined as:

$$\begin{array}{rccc} f \ominus B: & E & \to & T \\ & p & \mapsto & \bigwedge\limits_{b \in B} f(p+b) \end{array} . \tag{1.19}$$

B is called *flat structuring element*. If we fix a flat structuring element $B \subseteq G$, we can define the flat dilation and erosion operators

$$\begin{array}{rccc} \delta_B : & T^E & \to & T^E \\ & f & \mapsto & \delta_B[f] = f \oplus B \end{array} \tag{1.20}$$

$$\begin{array}{rccc} \varepsilon_B : & T^E & \to & T^E \\ & f & \mapsto & \varepsilon_B[f] = f \ominus B \end{array} \tag{1.21}$$

that form an adjunction.

Opening and closing associated to flat erosion and dilation are defined as in Proposition 1.20.

It is interesting to note that flat morphology generalizes binary morphology through the usage of the characteristic function.

Proposition 1.22. *We recall that*

$$\begin{array}{rccc} \mathbb{1} : & \mathcal{P}(E) & \to & \{0,1\}^E \\ & A & \mapsto & \mathbb{1}_A \end{array}$$

is the canonical bijection between the subsets of E and their characteristic functions. We have that

$$\mathbb{1}_{A \oplus B} = \mathbb{1}_A \oplus B \qquad \text{and} \qquad \mathbb{1}_{A \ominus B} = \mathbb{1}_A \ominus B\,.$$

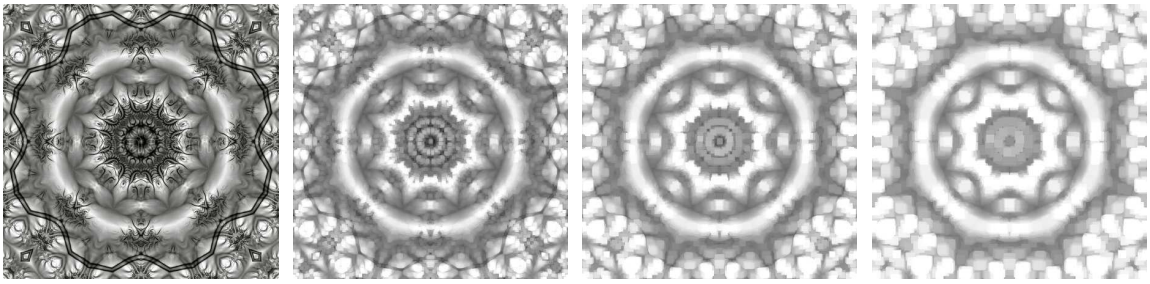
Let us illustrate the effect of these operators on an image, with $E = \mathbb{Z}^2$, $G = (\mathbb{Z}^2, +)$, and $T = \{0, 1, ..., 255\}$ (the 8-bit grayscale range).

The impact of dilation on an image can be observed in Figure 1.6 for various structuring elements. For each size[5] $k \in 2\mathbb{N} + 1$, the structuring element is the set $B = \{(x, y) \in \mathbb{Z}^2 : \max\{x, y\} \leq \frac{k-1}{2}\}$. We observe that dark details are suppressed by the dilation and that the overall image gets brighter.

In Figure 1.7, we can see the effect of erosion on the same image. In this case, the bright details are suppressed and the overall image gets darker.

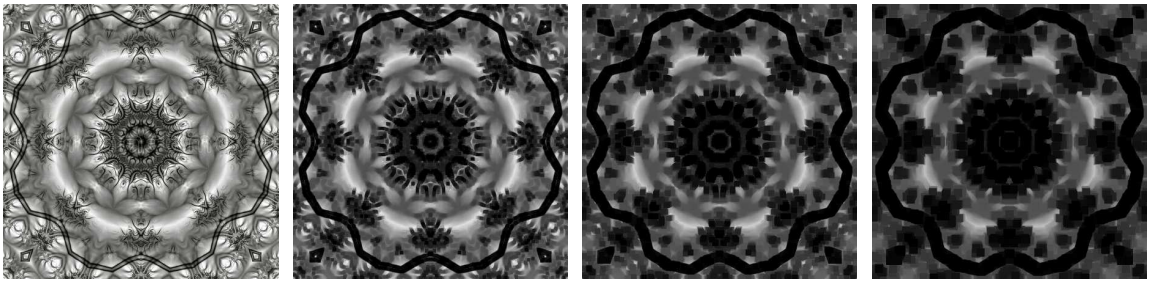
In Figure 1.8, we can observe the impact of the closing and opening defined by the compositions of erosion and dilation on the same image, with a fixed $k = 7$. For the opening, brighter details have disappeared, while the overall shape is preserved. On the other hand, the closing eliminates darker details (almost entirely); notably, the black line becomes a gray line. Despite these changes, the overall brightness level remains similar to the original image, with a slight shift towards darker (for opening) or brighter (for closing).

[5]We use the set $2\mathbb{N} + 1 = \{2n + 1 \in \mathbb{N} : n \in \mathbb{N}\}$ for referring the odd numbers. Whereas k may be even, we limit the structuring element sizes to odd numbers in order to get it centered at the origin.



(a) Input image (b) $k = 5$ (c) $k = 9$ (d) $k = 13$

Figure 1.6: Dilations of an image for different structuring elements sizes k.



(a) Input image (b) $k = 5$ (c) $k = 9$ (d) $k = 13$

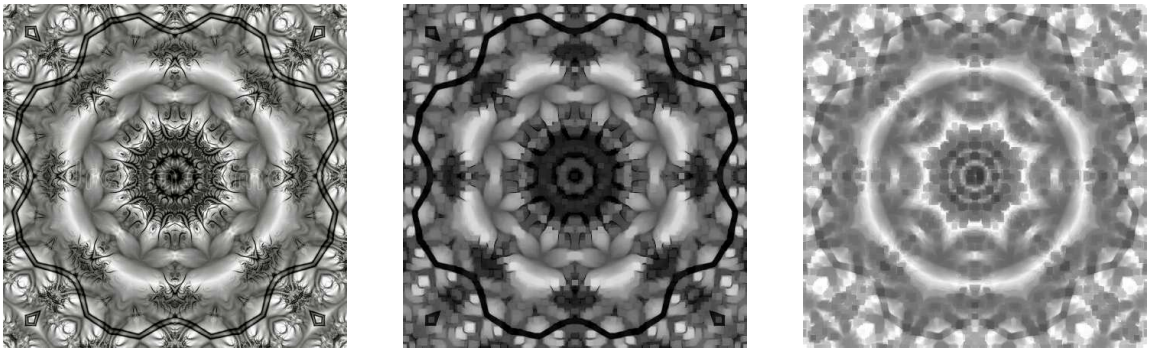
Figure 1.7: Erosions of an image for different structuring elements of size k.

1.2.3.2 Grayscale morphology

In this section, we consider grayscale morphology, that is, morphology where the structuring element is a function. The standard way of defining grayscale morphology (Heijmans & Ronse, 1990; Bloch et al., 2007; Najman & Talbot, 2010) is by using a function $b : G \to T$ and defining the dilation and erosion of $f : E \to T$ by b as: $\forall p \in E$,

$$\begin{aligned}(f \oplus b)(p) &= \sup_{x \in G} \left(f(p - x) + b(x)\right) \\ (f \ominus b)(p) &= \inf_{x \in G} \left(f(p + x) - b(x)\right).\end{aligned} \tag{1.22}$$

This definition has a flaw: we cannot assume *a priori* that addition and subtraction operations exist in the lattice T. Furthermore, in common cases like $T = \overline{\mathbb{R}}$ or $\overline{\mathbb{Z}}$, there is not canonical way to define these operations since the case $\infty - \infty$ is undefined. Bloch et al., 2007 propose to handle the case $\infty - \infty$ in a specific manner



(a) Input image (b) Opening (c) Closing

Figure 1.8: Morphological filters with a structuring element of size $k = 7$.

for each operator, stating that if $f(q)+b(p-q)$ takes the form $\infty-\infty$, it is set equal to $-\infty$, and if $f(q)-b(q-p)$ takes the form $\infty-\infty$, it is set equal to ∞.

In this work, we propose an alternative approach to address this issue. We introduce the concept of a *residuated lattice triplet*, or simply a *residuated triplet*, which is built upon the idea of a *residuated lattice*. This framework offers a solution to the problem of undefined operations.

This approach is in the line of using fuzzy sets and logic in MM (Bloch & Maître, 1994). Specifically, we introduce analogs to *conjunction* and *implication* (Deng & Heijmans, 2002) through the utilization of *lattice multiplication* and *residuals*. Our approach closely resembles the one employed in (Maragos, 2005), with the extension to triplets of lattices in place of considering a single lattice for scalars. Our adaptation draws upon results from the theory of fuzzy mathematical morphology (Bloch, 2009, 2012) due to the similarity between the two contexts.

The theory of residuated lattices started with the works of Krull, 1924 and Ward and Dilworth, 1938, 1939 and has been developed during the 20^{th} century. For a modern approach, we suggest the survey of Jipsen and Tsinakis, 2002 or the monograph of Galatos et al., 2007. We recall the definition of a residuated lattice.

Definition 1.23 (Residuated lattice)**.** Let $(L, \leq, \vee, \wedge)$ be a lattice. Let $\bullet : L \times L \to L$ be an operation that is associative and has a neutral element $\mathbf{1} \in L$, i.e., $(L, \bullet, \mathbf{1})$ is a monoid. Then, $(L, \leq, \vee, \wedge, \bullet, \mathbf{1})$ is a **residuated lattice** if there exist two operations $/ : L \times L \to L$ and $\backslash : L \times L \to L$ such that $\forall x, y, z \in L$,

$$x \bullet y \leq z \Leftrightarrow y \leq x \backslash z \Leftrightarrow x \leq z / y \,. \tag{1.23}$$

We call $\backslash$ and $/$ the *right* and *left residuals* of $\bullet$, respectively.

We adapt this definition to the case where the operations $\bullet$, $\backslash$ and $/$ are defined for different lattices and we call this a *residuated triplet*.

Definition 1.24 (Residuated triplet)**.** Let $(L_1, \leq_1)$, $(L_2, \leq_2)$ and $(L_3, \leq_3)$ be three lattices. If there exists three operations

$$\begin{array}{rccc} \bullet: & L_1 \times L_2 & \to & L_3 \\ & (a_1, a_2) & \mapsto & a_1 \bullet a_2 \\ / : & L_3 \times L_2 & \to & L_1 \\ & (a_3, a_2) & \mapsto & a_3 / a_2 \\ \backslash : & L_1 \times L_3 & \to & L_2 \\ & (a_1, a_3) & \mapsto & a_1 \backslash a_3 \end{array}$$

such that $\forall a_1 \in L_1$, $\forall a_2 \in L_2$, $\forall a_3 \in L_3$,

$$a_2 \leq_2 a_1 \backslash a_3 \Leftrightarrow a_1 \bullet a_2 \leq_3 a_3 \Leftrightarrow a_1 \leq_1 a_3 / a_2 \tag{1.24}$$

we say that they form a **residuated triplet** denoted by (L_1, L_2, L_3). The operations $/$ and $\backslash$ are called respectively *left* and *right residuals* (with respect to $\bullet$).

Residuated triplets provide a versatile framework for defining greyscale morphology. We choose to set the theory for triplets as it encompasses the most general case (relevant for some applications to music). However, if we have $L_1 = L_3$ or $L_1 = L_2 = L_3$, we will encounter particular cases that are also valuable. Notably, when $L := L_1 = L_2 = L_3$ and L has an identity element for the lattice multiplication, we obtain an actual residuated lattice.

We will now outline the conditions we desire for an operation $\bullet$ (referred to as lattice multiplication) to serve our purposes.

Definition 1.25 (Lattice multiplication)**.** Let $(L_1, \leq_1)$, $(L_2, \leq_2)$ and $(L_3, \leq_3)$ be three complete lattices. We say that an operation

$$\begin{array}{rccc} \bullet: & L_1 \times L_2 & \to & L_3 \\ & (a_1, a_2) & \mapsto & a_1 \bullet a_2 \end{array} \tag{1.25}$$

is a **lattice multiplication** if $\forall a_1 \in L_1$, $\forall a_2 \in L_2$, $\forall A_1 \subseteq L_1$, $\forall A_2 \subseteq L_2$,

1. $a_1 \bullet (\bigvee_2 A_2) = \bigvee_3 (a_1 \bullet A_2)$ (Distributive property in L_2)
2. $(\bigvee_1 A_1) \bullet a_2 = \bigvee_3 (A_1 \bullet a_2)$ (Distributive property in L_1)

where $A_1 \bullet a_2 = \{a \bullet a_2 \in L_3 : a \in A_1\}$ and $a_1 \bullet A_2 = \{a_1 \bullet a \in L_3 : a \in A_2\}$.

Corollary 1.26. *Let $(L_1, \leq_1)$, $(L_2, \leq_2)$ and $(L_3, \leq_3)$ be three complete lattices and $\bullet : L_1 \times L_2 \to L_3$ a lattice multiplication. Then, $\forall a_1, b_1 \in L_1$, $\forall a_2, b_2 \in L_2$,*

1. $a_1 \leq_1 b_1 \Rightarrow a_1 \bullet a_2 \leq_3 b_1 \bullet a_2$ *(Order preserving in L_1)*
2. $a_2 \leq_2 b_2 \Rightarrow a_1 \bullet a_2 \leq_3 a_1 \bullet b_2$ *(Order preserving in L_2)*

Proof. Let us prove 1 and the other is analogous. We know that $a_1 \leq_1 b_1$ which is equivalent to say that $b_1 = \bigvee\{a_1, b_1\}$. Then,

$$b_1 \bullet a_2 = \bigvee\{a_1, b_1\} \bullet a_2 \overset{\text{Definition 1.25 2}}{=} \bigvee\{a_1 \bullet a_2, b_1 \bullet a_2\} \Rightarrow a_1 \bullet a_2 \leq_3 b_1 \bullet a_2 \,.$$

□

These conditions ensure that the resulting operations satisfy the properties of residuals, thus forming a residuated triplet.

Definition 1.27 (Left and right residuals)**.** Let $(L_1, \leq_1)$, $(L_2, \leq_2)$ and $(L_3, \leq_3)$ be three complete lattices. Let $\bullet : L_1 \times L_2 \to L_3$ be a lattice multiplication. We define the **left** and **right residuals** of $\bullet$ by

$$\begin{array}{rccl} / : & L_3 \times L_2 & \to & L_1 \\ & (a_3, a_2) & \mapsto & a_3/a_2 := \bigvee_1 a_2 \downarrow a_3 \\ \backslash : & L_1 \times L_3 & \to & L_2 \\ & (a_1, a_3) & \mapsto & a_1 \backslash a_3 := \bigvee_2 a_1 \downarrow a_3 \end{array}$$

where $a_2 \downarrow a_3 := \{a \in L_1 : a \bullet a_2 \leq_3 a_3\}$ and $a_1 \downarrow a_3 := \{a \in L_2 : a_1 \bullet a \leq_3 a_3\}$.

Proposition 1.28. *The operations $\bullet$, $/$ and $\backslash$ defined in Definition 1.25 and Definition 1.27 make (L_1, L_2, L_3) a residuated triplet.*

Proof. We shall prove that $\forall a_1 \in L_1$, $\forall a_2 \in L_2$, $\forall a_3 \in L_3$,

$$a_2 \leq_2 a_1 \backslash a_3 \Leftrightarrow a_1 \bullet a_2 \leq_3 a_3 \Leftrightarrow a_1 \leq_1 a_3/a_2 \,.$$

Let us prove that $a_2 \leq_2 a_1 \backslash a_3 \Leftrightarrow a_1 \bullet a_2 \leq_3 a_3$. The prove of $a_1 \bullet a_2 \leq_3 a_3 \Leftrightarrow a_1 \leq_1 a_3/a_2$ is analogous.

$\Leftarrow$

Since $a_1 \bullet a_2 \leq_3 a_3$ then $a_2 \in a_1 \downarrow a_3 \Rightarrow a_2 \leq_2 \bigvee_2 a_1 \downarrow a_3 \Rightarrow a_2 \leq_2 a_1 \backslash a_3$.

$\Rightarrow$

$$
\begin{aligned}
& a_2 \leq_2 a_1 \backslash a_3 \\
\overset{\text{Corollary 1.26 2}}{\Rightarrow} \quad & a_1 \bullet a_2 \leq_3 a_1 \bullet (a_1 \backslash a_3) \\
\Rightarrow \quad & a_1 \bullet a_2 = a_1 \bullet \left(\bigvee\nolimits_2 a_1 \downarrow a_3 \right) \\
& \overset{\text{Definition 1.25 1}}{=} \bigvee\nolimits_3 (a_1 \bullet a_1 \downarrow a_3)
\end{aligned}
$$

where $a_1 \bullet a_1 \downarrow a_3 := \{a_1 \bullet a \in L_3 : a \in a_1 \downarrow a_3\}$.

Since $\forall a \in a_1 \downarrow a_3$, $a_1 \bullet a \leq_3 a_3$ then a_3 is an upper bound of $a_1 \bullet a_1 \downarrow a_3$ and thus $\bigvee_3 (a_1 \bullet a_1 \downarrow a_3) \leq a_3$. We have finally $a_1 \bullet a_2 \leq_3 \bigvee_3 (a_1 \bullet a_1 \downarrow a_3) \leq a_3$.

□

Now that we have a residuated triplet, we can define the grayscale dilation and erosion.

Definition 1.29 (Grayscale dilation and erosion)**.** Let $(T_1, \leq_1)$, $(T_2, \leq_2)$ and $(T_3, \leq_3)$ be three complete lattices that form a residuated triplet with the operations $\bullet$, $/$ and $\backslash$, where $\bullet$ is a lattice multiplication.

Let $g \in T_1^E$, $b \in T_2^G$ and $f \in T_3^E$. We define the **greyscale dilation** $g \oplus b$ of g by b as:

$$
\begin{array}{rccl}
g \oplus b : & E & \to & T_3 \\
 & p & \mapsto & \bigvee\limits_{x \in G} g(p+x) \bullet b(-x)
\end{array}
\quad . \tag{1.26}
$$

We define the **greyscale erosion** $f \ominus b$ of f by b as:

$$
\begin{array}{rccl}
f \ominus b : & E & \to & T_1 \\
 & p & \mapsto & \bigwedge\limits_{x \in G} f(p+x) / b(x)
\end{array}
\quad . \tag{1.27}
$$

b is called *structuring function*. If we fix a structuring function $b \in T_2^G$, we can define the flat erosion and dilation operators

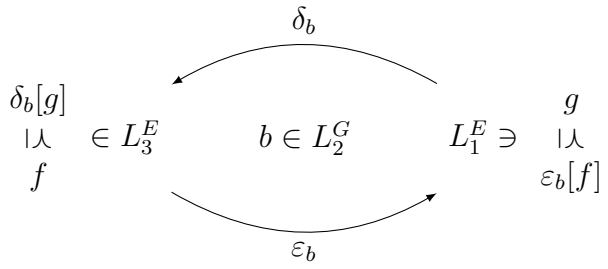
$$
\begin{array}{rccl}
\varepsilon_b : & T_3^E & \to & T_1^E \\
 & f & \mapsto & \varepsilon_b[f] = f \ominus b
\end{array}
\tag{1.28}
$$

$$
\begin{array}{rccl}
\delta_b : & T_1^E & \to & T_3^E \\
 & g & \mapsto & \delta_b[g] = g \oplus b
\end{array}
\quad . \tag{1.29}
$$

Proposition 1.30. *The greyscale erosion and dilation form an adjunction and thus are actual erosions and dilations.*

Proof. The proof is a direct consequence of the structure of residuated triplet. See (Bloch, 2009) for a similar proof in the context of conjunctions and implications. □

This is the first example in this work of an erosion and dilation with different domains; indeed, the adjunction may be represented by the following diagram:



This diagram also illustrates the anti-extensivity and extensivity properties of the composition of erosion and dilation (resulting in opening and closing, respectively), a consequence of the adjunction. By following the diagram, we have

$$\delta_b[\varepsilon_b[f]] \preceq f \qquad \text{and} \qquad g \preceq \varepsilon_b[\delta_b[g]]\,.$$

It is worth specifying what constitutes the lattice multiplication and residual in the standard definition of greyscale morphology exposed in Equation (1.22). The residuated triplet (which is actually a residuated lattice) is $(\overline{\mathbb{R}}, \overline{\mathbb{R}}, \overline{\mathbb{R}})$ with the extended addition, defined as

$$\begin{array}{rccl} + : & \overline{\mathbb{R}} \times \overline{\mathbb{R}} & \to & \overline{\mathbb{R}} \\ & (x,y) & \mapsto & x+y = \begin{cases} -\infty & \text{if } (x,y) = (-\infty,\infty) \\ -\infty & \text{if } (x,y) = (\infty,-\infty) \\ x+y & \text{otherwise} \end{cases} \end{array} \tag{1.30}$$

and the extended subtraction, defined as

$$\begin{array}{rccl} - : & \overline{\mathbb{R}} \times \overline{\mathbb{R}} & \to & \overline{\mathbb{R}} \\ & (x,y) & \mapsto & x-y = \begin{cases} \infty & \text{if } (x,y) = (\infty,\infty) \\ \infty & \text{if } (x,y) = (-\infty,-\infty) \\ x-y & \text{otherwise} \end{cases} \end{array} . \tag{1.31}$$

The extended addition happens serves as lattice multiplication (with identity element being 0) while the extended subtraction acts as its left residual (the right residual is analogous).

Although the concept of a residuated triplet is not particularly relevant in this context (since a residuated lattice would suffice), in Chapter 2 we will present a residuated triplet with different L_1, L_2, L_3 when presenting our model for piano rolls. For presenting the derived operators of MM we will restrict ourselves to the structure of residuated lattice (instead of triplet) to remain closer to the existing literature.

1.2.4 Derived Operators

So far, we have introduced the four fundamental operators of mathematical morphology: dilation and erosion, which are the basic operators, as well as opening and closing, which are morphological filters. However, it is important to note that Mathematical Morphology offers a range of other operators that provide additional capabilities. In the following sections, we present some of these additional operators, that will be employed in this work.

1.2.4.1 Hit-or-miss transform

The *hit-or-miss transform* is an operator from mathematical morphology derived from erosion. In its binary version, it uses the complementary of a set imposing specific conditions on it. As an intuition, the hit-or-miss transform with two structuring elements C and D imposes that the translation of C, $p + C$, should be contained in the set and the translation of D, $p + D$, should be contained in the complement. The following definition presents this concept formally.

Definition 1.31 (Hit-or-miss transform)**.** Let $A \subseteq E$ and $C, D \subseteq G$ with $C \cap D = \emptyset$. Then, the **hit-or-miss transform** of A by (C, D), denoted by $A \circledast (C, D)$, is given by

$$A \circledast (C, D) = \{p \in E : p + C \subseteq A \text{ and } p + D \subseteq A^c\} \tag{1.32}$$

$$= (A \ominus C) \cap (A^c \ominus D)\,. \tag{1.33}$$

We call C and D the *foreground* and *background structuring elements*, respectively.

Extending the hit-or-miss operator to greyscale morphology is not straightforward, primarily because the hit-or-miss is not an increasing operator. To address this challenge, several approaches have been proposed; some notable works in this area include (Ronse, 1996; Soille, 2002, 2013; Barat et al., 2003a, 2003b), which are summarized and unified in (Naegel et al., 2007).

In this work, we rely on the hit-or-miss transform proposed in (Soille, 2002), called there the *unconstrained hit-or-miss*, but in a version closer to the one exposed in (Naegel et al., 2007). We call it *flat hit-or-miss*.

Definition 1.32 (Flat hit-or-miss)**.** Let $(T, \vee, \wedge, +, -, \mathbf{0})$ be a residuated lattice[6]. Let $f \in T^E$ and $C, D \subseteq G$ with $C \cap D = \emptyset$. The **flat hit-or-miss** of f by (C, D), denoted by $f \circledast (C, D)$, is defined by: $\forall p \in E$,

$$(f \circledast (C, D))(p) = \big((f \ominus C)(p) - (f \oplus \check{D})(p)\big) \vee \mathbf{0}\,. \tag{1.34}$$

Our definition is slightly different from the one in (Naegel et al., 2007) by two aspects: first, we only define it for flat structuring elements and second, we handle the subtraction of infinities by the residuation operation; this leads, for instance, to a different value of $\infty - \infty$: in (Naegel et al., 2007) it is set to 0 and we set it to ∞ (when dealing with the lattice $(\overline{\mathbb{R}}, \leq)$). This yields to the same output for the thinning operation.

1.2.4.2 Thinning

We now introduce the thinning operation, which will prove to be highly useful in Chapter 3. We present together both the binary and greyscale cases, each relying on its respective hit-or-miss transform.

Definition 1.33 (Thinning)**.** Let $C, D \subseteq G$ such that $0 \in C$ and $C \cap D = \emptyset$. Let Ψ be either a set ($\Psi \subseteq E$) or a function ($\Psi \in T^E$). Then, the **elementary thinning** of Ψ by (C, D), denoted by $\Psi \bigcirc (C, D)$, is given by:

$$\Psi \bigcirc (C, D) = \Psi - (\Psi \circledast (C, D)) \tag{1.35}$$

where $-$ denotes either set difference or left residuation, depending on the nature of Ψ.

Let $(C_1, D_1, C_2, D_2, ..., C_n, D_n) \in \mathcal{P}(G)^{2n}$ be a sequence of structuring elements that we call *templates*. The application of successive elementary thinnings

$$((((\Psi \bigcirc (C_1, D_1))\bigcirc)(C_2, D_2) \bigcirc ...) \bigcirc (C_n, D_n)) \tag{1.36}$$

is called a **thinning**.

If we apply this operation iteratively until stability is reached, we obtain what is known as an **ultimate thinning**, denoted by $\bigcirc^\infty$.

[6] We use the additive notation for the residuated lattice since it is closer to our applications. The correspondence is the following: we use $+$ in the place of $\bullet$, $-$ in the place of $/$ and $\mathbf{0}$ in the place of $\mathbf{1}$.

1.2.4.3 Top-hat

The top-hat operations fall under the category of *residues* in mathematical morphology (Najman & Talbot, 2010). These operations involve taking the difference between an input image and its opening (white top-hat) or between the closing and the image (black top-hat) (Soille, 2013). As opening is anti-extensive and closing is extensive, the resulting values are positive.

Definition 1.34 (Top-hat)**.** Let $(T, \vee, \wedge, +, -, \mathbf{0})$ be a residuated lattice. Let $f \in T^E$ and $B \subseteq G$.

The **white top-hat** of f by B is given by: $\forall p \in E$,

$$\mathrm{WTH}_B[f](p) = \begin{cases} f(p) - \gamma_B[f](p) & \text{if } f(p) \neq \gamma_B[f](p) \\ \mathbf{0} & \text{if } f(p) = \gamma_B[f](p) \end{cases}. \tag{1.37}$$

The **black top-hat** of f by B is given by: $\forall p \in E$,

$$\mathrm{BTH}_B[f](p) = \begin{cases} \varphi_B[f](p) - f(p) & \text{if } \varphi_B[f](p) \neq f(p) \\ \mathbf{0} & \text{if } f(p) = \varphi_B[f](p) \end{cases}. \tag{1.38}$$

We present the definition in multiple cases because when using the residuation $-$, we have $\infty - \infty = \infty$ instead of the desired[7] $\infty - \infty = 0$.

1.2.4.4 Skeleton

Regarding the skeleton, we use Lantuéjoul formula (Lantuéjoul, 1978).

Definition 1.35 (Skeleton)**.** Let $A \subseteq E$. Then, the **skeleton** of A, denoted by $S(A)$, is the set

$$S(A) = \bigcup_{i \in \mathbb{N}} \varepsilon^i_{B_1}[A] \setminus \gamma_{B_1}[\varepsilon^i_{B_1}[A]] \tag{1.39}$$

where $B_1 \subseteq G$ is the elementary structuring element[8].

This skeleton is equal to the set of the centers of maximal balls (according to the same distance) included in A.

[7] As far as we know, this is the only limitation of residuation when formalizing addition and subtraction in morphological operators.

[8] This structuring element determines the notion of connectivity.

1.2.5 Geodesic Transformations

Up to this point, we have introduced operations that involve a single input and one or two structuring elements. Geodesic transformations, however, require two different inputs: a *marker* and a *mask*. The marker is either expanded (dilation) or shrunk (erosion), and the mask imposes limitations to this expansion or shrinking, hence the term "geodesic". We now present both geodesic dilation and erosion, as well as their corresponding reconstructions, which lead to and opening and a closing, respectively. We define the operations in the functional case since the others are restrictions of it.

1.2.5.1 Geodesic dilation

Definition 1.36 (Geodesic dilation)**.** Let $f, g \in T^E$. The **geodesic dilation of size 1** of the *marker* f with respect to the *mask* g, denoted by $\delta_g^1[f]$, is defined by:

$$\delta_g^1[f] = \delta_{B_1}[f] \wedge g \tag{1.40}$$

where $B_1 \subseteq G$ is the unit ball of the grid.

The **geodesic dilation of size** n of f with respect to g, denoted by $\delta_g^n[f]$, is defined recursively by:

$$\delta_g^n[f] = \delta_g^1[\delta_g^{n-1}[f]] \tag{1.41}$$

where $\delta_g^0[f] = f \wedge g$.

Geodesic dilation exhibits some useful properties.

Proposition 1.37. *Geodesic dilation is increasing in both arguments, extensive in the marker argument and anti-extensive in the mask argument, i.e.,*

1. $\forall f_1, f_2, g_1, g_2 \in T^E,\ f_1 \preceq f_2 \wedge g_1 \preceq g_2 \Rightarrow \delta_{g_1}^1[f_1] \preceq \delta_{g_2}^1[f_2]$,
2. $\forall f, g \in T^E,\ f \wedge g \preceq \delta_g^1[f]$,
3. $\forall f, g \in T^E,\ \delta_g^1[f] \preceq g$.

1.2.5.2 Geodesic erosion

Definition 1.38 (Geodesic erosion)**.** Let $f, g \in T^E$. The **geodesic erosion of size 1** of the *marker* f with respect to the *mask* g, denoted by $\varepsilon_g^1[f]$, is defined by:

$$\varepsilon_g^1[f] = \varepsilon_{B_1}[f] \vee g \tag{1.42}$$

where $B_1 \subseteq G$ is the elementary structuring element.

The **geodesic erosion of size** n of f with respect to g, denoted by $\varepsilon_g^n[f]$, is defined recursively by:

$$\varepsilon_g^n[f] = \varepsilon_g^1[\varepsilon_g^{n-1}[f]] \tag{1.43}$$

where $\varepsilon_g^0[f] = f \vee g$.

Geodesic erosion exhibits some useful properties.

Proposition 1.39. *Geodesic erosion is increasing in both arguments, anti-extensive in the marker argument and extensive in the mask argument, i.e.,*

1. $\forall f_1, f_2, g_1, g_2 \in T^E,\ f_1 \preceq f_2 \wedge g_1 \preceq g_2 \Rightarrow \varepsilon_{g_1}^1[f_1] \preceq \varepsilon_{g_2}^1[f_2]$,
2. $\forall f, g \in T^E,\ \varepsilon_g^1[f] \preceq f \vee g$,
3. $\forall f, g \in T^E,\ g \leq \varepsilon_g^1[f]$.

1.2.5.3 Morphological reconstruction

In practical situations, geodesic dilation and erosion are typically applied iteratively until stability is achieved. This iterative process enables us to define morphological reconstructions, specifically the *reconstruction by dilation* and *reconstruction by erosion*.

Definition 1.40 (Reconstruction by dilation)**.** Let $f, g \in T^E$. Then, the **reconstruction by dilation** of the *mask* g from the *marker* f, denoted[9] by $\delta_f^\infty[g]$, is defined by:

$$\delta_f^\infty[g] = \delta_g^N[f] \tag{1.44}$$

where $N \in \mathbb{N}$ is such that $\delta_g^N[f] = \delta_g^{N+1}[f]$. When E is finite, N always exists.

Definition 1.41 (Reconstruction by erosion)**.** Let $f, g \in T^E$. Then, the **reconstruction by erosion** of the *mask* g from the *marker* f, denoted by $\varepsilon_f^\infty[g]$, is defined by:

$$\varepsilon_f^\infty[g] = \varepsilon_g^N[f] \tag{1.45}$$

where $N \in \mathbb{N}$ is such that $\varepsilon_g^N[f] = \varepsilon_g^{N+1}[f]$. When E is finite, N always exists.

Proposition 1.42. *The reconstruction by dilation is an opening and the reconstruction by erosion is a closing.*

[9] To emphasize the role of each operand, we have permuted the order of the operands.

1.3 Implementation of Mathematical Morphology Operators

In this section, we will discuss the implementation of the mathematical morphology operators discussed in the previous sections.

We have chosen to use Python 3 (Rossum & Drake, 2009) for the implementations and applications in this thesis. There are several libraries in Python that implement MM operators; we found four: SciPy (Virtanen et al., 2020), scikit-image (Walt et al., 2014), OpenCV (Bradski, 2000) and Kornia (Riba et al., 2020).

However, these libraries do not fully align with our needs and thus we have chosen to implement our own libraries: `PyMorpho`[10], serving as a general-purpose MM library, and `nnMorpho`[11], which is equipped with a PyTorch engine for GPU acceleration and seamless integration with neural network architectures.

In the following section, we expose the main implementation considerations specific to our work. Before delving into the computational aspects, let us recapitulate the mathematical objects that we intend to implement computationally. We require a space E equipped with an additive group $(G, +)$ that acts on it. Furthermore, we need a lattice $(T, \leq)$ to serve as the range for functional morphology. For implementing greyscale morphology, we must also have a residuated triplet (T_1, T_2, T_3).

1.3.1 Implementation Considerations

There are several aspects that should be decided when implementing a library for mathematical morphology operators, namely: the data structure, the data types, the operators families (binary, flat, grayscale), the management of the origin, the management of the border, the dimensions and the topology of the underlying space.

1.3.1.1 Data structure

The elements on which we apply mathematical morphology operators can be mathematically represented as either sets ($A \subseteq E$) or functions ($f \in T^E$). In a computational context, both sets and functions can be modeled using *arrays*.

An array is a collection of elements, typically of the same data type, that are indexed by a set of integers. Each element in the array can be accessed using its corresponding index. Arrays allow for efficient storage and retrieval of data, making them well-suited for representing sets and functions in computational settings.

[10] https://github.com/Manza12/PyMorpho
[11] https://github.com/Manza12/nnMorpho

For instance, a set can be represented as a binary array where each element of the base space E corresponds to an index in the array. If an element is present in the set, its corresponding entry in the array is marked as `true`; otherwise, it is marked as `false`. In the case of functions, the value of the array at a specific index represents the value of the function.

1.3.1.2 Data types

The data type represents our lattice $(T, \leq)$; indeed, a data type that serves our purposes need to be equipped with an order and with arithmetic operations, as well as bottom and top elements. The most frequent data types are listed in Table 1.1.

Data type	Alias	Bottom ($\bot$)	Top ($\top$)
Boolean	`bool`	`false`	`true`
8-bit integer (unsigned)	`uint8`	0	255
8-bit integer (signed)	`int8`	-128	127
16-bit integer (signed)	`int16`	$-32\,768$	$32\,767$
32-bit integer (signed)	`int32`	$\approx -2 \times 10^9$	$\approx 2 \times 10^9$
64-bit integer (signed)	`int64`	$\approx -9 \times 10^{18}$	$\approx 9 \times 10^{18}$
16-bit floating point	`float16`	$-\infty$	∞
32-bit floating point	`float32`	$-\infty$	∞
64-bit floating point	`float64`	$-\infty$	∞

Table 1.1: Common data types.

These data types come with the order operator `<=` and the arithmetic operators `+` and `-`, which will serve as our lattice multiplication and left residuation, respectively. We recall that the lattice multiplication will have the additive form since it is the most common in the greyscale MM literature.

It is important to note that critical cases of addition and subtraction, such as $\infty - \infty$ in floating-point numbers and overflow in integers, are not handled as they should in some implementations. For instance, in the case of SciPy, these operations may result in `nan` values in floating-point numbers or in modular overflow in integers, as exemplified by $255 + 2 = 1$ in `uint8` data type.

1.3.1.3 Operators families

We have presented three families of operators: binary, flat and grayscale. Usually, libraries distinguish between binary and greyscale morphology, and consider flat

morphology as a particular case of greyscale morphology[12]. In our libraries, we will also distinguish between binary and greyscale, but we do this way because of different reasons.

In principle, by the data types of the input data (the input array and the structuring element) we should be able to infer the family of the operator, namely[13]

Input data type	Structuring element data type	Family
Boolean	Boolean	Binary
Numeric	Boolean	Flat
Numeric	Numeric	Greyscale

.

However, for our applications we require another type of morphology for numeric data types, that actually corresponds to the residuated triplet presented in the following example.

Example 1.43. *Let $(T, \leq)$ be a lattice and $(\{0,1\}, \leq)$ be the boolean lattice. Then, the residuated triplet $(\{0,1\}, T, T)$ is defined with the following lattice multiplication[14]:*

$$\begin{aligned} \bullet: \ \{0,1\} \times T &\to T \\ (b, x) &\mapsto b \bullet x = \begin{cases} x & \text{if } b = 1 \\ \bot & \text{if } b = 0 \end{cases} \end{aligned} \tag{1.46}$$

which has the corresponding left residuation

$$\begin{aligned} / : \ T \times T &\to \{0,1\} \\ (x, y) &\mapsto x/y = \begin{cases} 1 & \text{if } y \leq x \\ 0 & \text{if } y \not\leq x \end{cases} \end{aligned} \quad . \tag{1.47}$$

This residuated triplet will prove to be useful for certain musical applications. From a computational perspective, it corresponds to choosing a numeric range for the data type of the input and the structuring element, and producing a Boolean output. This functionality can be encapsulated in two modules, namely `binary` and `greyscale`:

Module	Input dtype	Str. ele. dtype	Output dtype	Family
`binary`	Boolean	Boolean	Boolean	Binary
`binary`	Numeric	Numeric	Boolean	Greyscale
`greyscale`	Numeric	Boolean	Numeric	Flat
`greyscale`	Numeric	Numeric	Numeric	Greyscale

[12]Some of them only implement flat morphology.

[13]In the following table, we use the term *numeric* to refer to all types except the Boolean type.

[14]In this case we use multiplicative notation for the lattice multiplication and residuation since it is more appropriated.

1.3.1.4 Origin

Since the domain of the structuring element is a group $(G, +)$, there is a notion of *origin*. The origin is the identity element for the group (in our case, it is usually denoted as $\mathbf{0}$). When using arrays, they do not come with an associated origin (which is not a problem for a space E). As a result, we need to specify the origin of the array.

In common applications, the origin is typically set as the central element of the structuring element array, which is usually assumed to be of odd size. However, this approach is not sufficient in general and do not suit our musical applications.

In our implementation, we add a parameter for specifying which pixel is the origin, set by default to the center.

1.3.1.5 Border

Another important element to consider is the border. When the structuring element is not limited to a single pixel, translating the structuring element near the border of the input array can cause an overflow of indices. This is a consequence of working with a finite subset of the space.

To manage this overflow, there are two classical approaches:

1. the *Euclidean* approach: in this approach, we consider that the input array is extended with bottom elements outside the actual image. This is equivalent to considering that our image is defined in the entire space with compact support.

2. the *geodesic* approach: in this approach, we do not consider the values outside the input array when taking the infimum or supremum. This approach is called geodesic because of its similarities with geodesic operators.

In practice, this overflow issue only affects the computation of erosion: the dilation operation is not affected since extending with bottom elements does not alter the output of a supremum.

1.3.1.6 Dimensions

Even though our main focus will be on using two-dimensional inputs, it is important to note that mathematical morphology is defined for every space E with a group $(G, +)$ acting on it, including cases where $E = G = \mathbb{R}^d$ or $\mathbb{Z}^d$. In particular, mathematical morphology is extensively used in processing 3D scan images.

1.3.1.7 Topology of the underlying space

Every Mathematical Morphology library considers the arrays as being embedded in a Euclidean space like $\mathbb{R}^d$ or $\mathbb{Z}^d$. However, in Chapter 5, in Section 5.3 we will use a space $E \simeq \mathbb{Z} \times \mathbb{Z}_{12}$, which has cylindrical topology. This has a deep impact on the implementation of the operators since the translation in the direction of $\mathbb{Z}_{12}$ has no border and rotates the structuring element.

Although this behavior can be simulated by stacking copies of the input array, our libraries offer the flexibility to consider other topologies, particularly the cylindrical (or toroidal) topology.

1.3.2 Computational Model

In this section, we present the computational model implemented in our libraries, `PyMorpho` and `nnMorpho`. We will only discuss the abstraction needed for applying mathematical morphology with structuring elements, which forms the core of `PyMorpho`. The low-level algorithms in `nnMorpho` are mere adaptations of these abstractions, implemented as C++ extensions for PyTorch, including CUDA kernels.

The computational model presented below, with a syntax similar to that of Python, exposes the objects needed for implementation. We employ the terms *shift*, *point*, and *level* to refer to elements of the *group*, *space*, and *lattice*, respectively. The term *image* pays homage to the origins of MM but should be understood as the object upon which MM operations can be performed.

We expose one by one the classes we use with their attributes and methods[15] and give a brief explanation of them.

```
Shift:
    __neg__(self) -> Shift
```

`Shift` corresponds to an element of the group $x \in G$. We override the operator `-` for being able to refer to its opposite $-x$, needed in the definition of dilation.

```
Group:
    shift_type: Type[Shift]
    __iter__(self) -> Iterator[Shift]
```

`Group` corresponds to the group itself, $(G, +)$. We make it iterable for being able to go through each one of its elements, that are `Shift`s. We also include the `shift_type` as an attribute to link the group with its shift type[16].

[15] We use the notations of Python for the methods that override operators.

[16] For instance, we might use the constructor in the iteration.

```
Point:
    __add__(self, shift: Shift) -> Point
```

`Point` corresponds to an element of the space $p \in E$. We override the operator `+` that takes a `Shift` as second argument for being able to refer to compute the value $p + x$.

```
Space:
    point_type: Type[Point]
    __iter__(self) -> Iterator[Point]
```

`Space` corresponds to the space itself, E. We make it iterable for being able to go through each one of its elements, that are `Point`s. We also include the `point_type` as an attribute to link the space with its point type[16].

```
Level:
    __add__(self, other: Level) -> Level
    __sub__(self, other: Level) -> Level
    __le__(self, other: Level) -> bool
```

`Level` corresponds to an element of the lattice $t \in T$. We override the operators `+` and `-` for having a lattice multiplication and left residuation[17]. Moreover, we impose the levels to be comparable by the operator `<=`, even is this is not necessary for the implementation of operators.

```
Lattice:
    level_type: Type[Level]
    bot: Level
    top: Level
    supremum(a: Level, b: Level) -> Level
    infimum(a: Level, b: Level) -> Level
    __mul__(self, other: Lattice) -> Lattice
    __truediv__(self, other: Lattice) -> Lattice
```

`Lattice` corresponds to the lattice itself, $(T, \leq)$. We require it to have both the `bot` and `top` elements, that are `Level`s and to define two methods, `supremum` and `infimum`. We also include the `level_type` as an attribute to link the lattice with its level type[16]. In addition, we override the operators `*` and `/` to be able to determine the resulting lattice in lattice multiplication and left residuation.

[17]Notice that whereas the operators have arguments and output that are `Level`s, they may not be the same `Level` type, allowing for residuated triplets.

```
Image:
    array: numpy.ndarray
    space: Space
    lattice: Lattice
    __getitem__(self, point: Point) -> Level
    __setitem__(self, point: Point, value: Level)
```

`Image` corresponds to the input[18]. The accessing and assignment operators rely on `Point` and `Level`, elements of its attributes `space` and `lattice`.

```
StructuringElement:
    array: numpy.ndarray
    group: Group
    lattice: Lattice
    __getitem__(self, shift: Shift) -> Level
```

`StructuringElement` corresponds to the structuring element[19]. The accessing operator rely on `Shift` and `Level`, elements of the `group` and `lattice` attributes.

This abstraction allows us to implement the two fundamental operators of Mathematical Morphology: dilation and erosion. The general (suboptimal) algorithms for these operators are presented in Algorithms 1 and 2, and based on Equation (1.22):

$$(f \oplus b)(p) = \sup_{x \in G} (f(p - x) + b(x))$$
$$(f \ominus b)(p) = \inf_{x \in G} (f(p + x) - b(x))$$

It is important to note that this abstraction reflects the fact that we do not necessarily require a full group $(G, +)$ acting on E for defining MM operators. The only essential aspect we need is a notion of *neighborhood*, represented by the methods `add` and `neg` from `Point` and `Shift`, which can be transformed into `add` and `sub` associated with `Point`.

These algorithms, along with some examples of their applications, are implemented in the `PyMorpho` library. However, while these algorithms demonstrate decent performance when applied to small arrays (such as piano rolls), their implementation in pure Python with a high level of abstraction renders them less suitable for practical applications involving large arrays (as is often the case with spectrograms).

[18]We use the term *image* but another (overused) term would be *function*.

[19]We use the term *structuring element* as is more frequent in the literature than *structuring function*.

Algorithm 1 Algorithm for dilation

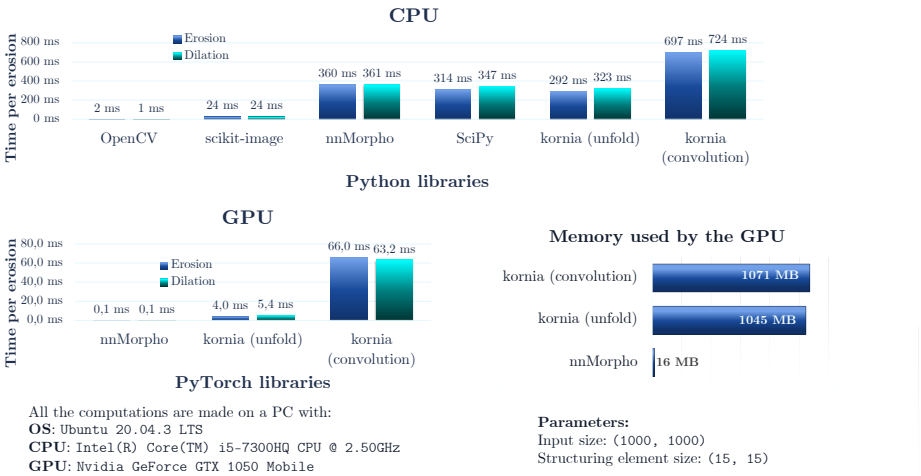
```
function DILATION(image: Image, str_el: StructuringElement)
    lattice = image.lattice * str_el.lattice
    output = Image(numpy.empty_like(image.array), image.space, lattice)
    for point in image.space do
        val ← lattice.bot
        for shift in str_el.group do
            tmp ← image[point + (-shift)] + str_el[shift]
            val ← lattice.supremum(tmp, val)
        end for
        output[point] = val
    end for
    return output
end function
```

Algorithm 2 Algorithm for erosion

```
function EROSION(image: Image, str_el: StructuringElement)
    lattice = image.lattice / str_el.lattice
    output = Image(numpy.empty_like(image.array), image.space, lattice)
    for point in image.space do
        val ← lattice.top
        for shift in str_el.group do
            tmp ← image[point + shift] - str_el[shift]
            val ← lattice.infimum(tmp, val)
        end for
        output[point] = val
    end for
    return output
end function
```

To address this limitation, we have developed the `nnMorpho` library, which includes GPU acceleration for some of the most commonly used cases.

In Figure 1.9, we present an overview of `nnMorpho`'s performance and features in comparison to other libraries, presented as a poster in DGMM (Romero-García, 2022). `nnMorpho` surpasses all other libraries when utilized with GPU acceleration. Furthermore, it offers extensive customization options.



Feature	SciPy	scikit-image	OpenCV	kornia	nnMorpho
PyTorch	✗	✗	✗	✓	✓
Non-flat structuring element	✓	✓	✗	✓	✓
GPU capability	✗	✗	✗	✓	✓
Border parameter	✗	✗	✗	✓	✓
Cylindric topology	✗	✗	✗	✗	✓
Batch processing	✗	✗	✗	✓	✓
More than 2D	✓	✗	✗	✗	✗
Computation of gradients	✗	✗	✗	✓	✓
Origin parameter	✓	✓	✗	✓	✓

(b) Features

Figure 1.9: `nnMorpho` compared to other Python libraries.

Chapter 2

Time-Frequency Representations of Music

In this chapter, our objective is to present various time-frequency representations of music that allow us to apply mathematical morphology to them. We will refer to each representation as a musical space, denoted as $\mathcal{M}$, and it will have the form:

$$\mathcal{M} = \mathcal{A}^{\mathcal{T}\times\mathcal{F}} = \{f : \mathcal{T} \times \mathcal{F} \to \mathcal{A}\} \tag{2.1}$$

where $\mathcal{T}$ represents time, $\mathcal{F}$ represents frequency, and $\mathcal{A}$ represents amplitude. Each element f in the musical space $\mathcal{M}$ is a function that maps a pair of time and frequency (t, ξ) to an amplitude value $f(t, \xi)$.

If we endow $\mathcal{A}$ with a complete lattice structure and we define a group $(G_{\mathcal{T}\times\mathcal{F}}, +)$ that acts on $\mathcal{T} \times \mathcal{F}$, we are able to apply greyscale MM with structuring elements. In order to achieve this, we consider separately time and frequency in a first instance, create groups that act on each of them, and then couple them through the Cartesian product.

In Section 2.1, we explore the various choices available for representing time, frequency and amplitude, and the resulting mathematical structures that arise from these choices. Subsequently, we delve into the two primary representations that we will employ in our applications: spectrograms, that are covered in Section 2.2, and piano rolls, discussed in Section 2.3.

2.1 Algebraic Structures for Musical Spaces

The algebraic structures underlying musical spaces play a crucial role in defining how music can be represented and analyzed. In this section, we explore the different

choices available for each element in a musical space. By examining various combinations of these choices, we will uncover the diverse representations that can be achieved for musical data in this manner.

For the domain of the functions $f \in \mathcal{M}$, which represents the time-frequency plane, we use a space $\mathcal{T} \times \mathcal{F}$ with a group $(G_{\mathcal{T}\times\mathcal{F}}, +)$ acting on it (see Section 1.2.1). This choice is related to the Generalized Music Interval model proposed by Lewin, 1987.

It is worth mentioning that other approaches, such as the use of the differential geometry paradigm proposed by Tymoczko, 2009, 2016, would offer more refined possibilities, particularly when considering spaces with borders. However, for the spaces considered in this work, the use of a group defined for the entire space is adequate and sufficiently flexible.

For the codomain of f, we use a complete lattice $(\mathcal{A}, \leq)$ that models the amplitude. This endows $(\mathcal{M}, \preceq)$ with the complete lattice structure induced by the pointwise order.

2.1.1 Time

In order to achieve the desired algebraic structure for representing time, we introduce a space $\mathcal{T}$, the elements of which are referred to as *timestamps*, and a group $(G_{\mathcal{T}}, +)$, the elements of which are known as *time shifts*. This combination will enable us to define the necessary operations for time manipulation and analysis within the musical space.

Indeed, time can be measured using different units, and we can represent each of these units using different mathematical structures. Here we consider three ways of representing time:

1. time measured in seconds, that can be modeled using the set of real numbers $\mathbb{R}$,

2. time measured in a computational unit (such as samples or MIDI ticks), that can be modeled using the set of integers $\mathbb{Z}$,

3. time measured in wholes[1], that can be modeled using the set of rational numbers $\mathbb{Q}$.

[1] A whole is a note value in music, represented by $\mathbb{o}$, that can be associated to the value $1 \in \mathbb{Q}$.

2.1.1.1 Measuring time in seconds

For measuring time in seconds, we assume that time is continuous and use $(G_{\mathcal{T}}, +) = (\mathbb{R}, +)$ as the group for time shifts. The space of timestamps, denoted as $\mathcal{T}_{\text{s}}$, represents time in seconds elapsed from a particular reference point, such as the start of a piece of music. This leads to a straightforward bijection[2] between $\mathcal{T}_{\text{s}}$ and $(\mathbb{R}, +)$ which, as shown in Proposition 1.16, induces a group action of $(\mathbb{R}, +)$ on $\mathcal{T}_{\text{s}}$.

The timestamps can be represented in the format[3] `hh:mm:ss.ms`, following the ISO standard ISO, 2019a, or using abbreviations like `mm:ss` or `ss.ms`. This split into hours, minutes, and seconds makes the action of $(\mathbb{R}, +)$ on $\mathcal{T}_{\text{s}}$ more complex, as it requires successive Euclidean divisions. Nevertheless, we choose this representation as opposed to providing the full number of seconds, which may be overwhelming in a long excerpt.

2.1.1.2 Measuring time in computational units

When working with digital representations of music, such as `.wav` or `.midi` files, we use a discrete representation of time. In this context, we consider the group for time shifts as $(G_{\mathcal{T}}, +) = (\mathbb{Z}, +)$. For representing timestamps, we use $\mathcal{T}_1 = \mathbb{Z}$, where $t \in \mathcal{T}_1$ counts the number of units elapsed from the start of the file.

In the context of `.wav` files, time is measured in samples, and the sampling frequency provides the conversion factor between the number of samples and seconds. The common sampling frequencies are 44.1 kHz or 48 kHz.

For `.midi` files, time is measured in ticks. To convert between ticks and seconds, we need the information about the ticks per beat and the beats per minute. These parameters allow us to determine the tempo and perform the conversion from ticks to seconds. We will delve into this conversion process in more detail in Section 2.3.2.1.

2.1.1.3 Measuring time in wholes

Measuring time inside a musical score is indeed a critical task, and different approaches can be taken based on the specific requirements and goals of the analysis.

One possible approach is to use the tempo information provided in the score to transform all note durations into seconds and then measure time using seconds as

[2] If the start of the piece is called $t_0 \in \mathcal{T}_{\text{s}}$, the bijection is $\mathcal{T}_{\text{s}} \to \mathbb{R}$, $t \mapsto e(t_0, t) \in \mathbb{R}$, where $e(t_0, t)$ is the time elapsed between t_0 and t measured in seconds.

[3] In our computations, we limit the representation to millisecond precision for practical reasons. However, it is important to note that, in theory, we could refine the time precision as much as needed since the space is bijective with $\mathbb{R}$.

the unit. This method can be valid and straightforward when tempo information is available and consistent throughout the score. However, it does have some limitations. For example, if the tempo indication is not provided or if there are frequent tempo changes throughout the score, this approach may become less practical and relevant.

Alternatively, another approach is to take advantage of the existing abstraction of note values in the score. To do that, we measure durations, i.e., time shifts, using note values. Some common note values are 𝅝, 𝅗𝅥, ♩, ♪ and 𝅘𝅥𝅯, and further refined note values exist. They are defined by the relation

$$𝅝 = 2\,𝅗𝅥 = 4\,♩ = 8\,♪ = 16\,𝅘𝅥𝅯\,. \tag{2.2}$$

As pointed out by Equation (2.2), all the note values are measured in terms of the whole note 𝅝. In fact, we can re-arrange Equation (2.2) as

$$𝅗𝅥 = \frac{1}{2}\,𝅝, \qquad ♩ = \frac{1}{4}\,𝅝, \qquad ♪ = \frac{1}{8}\,𝅝, \qquad 𝅘𝅥𝅯 = \frac{1}{16}\,𝅝\,. \tag{2.3}$$

It is natural then to associate note values with the rational numbers $\mathbb{Q}$, and consider then the group $(\mathbb{Q}, +)$ to be the group $(G_{\mathcal{T}}, +)$. Notice that durations can be different from negative powers of 2; if we want to define the duration that is equal to $\frac{3}{4}\,𝅝$, the symbol 𝅗𝅥‿♩ might be used. The use of tuplets allows us to consider base powers different from 2 and, in theory, we are able to create a specific note value that has the duration $\frac{p}{q}\,𝅝$ for every $\frac{p}{q} \in \mathbb{Q}$.

Whereas in previous measuring options we decided to fix a starting point and measure time from it, in this case we try to stick to musicological canons; thus, we avoid naming timestamps by counting the elapsed wholes from the starting point.

When musicians need to specify a particular time in the score, they use *bars*. The first bar is given the value 1, and if an anacrusis is present in the score, we assign it the value 0. To be more precise in the specification of time, musicians use *beats* to specify timestamps inside a bar. A beat represents the number of beats[4] elapsed from the start of the bar, starting at 1.

For example, in a $\frac{4}{4}$ time signature, where the beat is ♩, there are four beats (1, 2, 3, and 4), each of them one beat apart from the previous. We set the beat of a time signature to be the note value corresponding to the denominator of the time signature. For instance, the beat of $\frac{4}{4}$ is ♩ and the beat of $\frac{6}{8}$ is ♪. However, it is important to note that this convention deviates slightly from musical standards, where ♩. is the usual beat associated to $\frac{6}{8}$. We adopt this simplified convention for practical reasons.

[4]Note the overload of the term *beat*, referring to both a timestamp and a time shift.

Whereas this notation may be enough for communicating timestamps between musicians, we introduce a refined notation for musicological purposes. In this notation, we add an additional element called the *offset*, which allows us to fully determine every timestamp in a piece. The offset represents the displacement from the beat and starts at 0 (finally in alignment with computational conventions). By incorporating the offset, we can specify timestamps in a more precise and flexible manner, enabling accurate representation and analysis of musical data.

The time space that we use is $\mathcal{T}_{\mathbf{o}}^{\frac{p}{q}}$, where $\frac{p}{q}$ is the corresponding time signature[5]. It can be defined as

$$\mathcal{T}_{\mathbf{o}}^{\frac{p}{q}} = \{(m, b, o) \in \mathbb{N} \times \mathbb{N}^* \times \mathbb{Q} : b \in \{1, 2, ..., p\},\ 0 \leq o < \frac{1}{q}\}\,. \tag{2.4}$$

This refined representation allows for precise specification of timestamps within a musical piece, remaining close to common musical nomenclature.

Let us provide an example to clarify this notation.

Example 2.1. *We consider the beginning of the Violin Sonata No.1 in G minor, BWV 1001 from Johann Sebastian Bach (see Figure 2.1). We use the space* $\mathcal{T}^{\mathbf{C}} := \mathcal{T}^{\mathbf{4}}_{\mathbf{4}}$. *The following list*[6] *shows when the onsets of the consecutive notes*[7] *occur:*

1. Gm $\to (1, 1, 0)$,
2. F5 $\to (1, 2, \frac{1}{32})$,
3. E♭5 $\to (1, 2, \frac{2}{32})$,
4. D5 $\to (1, 2, \frac{3}{32})$,
5. C5 $\to (1, 2, \frac{4}{32})$,
6. B♭4 $\to (1, 2, \frac{5}{32})$,
7. A4 $\to (1, 2, \frac{6}{32})$,
8. B♭4 $\to (1, 2, \frac{7}{32})$,
9. G4 $\to (1, 2, \frac{15}{64})$.

It is worth to mention that this system does not lead to a straightforward action between the group $(\mathbb{Q}, +)$ and the space $\mathcal{T}_{\mathbf{o}}^{\frac{p}{q}}$. It can be compared to the hexadecimal approach of minutes and hours used for time. While measuring time in wholes and keeping it simple would have been an option, we chose to adhere to musicological standards.

2.1.2 Frequency

For representing frequency, we also require a space $\mathcal{F}$ and a group $(G_{\mathcal{F}}, +)$ acting on it. In this context, the elements of the space are *pitches* or *chromas*, depending

[5]Notice that we assume the time space is defined by a single time signature. This may not be the case in a musical piece, but we left more complex cases for future reasearch.

[6]See Section 2.1.2.2 for the notation used for notes and chords.

[7]The first chord, Gm, has a single time associated since it is played at once.

Figure 2.1: Beginning of the Violin Sonata No.1 in G minor, BWV 1001 from Johann Sebastian Bach.

on the associated space, while the elements of the group are *frequency shifts*. As in the case of time, this choice is closely related to the Generalized Music Interval framework (Lewin, 1987).

Frequency can be measured using different units; we explore two options:

1. frequency measured in Hertz,

2. frequency measured in semitones.

Depending on the specific use cases, the groups used for each of these units may vary. In the following, we detail these options.

2.1.2.1 Measuring frequency in Hertz

In signal processing, the unit commonly used to measure frequency is Hertz, which corresponds to cycles per second. If we assume that the frequency space is continuous, we can define $\mathcal{F}_{\text{Hz}} = \mathbb{R}$ as the space of frequencies measured in Hertz. We naturally assign $(\mathbb{R}, +)$ as the group acting on it.

However, in some cases, the frequency varies logarithmically, such as when using the Constant-Q transform (CQT) (see Section 2.2.1.2). In such scenarios, the space of frequencies is still measured in Hertz but restricted to positive frequencies, leading us to define $\mathcal{F}_{\text{Hz}}^{\text{log}} = \mathbb{R}^{+*}$. In this case, the group acting on it is $(\mathbb{R}^{+*}, \cdot)$.

2.1.2.2 Measuring frequency in semitones

In Western classical music, the common practice is to measure frequency in semitones. In this case, the set of frequencies $\mathcal{F}$ will consist of the pitches used in Western Classical Music, denoted by $\mathcal{N}$. These pitches are a combination of the octave on

which they are played and one of the twelve chromas[8]:

$$\mathcal{N}_{12} = \{\mathrm{C}, \mathrm{C}\sharp, \mathrm{D}, \mathrm{E}\flat, \mathrm{E}, \mathrm{F}, \mathrm{F}\sharp, \mathrm{G}, \mathrm{A}\flat, \mathrm{A}, \mathrm{B}\flat, \mathrm{B}\}\,. \tag{2.5}$$

We assume the enharmonic equivalences, such as $\mathrm{C}\sharp = \mathrm{D}\flat$ and so on. The pitches are then defined as:

$$\mathcal{N} = \mathcal{N}_{12} \times \mathbb{Z}\,. \tag{2.6}$$

While theoretically, we could write (N, n), where $N \in \mathcal{N}_{12}$ and $n \in \mathbb{Z}$, we use the notation [pitch][octave] instead, according to the notation suggested by the Acoustical Society of America (Young, 2005). For instance, the central C of the piano is notated as C4, and the pitch with a frequency of 440 Hz is called A4, following the ISO standard ISO, 1975.

In this case, the group $G_{\mathcal{F}}$ is set to $(\mathbb{Z}, +)$. It acts on pitches by shifting the pitch upwards (for positive shifts) or downwards (for negative shifts). For instance, $\mathrm{C4} + 1 = \mathrm{C}\sharp 4$, $\mathrm{C4} + 12 = \mathrm{C5}$ and $\mathrm{C4} - 5 = \mathrm{G3}$. Since there is a bijection[9] between $\mathcal{N}$ and $\mathbb{Z}$, we can also define the subtraction between pitches; the result is the shift needed to translate one pitch into the other. For instance, $\mathrm{C4} - \mathrm{C3} = 12$ and $\mathrm{C4} - \mathrm{E4} = -4$.

It is important to note that we can extend semitones continuously, allowing $(\mathbb{R}, +)$ to be the group for frequency shifts. We call the space of semitones $\mathcal{F}_{\mathrm{st}} = \mathbb{R}$, where st stands for semitone. This extension would grant access to finer intervals, such as quarter tones and other divisions of the octave, and also allows us to explore different temperaments beyond the equal temperament.

Lastly, let us discuss another space for representing frequencies: the chromas. As mentioned before, we denote this space as $\mathcal{N}_{12}$. It is particularly interesting from a musical perspective because many considerations, especially harmonic ones, are made up to an octave. In this case, the group $G_{\mathcal{F}}$ for the chromas would be $\mathbb{Z}_{12}$. We can also extend this representation continuously to the space of continuous chromas using $(12\mathbb{T}, +)$ as the group, where $\mathbb{T}$ represents the one dimensional torus $\mathbb{T} = {}^{\mathbb{R}}\!/_{\mathbb{Z}}$.

2.1.3 Lattice Structure for the Amplitude Range

For the musical space to be a complete lattice with pointwise order, we need the amplitude range $\mathcal{A}$ to be a complete lattice. In the following, we have a look at the different options for $\mathcal{A}$ that we will use throughout this work.

[8] We use the term *chroma* to refer to what is also called a *pitch class* in music theory. While the latter may be more mathematically accurate (as it refers to an equivalence class of pitches), the term *chroma* emphasizes the circularity of the pitch space, as proposed by Shepard, 1964.

[9] Given by the midi number, exposed in Section 2.3.2.2.

We split them into two classes: *continuous* lattices (whose cardinal is strictly superior to $\aleph_0$) and discrete ones (with cardinal less or equal to $\aleph_0$). All the lattices that we present are complete lattices.

2.1.3.1 Continuous lattices

We use the following continuous lattices:

$$([0,1],\leq) \qquad (\overline{\mathbb{R}},\leq) \qquad (\overline{\mathbb{R}}^{+},\leq) \qquad (\overline{\mathbb{R}}^{-},\leq)$$

These continuous lattices are used for spectrograms, with their specific choice depending on the transform and the unit (e.g., no unit or dB).

It is noteworthy that all of these lattices are isomorphic. In fact, we have the following increasing bijections:

$$\begin{array}{rccc} \tan: & [0,1] & \to & \overline{\mathbb{R}} \\ & x & \mapsto & \tan\left((x-\frac{1}{2})\cdot\pi\right) \end{array} \tag{2.7}$$

$$\begin{array}{rccc} 20\log_{10}: & [0,1] & \to & \overline{\mathbb{R}}^{-} \\ & x & \mapsto & 20\log_{10}(x) \end{array} \tag{2.8}$$

$$\begin{array}{rccc} 20\log_{10}: & \overline{\mathbb{R}}^{+} & \to & \overline{\mathbb{R}} \\ & x & \mapsto & 20\log_{10}(x) \end{array} . \tag{2.9}$$

These bijections preserve the order, ensuring that these lattices have the same structure.

We equip these lattices with a lattice multiplication and thus a structure of residuated lattice by using the canonical residuation presented in Definition 1.27.

For $[0,1]$, we employ the classical multiplication $\cdot$, which is an internal operation with an absorbing bottom element. Its corresponding residuation is the classical division / where the case $x/0$ is set to 1, a consequence of the definition of residuation. Indeed, $\forall x \in [0,1]$, $x/0 = \bigvee\{a \in [0,1] : a \cdot 0 \leq x\} = 1$.

Similarly, for $\overline{\mathbb{R}}^{+}$, we use multiplication with the absorbing bottom element, meaning $0 \cdot \infty = 0$. The residuation here behaves similarly to $[0,1]$, with $x/0 = \infty$.

In the case of $\overline{\mathbb{R}}$, we use the extended addition defined in Equation (1.30).

Finally, for $\overline{\mathbb{R}}^{-}$, we use the addition, which exhibits similar behavior to $\overline{\mathbb{R}}$, but with the residuation (the subtraction) resulting in:

$$-\infty - (-\infty) = \bigvee\{a \in \overline{\mathbb{R}}^{-} : a + (-\infty) \leq -\infty\} = \bigvee \overline{\mathbb{R}}^{-} = 0\,.$$

2.1.3.2 Binary and ternary lattices

The binary and ternary lattices are $\mathcal{A}_2 = \{0, 1\}$ and $\mathcal{A}_3 = \{\bot, \cdot, \times\}$, respectively. The former, that has the usual order, is called the **Boolean lattice**. The later, whose order is $\bot < \cdot < \times$, is not given in numeric form for distinguishing it from the Boolean one (as they will appear together often). It is called the **rhythmic range** because it helps us to model rhythms. Its elements are called *silence* ($\bot$), *sustain* ($\cdot$) and *onset* ($\times$). As discussed in Section 4.1.1, we will use this range to define the notion of rhythm (see Definition 4.1).

The Boolean lattice can be combined with any other lattice to form a residuated triplet by defining the lattice multiplication and left residuation[10] by:

$$\begin{array}{rccc} \bullet: & \mathcal{A}_2 \times \mathcal{A} & \to & \mathcal{A} \\ & (a,b) & \mapsto & \begin{cases} b & \text{if } a = 1 \\ \bot & \text{if } a = 0 \end{cases} \end{array} \qquad \begin{array}{rccc} /: & \mathcal{A} \times \mathcal{A} & \to & \mathcal{A}_2 \\ & (a,b) & \mapsto & \begin{cases} 1 & \text{if } a \geq b \\ 0 & \text{if } a \ngeq b \end{cases} \end{array} .$$

In particular, the combination of the Boolean lattice with the rhythmic range will be extensively used in Chapters 4 and 5.

2.1.3.3 Dynamics lattices

In music, the intensity with which a note is played is often expressed using *dynamics*. These dynamics are represented in scores by symbols such as $\boldsymbol{p}$, $\boldsymbol{mf}$, $\boldsymbol{f}$, and so on. We define the lattice of score dynamics as follows:

$$\mathcal{D}_{\boldsymbol{pf}} = \{\bot < ... < \boldsymbol{ppp} < \boldsymbol{pp} < \boldsymbol{p} < \boldsymbol{mp} < \emptyset < \boldsymbol{mf} < \boldsymbol{f} < \boldsymbol{ff} < \boldsymbol{fff} < ... < \top\} \quad (2.10)$$

where the $\emptyset$ dynamic means that there is no dynamic specified.

When working with a MIDI file, we do not have the symbolic representation of intensity (dynamics) as in traditional music scores. Instead, we use a numeric representation known as *MIDI dynamics*.

The MIDI dynamics lattice is defined as follows:

$$\mathcal{D}_{128} = (\{0, 1, ..., 127\}, \leq) . \quad (2.11)$$

Each level in this lattice indicates a specific intensity level, with 0 representing silence.

[10] See Definition 1.24 for the definitions of these concepts.

2.1.3.4 Amplitudes for piano rolls

In this section, we present the most refined range we have developed for representing MIDI files and scores as piano rolls in a musical space. This is, in fact, the first example of the application of a residuated triplet in which none of the triplet's elements are repeated.

For the sake of generality, we use $\mathcal{D}$ for either $\mathcal{D}_{\boldsymbol{pf}}$ or $\mathcal{D}_{128}$, depending on whether we want to model scores or MIDI files. We then proceed to describe two different ways of coupling $\mathcal{D}$ with $\mathcal{A}_3$:

1. the pianistic dynamics,

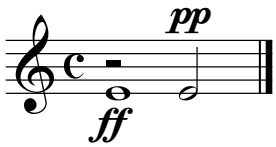
2. the sustained dynamics.

The pianistic dynamics represent the dynamics that can be performed on a piano, where once a note is hit, there is no further control over its intensity. The resulting lattice is given by:

$$\mathcal{A}_{\mathcal{D}}^{P} = \mathcal{D} \cup \{\cdot\} \tag{2.12}$$

where $\bot_{\mathcal{D}} < \cdot < d$, $\forall d \in \mathcal{D} \setminus \{\bot_{\mathcal{D}}\}$.

We claim that $\cdot$ is smaller than every other dynamic to capture the concept of piano scores. In piano notation, whenever a note appears, it should be played at the indicated dynamic level. Consider the following example:



In this example, the first note is hit with ***ff*** intensity, and the second note is hit with ***pp*** intensity while the previous one is still sustained. In this case, the last two beats would sound at ***pp***, since the last dynamic takes priority. Although this may seem unusual, it is allowed in musical staff notation.

The sustained dynamics represent the dynamics that can be performed on those music instruments that can control the intensity of a note after hitting it (like strings or winds). It is defined as

$$\mathcal{A}_{\mathcal{D}} = {}^{(\mathcal{A}_3 \times \mathcal{D})}\!/_{\sim} \tag{2.13}$$

where

$$(a,b) \sim (a',b') \Leftrightarrow \begin{cases} (a,b) = (a',b') \\ a = a' = \bot_{\mathcal{A}_3} \\ b = b' = \bot_{\mathcal{D}} \end{cases} \tag{2.14}$$

We associate to $\mathcal{A}_{\mathcal{D}}$ the pointwise order. We can also represent it by

$$\mathcal{A}_{\mathcal{D}} = \{\bot\} \cup \left(\{\cdot, \times\} \times \mathcal{D}\right). \tag{2.15}$$

This range allows in particular the execution of *crescendos* and *diminuendos*; the following excerpt

might now be expressed by the sequence of amplitudes

$$(\times,\boldsymbol{pp}), (\cdot,\boldsymbol{p}), (\cdot,\boldsymbol{mp}), (\cdot,\boldsymbol{mf}), (\cdot,\boldsymbol{f}), (\times,\boldsymbol{ff}).$$

Moreover, we are able to give a sense to the dynamic ***fp*** through $(\times,\boldsymbol{f}), (\cdot,\boldsymbol{p})$.
Whereas the pianistic dynamics are totally ordered and can be visualized as

$$\bot_{\mathcal{D}} < \ \cdot\ < \cdots < \boldsymbol{pp} < \boldsymbol{p}\ < \boldsymbol{mp} < \ \emptyset\ < \boldsymbol{mf} < \boldsymbol{f}\ < \boldsymbol{ff} < \cdots < \top_{\mathcal{D}}$$

the sustained dynamics are a partial order that is not total, and may be visualized as

$$\begin{array}{ccccccccccccccc} & & \cdots & < & (\times,\boldsymbol{p}) & < & (\times,\boldsymbol{mp}) & < & (\times,\emptyset) & < & (\times,\boldsymbol{mf}) & < & (\times,\boldsymbol{f}) & < \cdots < & (\times,\top_{\mathcal{D}}) \\ \bot & \nwarrow^{\swarrow} & & & \vee & & \vee & & \vee & & \vee & & \vee & & \vee \\ & & \cdots & < & (\cdot,\boldsymbol{p}) & < & (\cdot,\boldsymbol{mp}) & < & (\cdot,\emptyset) & < & (\cdot,\boldsymbol{mf}) & < & (\cdot,\boldsymbol{f}) & < \cdots < & (\cdot,\top_{\mathcal{D}}) \end{array}.$$

We can endow the lattice of sustained dynamics with a residuated triplet structure. Let us define the lattice multiplication

$$\begin{array}{rccc} \bullet: & \mathcal{A}_3 \times \mathcal{D} & \to & \mathcal{A}_{\mathcal{D}} \\ & (a,b) & \mapsto & [(a,b)] \end{array} \tag{2.16}$$

where $[(a, b)]$ is the equivalence class of (a, b) under the equivalence relation $\sim$.

It is trivial to see that it is a lattice multiplication, since $\forall A \subseteq \mathcal{A}_3$, $\forall b \in \mathcal{D}$,

$$\left(\bigvee A\right) \bullet b = \left(\bigvee A, b\right) = \bigvee\{(a, b) \in \mathcal{A}_\mathcal{D} : a \in A\} = \bigvee(A \bullet b)\,.$$

Its residuation is defined by

$$\begin{array}{rccl} / : & \mathcal{A}_\mathcal{D} \times \mathcal{D} & \to & \mathcal{A}_3 \\ & ((a, b), c) & \mapsto & \begin{cases} a & \text{if } b \leq c \\ \bot & \text{if } b \not\leq c \end{cases} \end{array} \qquad \begin{array}{rccl} \setminus : & \mathcal{A}_3 \times \mathcal{A}_\mathcal{D} & \to & \mathcal{D} \\ & (a, (b, c)) & \mapsto & \begin{cases} c & \text{if } a \leq b \\ \bot & \text{if } a \not\leq b \end{cases} \end{array} . \quad (2.17)$$

2.2 Representing Music with Spectrograms

In this section, we explore how to represent music using spectrograms, a widely used and valuable time-frequency representation. Spectrograms serve as a means to transform music from a signal format (e.g., encoded in `.wav` format) into an time-frequency representation, i.e., a function f within a musical space $\mathcal{M}$.

We will present two types of spectrograms and a generalization of both:

1. a spectrogram generated from the Short-time Fourier transform (STFT),

2. a spectrogram generated from the Constant-Q transform (CQT),

3. a combined approach, the Time-frequency-scale transform (TFST), which encompasses both STFT and CQT.

All of these spectrograms are Fourier-based transformations, and we will use the notations introduced in the Preamble and described in Appendix B.

The basic idea behind spectrograms is to transform a function $f : \mathcal{T} \to \mathbb{R}$, which represents a wave of musical sound, into a function $S_f : \mathcal{T} \times \mathcal{F} \to \mathcal{A}$. The choices of $\mathcal{T}$, $\mathcal{F}$, and $\mathcal{A}$ are dependent on the specific spectrogram and the nature of $\mathcal{T}$.

In the following sections, we first define the continuous transformations from which spectrograms can be extracted in Section 2.2.1. Next, we adapt these continuous transformations to the discrete framework in Section 2.2.2, which is used for actual computations. Finally, in Section 2.2.3, we show how to compute a spectrogram using these transformations.

2.2.1 Continuous Definitions

2.2.1.1 Short-time Fourier transform

The Short-time Fourier transform (STFT) is a widely used operator in signal analysis and processing, particularly in audio signals. We use the definitions from Gröchenig, 2001 and the notations presented in the Preamble and the Appendix B.

Definition 2.2 (Short-time Fourier transform)**.** Let $f \in \mathcal{L}_\infty(\mathbb{R};\mathbb{C})$ and let $g \in \mathcal{L}_1(\mathbb{R};\mathbb{C})$. The **Short-time Fourier transform** of f with respect to the window g is defined by

$$\begin{array}{rccl} \mathrm{STFT}_g[f]: & \mathbb{R}\times\mathbb{R} & \to & \mathbb{C} \\ & (\tau,\omega) & \mapsto & \int_{\mathbb{R}} f(t)\overline{g(t-\tau)}e^{-2\pi\imath t\omega}\,dt \end{array} . \tag{2.18}$$

Some properties of the STFT are stated in the next proposition.

Proposition 2.3. *Let $g \in \mathcal{L}_1(\mathbb{R};\mathbb{C})$. Then, the operator*

$$\begin{array}{rccl} \mathrm{STFT}_g: & \mathcal{L}_\infty(\mathbb{R};\mathbb{C}) & \to & \mathcal{L}_\infty(\mathbb{R}\times\mathbb{R};\mathbb{C}) \\ & f & \mapsto & \mathrm{STFT}_g[f] \end{array} \tag{2.19}$$

is linear and

$$||\mathrm{STFT}_g[f]||_\infty \leq ||f||_\infty ||g||_1 \tag{2.20}$$

which makes it a continuous operator from $\mathcal{L}_\infty(\mathbb{R};\mathbb{C})$ to $\mathcal{L}_\infty(\mathbb{R}\times\mathbb{R};\mathbb{C})$.

In particular, if $||g||_1 = 1$,

$$||\mathrm{STFT}_g[f]||_\infty \leq ||f||_\infty . \tag{2.21}$$

If we limit ourselves to the functions in $\mathcal{L}_2(\mathbb{R};\mathbb{C})$, we can express the STFT using the translation and modulation operators, defined as $T_\tau f(t) = f(t-\tau)$ and $M_\omega f(t) = f(t)\cdot e^{2\pi\imath t\omega}$, respectively (detailed in Appendix B).

Proposition 2.4. *Let $f, g \in \mathcal{L}_2(\mathbb{R};\mathbb{C})$. Then, $\forall(\tau,\omega)\in\mathbb{R}\times\mathbb{R}$,*

$$\mathrm{STFT}_g[f](\tau,\omega) = \langle f, M_\omega T_\tau g\rangle . \tag{2.22}$$

An important property, called the *fundamental identity of time-frequency analysis* in (Gröchenig, 2001), that we will use in Chapter 3, is the following.

Proposition 2.5 (Fundamental identity of time-frequency analysis)**.** *Let $f, g \in \mathcal{L}_2(\mathbb{R};\mathbb{C})$. Then, $\forall(\tau,\omega)\in\mathbb{R}\times\mathbb{R}$,*

$$\mathrm{STFT}_g[f](\tau,\omega) = e^{-2\pi\imath\omega t}\mathrm{STFT}_{\mathscr{F}[g]}[\mathscr{F}[f]](\omega,-\tau) . \tag{2.23}$$

Proof. Since this property is not proved in (Gröchenig, 2001) we include its proof here:

$$\begin{aligned}\mathrm{STFT}_g[f](\tau,\omega) &\overset{\text{Proposition 2.4}}{=} \langle f, M_\omega T_\tau g\rangle \\ &= \langle \mathscr{F}[f], \mathscr{F}[M_\omega T_\tau g]\rangle \\ &= \langle \mathscr{F}[f], T_\omega M_{-\tau}\mathscr{F}[g]\rangle \\ &= \langle f, e^{2\pi\imath\tau\omega} M_{-\tau} T_\omega g\rangle \\ &= e^{-2\pi\imath\tau\omega}\langle f, M_{-\tau}T_\omega g\rangle \\ &= e^{-2\pi\imath\tau\omega}\mathrm{STFT}_{\mathscr{F}[g]}[\mathscr{F}[f]](\omega,-\tau)\end{aligned}$$

□

To finish with a brief introduction to the STFT, let us recall the inversion formula.

Proposition 2.6 (Inversion formula for the STFT)**.** *Let* $g, \gamma \in \mathcal{L}_2(\mathbb{R};\mathbb{C})$ *such that* $\langle g,\gamma\rangle \neq 0$*. Then,* $\forall f \in \mathcal{L}_2(\mathbb{R};\mathbb{C})$

$$f = \frac{1}{\langle \gamma, g\rangle}\int_{\mathbb{R}}\int_{\mathbb{R}} \mathrm{STFT}_g[f](\tau,\omega) M_\omega T_\tau \gamma \, d\omega \, d\tau \,. \tag{2.24}$$

This leads us to the definition of the inverse STFT.

Definition 2.7 (Inverse short-time Fourier transform)**.** Let $g, \gamma \in \mathcal{L}_2(\mathbb{R};\mathbb{C})$ such that $\langle g,\gamma\rangle \neq 0$. Then, $\forall S \in \mathcal{L}_2(\mathbb{R}\times\mathbb{R};\mathbb{C})$ the **inverse short-time Fourier transform** of S, denoted by $\mathrm{iSTFT}[S]$, is given by:

$\forall t \in \mathbb{R}$,

$$\mathrm{iSTFT}[S](t) = \frac{1}{\langle \gamma, g\rangle}\int_{\mathbb{R}}\int_{\mathbb{R}} S(\tau,\omega) M_\omega T_\tau \gamma(t) \, d\omega \, d\tau \,. \tag{2.25}$$

2.2.1.2 Constant-Q transform

The Constant-Q transform (CQT) is an operator that is well-suited for music representation due to its logarithmic frequency resolution, differing from STFT, which has a linear frequency resolution. This characteristic is particularly important for music since the musical tuning is based on the concept of octave intervals, which is a logarithmic feature.

The CQT was initially proposed by Youngberg and Boll, 1978, and later an efficient method for its computation was introduced (Brown, 1991). As mentioned in (Schörkhuber & Klapuri, 2010), the CQT can be viewed as a particular case of a wavelet transform.

Definition 2.8 (Continuous wavelet transform)**.** Let $\psi \in \mathcal{L}_2(\mathbb{R};\mathbb{C})$ such that $\int_{\mathbb{R}} \psi(t)\,dt = 0$. Then, the **continuous wavelet transform** (CWT) is defined by

$$\begin{array}{rccl} W: & \mathcal{L}_2(\mathbb{R};\mathbb{C}) & \to & \mathcal{L}_\infty(\mathbb{R}\times\mathbb{R}^{+*};\mathbb{C}) \\ & f & \mapsto & W[f]: \mathbb{R}\times\mathbb{R}^{+*} \to \mathbb{C} \\ & & & (\tau,\sigma) \mapsto \frac{1}{\sqrt{\sigma}}\int_{\mathbb{R}} f(t)\overline{\psi(\frac{t-\tau}{\sigma})}\,dt \end{array} . \tag{2.26}$$

We can express the continuous wavelet transform in terms of the scalar product. To do that, we recall that the 2-unitary dilation operator is $D^2_\sigma f(t) = \frac{1}{\sqrt{\sigma}} f\left(\frac{t}{\sigma}\right)$ (detailed in Appendix B).

Proposition 2.9. *Let $f, \psi \in \mathcal{L}_2(\mathbb{R};\mathbb{C})$. Then, $\forall(\tau,\sigma) \in \mathbb{R}\times\mathbb{R}^{+*}$,*

$$W_\psi[f](\tau,\sigma) = \langle f, T_\tau D^2_\sigma \psi\rangle \tag{2.27}$$

and

$$||W_\psi[f]||_\infty \leq ||f||_2 \cdot ||\psi||_2\,. \tag{2.28}$$

Proof. $\forall(\tau,\sigma) \in \mathbb{R}\times\mathbb{R}^{+*}$,

$$\begin{aligned} \langle f, T_\tau D^2_\sigma \psi\rangle &= \int_{\mathbb{R}} f(t)\cdot\overline{T_\tau D^2_\sigma \psi(t)}\,dt \\ &= \int_{\mathbb{R}} f(t)\cdot\overline{\frac{1}{\sqrt{\sigma}}\psi\left(\frac{t-\tau}{\sigma}\right)}\,dt \\ &= \frac{1}{\sqrt{\sigma}}\int_{\mathbb{R}} f(t)\cdot\overline{\psi\left(\frac{t-\tau}{\sigma}\right)}\,dt \\ &= W_\psi[f](\tau,\sigma)\,. \end{aligned}$$

Then, $\forall(\tau,\sigma) \in \mathbb{R}\times\mathbb{R}^{+*}$,

$$\begin{aligned} |W_\psi[f](\tau,\sigma)| &= |\langle f, T_\tau D^2_\sigma\psi\rangle| \\ &\leq ||f||_2 \cdot ||T_\tau D^2_\sigma\psi||_2 \\ &= ||f||_2 \cdot ||\psi||_2 \end{aligned}$$

which implies

$$||W_\psi[f](\tau,\sigma)||_\infty \leq ||f||_2 \cdot ||\psi||_2\,.$$

□

The Constant-Q transform can now be expressed as a particular case of a wavelet transform, where $\psi(t) = g(t)e^{2\pi\imath t}$, with g being a window function. However, we change the 2-unitary dilation to the 1-unitary dilation (presented in Appendix B), in order to obtain an inequality involving the infinity norm, as shown in Equation (2.20).

Definition 2.10 (Constant-Q transform)**.** Let $f \in \mathcal{L}_\infty(\mathbb{R};\mathbb{C})$. Let $g \in \mathcal{L}_1(\mathbb{R};\mathbb{C})$. We define the **Constant-Q transform** of f with the window g as

$$\begin{array}{rccl} \mathrm{CQT}_g[f]: & \mathbb{R}\times\mathbb{R}^{+*} & \to & \mathbb{C} \\ & (\tau,\sigma) & \mapsto & \frac{1}{\sigma}\int_{\mathbb{R}} f(t)\cdot\overline{g(\frac{t-\tau}{\sigma})}e^{-2\pi\imath\frac{t}{\sigma}}\,dt \end{array} \tag{2.29}$$

While the CQT is a time-scale transform, we can interpret it as a time-frequency transform with $\xi = \frac{1}{\sigma}$.

We can express the Constant-Q transform as a scalar product.

Proposition 2.11. *Let $f \in \mathcal{L}_\infty(\mathbb{R};\mathbb{C})$. Let $g \in \mathcal{L}_1(\mathbb{R};\mathbb{C})$. Then, $\forall(\tau,\sigma)\in\mathbb{R}\times\mathbb{R}^{+*}$,*

$$\mathrm{CQT}_g[f](\tau,\sigma) = \langle f, M_{\frac{1}{\sigma}}T_\tau D^1_\sigma g\rangle \tag{2.30}$$

and

$$||\mathrm{CQT}_g[f](\tau,\sigma)||_\infty \leq ||f||_\infty \cdot ||g||_1\,. \tag{2.31}$$

Proof. $\forall(\tau,\sigma)\in\mathbb{R}\times\mathbb{R}^{+*}$,

$$\begin{aligned} \langle f, M_{\frac{1}{\sigma}}T_\tau D^1_\sigma g\rangle &= \int_{\mathbb{R}} f(t)\cdot\overline{M_{\frac{1}{\sigma}}T_\tau D^1_\sigma g(t)}\,dt \\ &= \int_{\mathbb{R}} f(t)\cdot\overline{e^{2\pi\imath t\frac{1}{\sigma}}\frac{1}{\sigma}g\big(\frac{t-\tau}{\sigma}\big)}\,dt \\ &= \frac{1}{\sigma}\int_{\mathbb{R}} f(t)\cdot\overline{g\big(\frac{t-\tau}{\sigma}\big)}e^{-2\pi\imath\frac{t}{\sigma}}\,dt \\ &= \mathrm{CQT}_g[f](\tau,\sigma)\,. \end{aligned}$$

Then, $\forall(\tau,\sigma)\in\mathbb{R}\times\mathbb{R}^{+*}$,

$$\begin{aligned} |\mathrm{CQT}_g[f](\tau,\sigma)| &= |\langle f, M_{\frac{1}{\sigma}}T_\tau D^1_\sigma g\rangle| \\ &\leq ||f||_\infty \cdot ||M_{\frac{1}{\sigma}}T_\tau D^1_\sigma g||_1 \\ &= ||f||_\infty \cdot ||g||_1 \end{aligned}$$

which implies

$$||\mathrm{CQT}_g[f](\tau,\sigma)||_\infty \leq ||f||_\infty \cdot ||g||_1\,.$$

□

2.2.1.3 Time-frequency-scale transform

By using the scalar product formulation of the STFT and the CQT, we can observe their similarity. This similarity can be summarized as a three-dimensional transformation that we call *Time-frequency-scale transform* (TFST). The TFST combines the time-frequency approach of the STFT with the time-scale approach of the CQT, resulting in a three-dimensional representation.

While the TFST is not a time-frequency representation, it fits perfectly within the postulates of this work, where we need a space (in this case three-dimensional) and a group acting on it. The group would be $(\mathbb{R},+)\times(\mathbb{R},+)\times(\mathbb{R}^{+*},\cdot)$.

This transformation is implicitly present in deep-learning applications under different names such as *multi-scale spectral loss* (Engel et al., 2020) or *multi-resolution spectral distance* (Wang et al., 2020). In these cases, an intermediate TFST is computed and then contracted into a single scalar value, which is used as a loss function for gradient descent.

Previous works have explored similar directions (Levine et al., 1998; Bonada, 2000; Bonada, 2002; Dorran, 2005; Juillerat et al., 2008; Mateo & Talavera, 2020), proposing two-dimensional representations with varying time-frequency resolutions instead of using the third dimension.

The TFST can be viewed as a way to overcome the time-frequency uncertainty principle at the cost of introducing an extra dimension in the representation.

Definition 2.12 (Time-frequency-scale transform)**.** Let $g\in\mathcal{L}_1(\mathbb{R};\mathbb{C})$. We define the **Time-frequency-scale transform** with window g as

$$\begin{array}{rccl} \mathrm{TFST}_g: & \mathcal{L}_\infty(\mathbb{R};\mathbb{C}) & \to & \mathcal{L}_\infty(\mathbb{R}\times\mathbb{R}\times\mathbb{R}^{+*};\mathbb{C}) \\ & f & \mapsto & \mathrm{TFST}_g[f]: \quad \begin{array}{ccl} \mathbb{R} & \to & \mathbb{C} \\ (\tau,\omega,\sigma) & \mapsto & \langle f, T_\tau M_\omega D^1_\sigma g\rangle \end{array} \end{array} . \tag{2.32}$$

We can establish an inequality for the TFST, similar to those of the STFT and the CQT. This inequality implies that the TFST is a continuous operator between $\mathcal{L}_\infty(\mathbb{R};\mathbb{C})$ and $\mathcal{L}_\infty(\mathbb{R}\times\mathbb{R}\times\mathbb{R}^{+*};\mathbb{C})$.

Proposition 2.13. *Let $f\in\mathcal{L}_\infty(\mathbb{R};\mathbb{C})$. Let $g\in\mathcal{L}_1(\mathbb{R};\mathbb{C})$. Then,*

$$||\mathrm{TFST}_g[f]||_\infty \le ||f||_\infty\cdot||g||_1\,. \tag{2.33}$$

Proof. $\forall(\tau,\omega,\sigma)\in\mathbb{R}\times\mathbb{R}\times\mathbb{R}^{+*}$,

$$\begin{aligned} |\mathrm{TFST}_g[f](\tau,\omega,\sigma)| &= |\langle f, T_\tau M_\omega D^1_\sigma g\rangle| \\ &\le ||f||_\infty\cdot||T_\tau M_\omega D^1_\sigma g||_1 \\ &= ||f||_\infty\cdot||g||_1 \end{aligned}$$

then

$$||\mathrm{TFST}_g[f]||_\infty \leq ||f||_\infty \cdot ||g||_1$$

□

As mentioned, we can express the STFT and the CQT as particular cases of the TFST.

Proposition 2.14. *Let $f \in \mathcal{L}_\infty(\mathbb{R};\mathbb{C})$. Let $g \in \mathcal{L}_1(\mathbb{R};\mathbb{C})$. Then, $\forall(\tau,\omega) \in \mathbb{R}\times\mathbb{R}$,*

$$\mathrm{STFT}_g[f](\tau,\omega) = e^{-2\pi\imath\omega\tau}\mathrm{TFST}_g[f](\tau,\omega,1) \tag{2.34}$$

and $\forall(\tau,\sigma) \in \mathbb{R}\times\mathbb{R}^{+*}$,

$$\mathrm{CQT}_g[f](\tau,\sigma) = e^{-2\pi\imath\frac{\tau}{\sigma}}\mathrm{TFST}_g[f](\tau,\frac{1}{\sigma},\sigma)\,. \tag{2.35}$$

Proof. $\forall(\tau,\omega) \in \mathbb{R}\times\mathbb{R}$,

$$\begin{aligned}\mathrm{STFT}_g[f](\tau,\omega) &= \langle f, M_\omega T_\tau g\rangle \\ &= \langle f, e^{2\pi\imath\tau\omega}T_\tau M_\omega g\rangle \\ &= e^{-2\pi\imath\tau\omega}\langle f, T_\tau M_\omega D_1^1 g\rangle \\ &= e^{-2\pi\imath\omega\tau}\mathrm{TFST}_g[f](\tau,\omega,1)\end{aligned}$$

$\forall(\tau,\sigma) \in \mathbb{R}\times\mathbb{R}^{+*}$,

$$\begin{aligned}\mathrm{CQT}_g[f](\tau,\sigma) &= \langle f, M_{\frac{1}{\sigma}}T_\tau D_\sigma^1 g\rangle \\ &= \langle f, e^{2\pi\imath\tau\frac{1}{\sigma}}T_\tau M_{\frac{1}{\sigma}} D_\sigma^1 g\rangle \\ &= e^{-2\pi\imath\tau\frac{1}{\sigma}}\langle f, T_\tau M_{\frac{1}{\sigma}} D_\sigma^1 g\rangle \\ &= e^{-2\pi\imath\frac{\tau}{\sigma}}\mathrm{TFST}_g[f](\tau,\frac{1}{\sigma},\sigma)\end{aligned}$$

□

In addition, we can also see express the TFST as a convolution.

Proposition 2.15. *Let $f \in \mathcal{L}_\infty(\mathbb{R};\mathbb{C})$. Let $g \in \mathcal{L}_1(\mathbb{R};\mathbb{R})$. Then, $\forall(\tau,\omega,\sigma) \in \mathbb{R}\times\mathbb{R}\times\mathbb{R}^{+*}$,*

$$\mathrm{TFST}_g[f](\tau,\omega,\sigma) = (f * M_\omega D_\sigma^1 g^*)(\tau) \tag{2.36}$$

Proof. $\forall(\tau,\omega,\sigma)\in\mathbb{R}\times\mathbb{R}\times\mathbb{R}^{+*}$,

$$\begin{aligned}(f * M_\omega D^1_\sigma g^*)(\tau) &= \langle f, T_\tau (M_\omega D^1_\sigma g^*)^*\rangle \\ &= \langle f, T_\tau M_\omega D^1_\sigma (g^*)^*\rangle \\ &= \langle f, T_\tau M_\omega D^1_\sigma g\rangle \\ &= \mathrm{TFST}_g[f](\tau,\omega,\sigma)\,.\end{aligned}$$

□

Corollary 2.16. *Let $f\in\mathcal{L}_\infty(\mathbb{R};\mathbb{C})$. Let $g\in\mathcal{L}_1(\mathbb{R};\mathbb{R})$. Then, $\forall(\tau,\omega)\in\mathbb{R}\times\mathbb{R}$,*

$$\mathrm{STFT}_g[f](\tau,\omega) = e^{-2\pi\imath\tau\omega}(f * M_\omega g^*)(\tau) \tag{2.37}$$

and, $\forall(\tau,\sigma)\in\mathbb{R}\times\mathbb{R}^{+*}$,

$$\mathrm{CQT}_g[f](\tau,\sigma) = e^{-2\pi\imath\frac{\tau}{\sigma}}(f * M_{\frac{1}{\sigma}} D^1_\sigma g^*)(\tau)\,. \tag{2.38}$$

Representing the STFT, CQT, and TFST as convolutions proves to be advantageous for computational purposes. While the traditional approach of using Fast Fourier Transform (FFT) (Cooley & Tukey, 1965) is usually preferred due to its logarithmic complexity, modern GPU computations have shown superior performance by using highly parallelizable operations like convolutions (Cheuk et al., 2020).

2.2.2 Discrete Definitions

To perform actual computations, we need to transition from the continuous representation to the discrete one. We consider an input time series $\mathbf{f}=(\mathbf{f}[n])_{n=0}^{N-1}$ of size $N\in\mathbb{N}^*$ with a sampling frequency $\xi_s\in\mathbb{R}$, expressed in Hz.

2.2.2.1 Discrete STFT

The discrete STFT is then defined as follows.

Definition 2.17 (Discrete STFT)**.** Let $N,J\in\mathbb{N}^*$. Let $\mathbf{f}=(\mathbf{f}[n])_{n=0}^{N-1}\in\mathbb{C}^N$, let $\mathbf{g}=(\mathbf{g}[j])_{j=0}^{J-1}\in\mathbb{C}^J$. Let $j_0\in\{0,1,...,J-1\}$ be the index corresponding[11] to the

[11]This parameter, often overlooked in the literature by assuming it to be 0 or $M/2$, is actually essential for achieving good compatibility between the continuous and discrete STFT. This aspect is related to what has been discussed in Chapter 1 regarding the distinction between a set of points E and a group of shifts $(G,+)$; in this case, the indices of $\mathbf{f}$ and $\mathrm{STFT}[\mathbf{f}]$ correspond to points, while the indices of $\mathbf{g}$ correspond to time shifts.

element of $\mathbf{g}$ that acts as 0 of the group $(\mathbb{R}, +)$. The **discrete STFT** of $\mathbf{f}$ with window $\mathbf{g}$ and with $K \in \mathbb{N}^*$ frequency bins is defined by

$\forall m \in \{0, 1, ..., N-1\}$, $\forall k \in \{0, 1, ..., K-1\}$,

$$\mathrm{STFT}_{\mathbf{g}}[\mathbf{f}][m,k] = \sum_{n=m-j_0}^{m-j_0+J-1} \mathbf{f}[n]\overline{\mathbf{g}[n-m+j_0]}e^{-2\pi\imath n\frac{k}{K}} \tag{2.39}$$

$$\overset{j=n-m+j_0}{=} \sum_{j=0}^{J-1} \mathbf{f}[m+j-j_0]\overline{\mathbf{g}[j]}e^{-2\pi\imath(m+j-j_0)\frac{k}{K}}\,, \tag{2.40}$$

where we assume that $\mathbf{f}[n] = 0$, $\forall n \notin \{0, 1, ..., N-1\}$.

The discretization points of the STFT are $t_m = \frac{m}{\xi_s}$, and $\xi_k = k\xi_s$ with $\xi_s \in \mathbb{R}$ being the sampling frequency.

We can also introduce another parameter $H \in \mathbb{N}^*$, known as the *hop size*, and subsample the formula by considering $\mathrm{STFT}_{\mathbf{g}}[\mathbf{f}][mH, k]$.

This definition may differ slightly from other common definitions, especially regarding the parameter j_0. In many cases, there is no specific attention given to the computational problem of aligning the window with the signal. However, since windows are often concentrated in the center bin, we aim to align this bin (indexed by j_0) with the $\mathbf{f}[m]$ value. Additionally, we want the oscillatory factor to have the value 1 at this point. This consideration also resolves the problem of how to pad the signal, i.e., by adding j_0 zeroes at the beginning and $J-1-j_0$ at the end.

2.2.2.2 Discrete CQT

The discretization of the CQT that we use is based on (Schörkhuber & Klapuri, 2010). However, there are some modifications in our approach. The support of the window function is defined as $t \in [-\frac{1}{2}, \frac{1}{2}]$ in our case, whereas Schörkhuber and Klapuri, 2010 use $t \in [0, 1]$. Additionally, we center the complex exponential at $t = 0$, whereas they center it at $t = \frac{1}{2}$. Moreover, we allow g to be a complex-valued function, even though for practical cases we will often use a real-valued window function.

Definition 2.18 (Discrete CQT)**.** Let $\mathbf{f} = (\mathbf{f}[n])_{n=0}^{N-1} \in \mathbb{C}^N$. Let $g \in \mathcal{C}^\infty(\mathbb{R}; \mathbb{C})$ with $\mathrm{supp}(g) = [-\frac{1}{2}, \frac{1}{2}]$. Let $K \in \mathbb{N}^*$. The CQT of $\mathbf{f}$ with the window[12] g is defined by

[12]Notice that the window function cannot be a discrete function, as it needs to be sampled differently depending on the desired window size.

$\forall m \in \{0, 1, ..., N-1\}$, $\forall k \in \{0, 1, ..., K-1\}$,

$$\mathrm{CQT}_g[\mathbf{f}][m,k] = \frac{1}{N_k} \sum_{n=m-\lfloor \frac{N_k}{2} \rfloor}^{m+\lfloor \frac{N_k}{2} \rfloor} \mathbf{f}[n] \overline{g\left(\frac{n-m}{N_k}\right)} e^{-2\pi\imath n \frac{\xi_k}{\xi_s}} \tag{2.41}$$

$$= \frac{1}{N_k} \sum_{j=0}^{2\lfloor \frac{N_k}{2} \rfloor} \mathbf{f}\left[j+m-\lfloor \frac{N_k}{2} \rfloor\right] \overline{g\left(\frac{j-\lfloor \frac{N_k}{2} \rfloor}{N_k}\right)} e^{-2\pi\imath(m+j-\lfloor \frac{N_k}{2} \rfloor)\frac{\xi_k}{\xi_s}} \tag{2.42}$$

where $t_m = \frac{m}{\xi_s}$ are the discretization points of the time, $\xi_k = \xi_0 2^{\frac{k}{B}}$ the discretization points of the frequency, and $N_k = \left\lfloor \frac{q\xi_s}{\xi_k(2^{\frac{1}{B}}-1)} \right\rfloor$, with the following parameters:

- $\xi_s \in (0, \infty)$: the sampling frequency,
- $\xi_0 \in (0, \xi_s)$: the lowest frequency,
- $B \in \mathbb{N}^*$: the number of bins per octave,
- $q \in (0, 1)$: the scaling factor (inverse of the oversampling), typically equal 1.

With the given parameters, the resulting quality factors Q_k for each band[13] are expressed as:

$$\begin{aligned} Q_k &:= \frac{\xi_k}{\Delta\xi_k} \\ &= \frac{N_k \xi_k}{\Delta\omega \xi_s} \\ &\approx \frac{q}{\Delta\omega(2^{\frac{1}{B}}-1)} := Q \end{aligned}$$

where $\Delta\xi_k$ denotes the $-3\,\mathrm{dB}$ bandwidth of the frequency response of the time-frequency atom:

$$(a_k[j])_{j=0}^{N_k} := \left(\frac{1}{N_k} g\left(\frac{j}{N_k} - \frac{1}{2}\right) e^{2\pi\imath j \frac{\xi_k}{\xi_s}}\right)_{j=0}^{N_k} \tag{2.43}$$

[13]The quality factor is supposed to be equal for each band, but the rounding of N_k introduces a slight variation that is expressed by the $\approx$ symbol.

Additionally, $\Delta\omega$ is the $-3\,\mathrm{dB}$ bandwidth of the main lobe of the spectrum of the window function g, which is approximately equal to 1.50 frequency bins for the Hann window, for example.

It may be interesting to outline the relation between the dilation factor σ from the continuous CQT and all these parameters. From the frequency component, we derive that

$$\frac{1}{\sigma_k} = \frac{\xi_k}{\xi_s} \tag{2.44}$$

where $(\sigma_k)_{k=0}^{K-1}$ is the sampling of the variable σ from the continuous dilation space $\mathbb{R}^{+*}$. From that, we obtain the relation between N_k and σ_k, that is

$$N_k = \frac{q}{2^{\frac{1}{B}} - 1}\sigma_k = \lambda\sigma_k \tag{2.45}$$

with $\lambda = \frac{q}{2^{\frac{1}{B}}-1}$.

This relation implies a relation between the window function from the continuous case (denoted by g_c) and the window function from the discrete case (denoted by g_d); by identification from Equations (2.29) and (2.41), we have that

$\forall k \in \{0, 1, ..., K\}$, $\forall t \in \mathbb{R}$,

$$\frac{1}{\sigma_k} g_c\left(\frac{t}{\sigma_k}\right) = \frac{1}{N_k} g_d\left(\frac{t}{N_k}\right) = \frac{1}{\lambda\sigma_k} g_d\left(\frac{t}{\lambda\sigma_k}\right)$$

and hence

$$\begin{aligned} & g_c\left(\frac{t}{\sigma_k}\right) = \frac{1}{\lambda} g_d\left(\frac{t}{\lambda\sigma_k}\right) \\ \overset{t' := \frac{t}{\sigma_k}}{\Leftrightarrow} \quad & g_c(t') = \frac{1}{\lambda} g_d\left(\frac{t'}{\lambda}\right) \\ \Leftrightarrow \quad & g_c = D_\lambda^1 g_d\,. \end{aligned}$$

2.2.2.3 Discrete TFST

The TFST requires the discretization of three variables: time, frequency, and scale. We present its discrete version based on the convolution formula (Equation (2.36)).

Definition 2.19 (Discrete TFST)**.** Let $\mathbf{f} = (\mathbf{f}[n])_{n=0}^{N-1} \in \mathbb{C}^N$. Let $g \in \mathcal{C}^\infty(\mathbb{R}; \mathbb{C})$ with $\mathrm{supp}(g) = [-\frac{1}{2}, \frac{1}{2}]$. Let $K, L \in \mathbb{N}^*$. The TFST of $\mathbf{f}$ with the window g is defined by

$\forall m \in \{0, 1, ..., N-1\}$, $\forall k \in \{0, 1, ..., K-1\}$, $\forall l \in \{0, 1, ..., L-1\}$,

$$\mathrm{TFST}_g[\mathbf{f}][m,k,l] = \frac{1}{N_l} \sum_{n=m-\lfloor \frac{N_l}{2} \rfloor}^{m+\lfloor \frac{N_l}{2} \rfloor} \mathbf{f}[n] \overline{g\left(\frac{n-m}{N_l}\right)} e^{-2\pi\imath(n-m)\frac{\xi_k}{\xi_s}} \tag{2.46}$$

$$= \frac{1}{N_l} \sum_{j=0}^{2\lfloor \frac{N_l}{2} \rfloor} \mathbf{f}\left[j + m - \lfloor \frac{N_l}{2} \rfloor\right] \overline{g\left(\frac{j - \lfloor \frac{N_l}{2} \rfloor}{N_l}\right)} e^{-2\pi\imath(j - \lfloor \frac{N_k}{2} \rfloor)\frac{\xi_k}{\xi_s}} \tag{2.47}$$

where $\xi_s \in \mathbb{R}^{+*}$ is the sampling frequency, and $(\xi_k)_{k=0}^{K-1}$ and $(N_l)_{l=0}^{L-1}$ represent the discretization of the frequency and scale variables, respectively. The discretization points of the time are given by $t_m = \frac{m}{\xi_s}$.

In this transform, we have the flexibility to choose the level of discretization for the frequency and scale variables according to our needs. For instance, if we set a fixed value $N_l = N$ for all $l \in \{0, 1, ..., L-1\}$ and vary the frequency linearly as $\xi_k = \frac{k}{K}\xi_s$ for $k \in \{0, 1, ..., K-1\}$, we recover the STFT[14]. On the other hand, if we establish an inverse relationship between the scale and frequency, we obtain the CQT[14]. The time variable is always discretized linearly, as it is directly linked to the variable of $\mathbf{f}$, but we can subsample it using a hop size $H \in \mathbb{N}^*$, as mentioned earlier.

For computing the STFT and the CQT, we use the Python/PyTorch library `nnAudio` (Cheuk et al., 2020). This library is highly efficient, especially for the CQT, as it leverages GPU acceleration and performs computations using convolutions. As there is no standardized theory for the TFST, we have developed a custom implementation based on the principles of `nnAudio`.

2.2.3 Spectrograms

Up to this point, we have been working with transformations that produce complex-valued functions. This complex representation is useful in signal processing, as it makes the operators linear, but is inappropriate to our needs where we need the amplitude $\mathcal{A}$ to be a complete lattice[15].

To address this limitation, we adopt a common approach used in this kind of analysis: dropping the phase information and keeping only the modulus. This results in a representation called a *spectrogram*, which is the square modulus of the

[14] Up to a phase factor, as in the continuous case (exposed in Proposition 2.14).

[15] Complex numbers can be endowed with a partial order, for instance $\forall z, \omega \in \mathbb{C}$, $z \preceq \omega \Leftrightarrow |z| \leq |\omega|$), but it does not fit our purpose.

original transformation. Using the square instead of the modulus itself is a standard practice, often referred to as the *power* spectrogram due to its relations with power in electronics and acoustics.

For each transformation, we define a corresponding spectrogram by taking the square modulus of its complex value:

$$\mathrm{SPEC}_g^{\mathrm{STFT}} = |\mathrm{STFT}_g|^2, \quad \mathrm{SPEC}_g^{\mathrm{CQT}} = |\mathrm{CQT}_g|^2, \quad \mathrm{SPEC}_g^{\mathrm{TFST}} = |\mathrm{TFST}_g|^2 .$$

After taking the square modulus of the complex values, our transforms are no longer linear, and the codomain has changed from $\mathbb{C}$ to $\mathbb{R}^+$. As a result, we can now consider the usual order in $\mathbb{R}^+$ and set our amplitude as $\mathcal{A} = (\mathbb{R}^+, \leq)$.

It is a common practice to represent the amplitudes of a spectrogram in a logarithmic scale, particularly in decibels (dB). For an output value $z \in \mathbb{C}$, the equation is given by:

$$|z|^2 = 10 \log_{10} |\mathrm{z}|^2 \,\mathrm{dB} = 20 \log_{10} |\mathrm{z}| \,\mathrm{dB} . \tag{2.48}$$

As mentioned in Equation (2.9), the transformation of a positive value into decibels is an isomorphism between ordered sets.

Furthermore, it transforms products into sums. This is an interesting feature for our considerations in Mathematical Morphology, as the lattice multiplication we consider for greyscale morphology is either $\cdot$ or $+$, depending on whether we are working with a linear or logarithmic scale in the amplitude range.

Finally, to further restrict our amplitude range, we use Equations (2.20), (2.31) and (2.33). We consider an input function f with $||f||_\infty = 1$, which is equivalent to $\forall t \in \mathbb{R}, -1 \leq f(t) \leq 1$, a common condition for audio signals. Additionally, we choose the window function g with $||g||_1 = 1$, which is a normalization condition that can be applied to every window. With these conditions, we have the following properties.

Proposition 2.20. *Let $f \in \mathcal{L}_\infty(\mathbb{R}; \mathbb{C})$ with $||f||_\infty = 1$. Let $g \in \mathcal{L}_1(\mathbb{R}; \mathbb{C})$ with $||g||_1 = 1$. Then,*

$$||\mathrm{SPEC}_g^{\mathrm{STFT}}[f]||_\infty \leq 1, \quad ||\mathrm{SPEC}_g^{\mathrm{CQT}}[f]||_\infty \leq 1 \text{ and } \quad ||\mathrm{SPEC}_g^{\mathrm{TFST}}[f]||_\infty \leq 1 .$$

Proof. Since the STFT and the CQT are particular cases of the TFST with a complex exponential factor with modulus 1, it is enough to prove the result for the TFST.

$\forall (\tau, \omega, \sigma) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{+*}$,

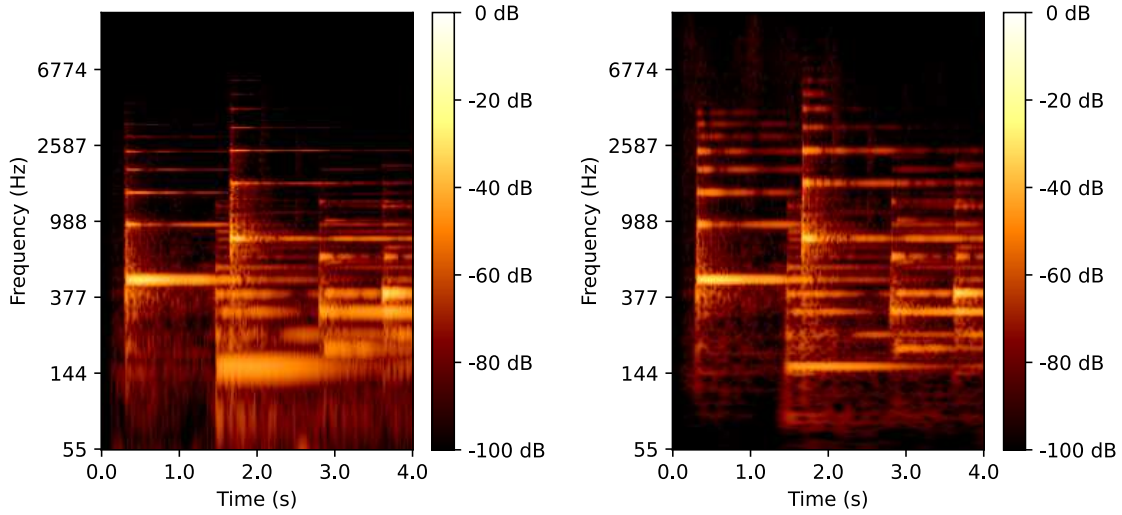
$$\mathrm{SPEC}_g^{\mathrm{TFST}}[f](\tau, \omega, \sigma) = |\mathrm{TFST}_g[f](\tau, \omega, \sigma)|^2 \overset{\textit{Equation } (2.33)}{\leq} \left(||f||_\infty \cdot ||g||_1\right)^2 = 1$$

which implies that $||\mathrm{SPEC}_g^{\mathrm{TFST}}[f]||_\infty \leq 1$. □

The fact that the spectrograms are bounded by 1 implies that when they are expressed in decibels, the amplitude range is $\overline{\mathbb{R}}^{-} = [-\infty, 0]$. This will be the amplitude range we use for spectrograms in the following.

In Figure 2.2, we present two spectrograms, one from STFT and another from CQT. Both spectrograms are displayed in logarithmic scale for both frequency and amplitude.

Additionally, Figure 2.3 illustrates a spectrogram generated by TFST with a logarithmic scale for both frequency and amplitude. Three different scale values are shown: $N_l = 1024, 4096, 46384$. This depiction clearly demonstrates that the TFST effectively addresses the time-frequency uncertainty by utilizing various window sizes, enabling more precise representations for different segments of the signal.



(a) STFT

(b) CQT

Figure 2.2: Spectrograms of the first notes of Chopin's Nocturne n°2, Op. 9.

2.3 Representing Music with Piano Rolls

While spectrograms are well-suited for audio signals, they are not directly applicable to symbolic representations of music such as MIDI files or scores. Indeed, since symbolic representations cannot be modeled as continuous functions $f : \mathcal{T} \to \mathbb{R}$, the

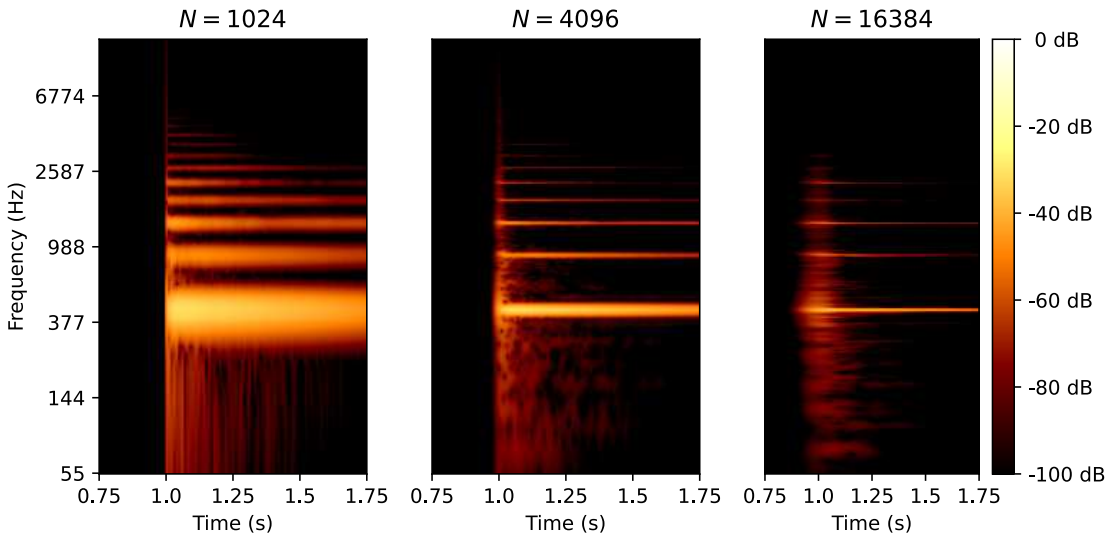


Figure 2.3: Spectrogram of TFST for three window sizes (1024, 4096, 16384) of a piano A4.

standard transformation formulas used for spectrograms are not applicable in this context.

Nevertheless, MIDI files and scores are already available in a format that straightforwardly yields a time-frequency representation. This format, commonly known as a *piano roll*, can be effectively modeled as a function within a musical space, denoted $P : \mathcal{T} \times \mathcal{F} \to \mathcal{A}$, as we will explore further in the following sections.

2.3.1 Piano Roll

Let us begin with a brief overview of piano rolls. Originally, piano rolls were a mechanical means of recording music before audio recording methods existed. They were used to preserve performances of great musicians from the early 20^{th} century, and many archives, such as the Stanford University Piano Roll Archive[16], still house these historical records.

In modern music contexts, the concept of a piano roll has evolved to encompass any piano roll-like representation of music. Specifically, it refers to a two-dimensional representation of musical notes, where one axis represents time and the other axis represents the notes of a piano. In this thesis, we adopt this extended notion of a

[16]https://exhibits.stanford.edu/supra

piano roll and provide a formal definition.

It is important to note that there is no universally accepted definition of what a piano roll should be. The term is used informally, assuming that we all refer to this particular representation. Temperley, 2004 exposes several piano rolls and points out the need of marking the onset of each note (which we will do by using the rhythmic lattice, in our case).

Thus, we present a formal definition of a piano roll.

Definition 2.21 (Piano roll)**.** Let $\mathcal{T}$ be a set representing time. Let $\mathcal{F}$ be a countable set representing pitches[17]. Let $(\mathcal{A}, \leq)$ be a complete lattice. Then, a **piano roll** P of $N \in \mathbb{N}$ notes is an element of $\mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ such that

$$P = \bigvee_{n=1}^{N} \nu_n \tag{2.49}$$

where $\forall n \in \{1, 2, ..., N\}$, $\nu_n \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ and $\text{supp}(\nu_n) = [s_n, e_n[\times\{\xi_n\}$ for some $s_n, e_n \in \mathcal{T}$ and $\xi_n \in \mathcal{F}$.

Each function $\nu_n \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ is called a *note*, with s_n being its *start*, e_n being its *end* and ξ_n being its pitch.

In this definition, a piano roll P is represented as a supremum of individual notes, each defined by a function ν_n. The specific way we define a note varies depending on the amplitude range $\mathcal{A}$ that we use.

In Chapters 4 and 5, we will use another related concept that is the *activations piano roll*, which we abbreviate as *activations*.

Definition 2.22 (Activations piano roll)**.** Let $\mathcal{T}$ be a set representing time. Let $\mathcal{F}$ be a countable set representing pitches. Let $(\mathcal{A}, \leq)$ be a either $\mathcal{A}_2$ or one of the presented dynamics $\mathcal{D}$. Then, an **activations piano roll** A of $N \in \mathbb{N}$ notes is an element of $\mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ such that

$$A = \bigvee_{n=1}^{N} \alpha_n \tag{2.50}$$

where $\forall n \in \{1, 2, ..., N\}$, $\alpha_n \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ and $\text{supp}\,\alpha_n = \{t_n\} \times \{\xi_n\}$ for some $t_n \in \mathcal{T}$ and $\xi_n \in \mathcal{F}$.

Each function $\alpha_n \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ is called an *activation*, with t_n being its *timestamp* and ξ_n being its pitch.

[17]The requirement of $\mathcal{F}$ to be countable ensures that pitches are essentially discrete, as opposed to frequencies, which can vary continuously.

It can be a bit obscure what this concept means. We present two intuitions: in the MM framework, the activations are either the input of a dilation or the output of an erosion; they mean when a structuring element should be replicated (in the case of dilation) or may be present (erosion). In the framework of music representations, each activation means that we activate a motive at a particular time-frequency point (with, eventually, a dynamic information).

Before diving into the different piano roll representations, we introduce a derived notion that will help us handle more complex musical data.

Definition 2.23 (Piano roll stack)**.** Let $\mathcal{T}$, $\mathcal{F}$ and $\mathcal{A}$ three sets that may represent a piano roll. Let I be a countable set of indexes. We call a **piano roll stack** a sequence $\boldsymbol{S} = (P_i)_{i \in I}$ of piano rolls belonging to $\mathcal{A}^{\mathcal{T} \times \mathcal{F}}$.

We have

$$\begin{array}{rccl} \boldsymbol{S}: & \mathcal{T} \times \mathcal{F} \times I & \to & \mathcal{A} \\ & (t, \xi, i) & \mapsto & \boldsymbol{S}(t, \xi, i) = P_i(t, \xi) \end{array} . \tag{2.51}$$

This mathematical object allows us to consider several piano rolls that share time, frequency, and amplitude. It is particularly useful for handling MIDI files with several tracks and scores with multiple instruments. While we presented for piano rolls, the same principle applies to activations piano rolls.

In the following sections, we explain how we define piano rolls and stacks for each input format: in Section 2.3.2, we define it for MIDI files, and in Section 2.3.3, we define it for scores. Finally, in Section 2.3.4, we introduce a derived version of a piano roll: the *chroma roll*.

2.3.2 Representing MIDI Files as Piano Rolls

MIDI is a widely used format for sharing musical data, and is particularly well adapted for being represented as piano roll with minimal information loss. In the following sections, we specify the various choices we can make for $\mathcal{T}$, $\mathcal{F}$, and $\mathcal{A}$ when dealing with MIDI files. These choices allow us to customize the piano roll representation to suit different applications and requirements.

2.3.2.1 Time

Inside a MIDI file, time is expressed using a unit called *tick*. Specifically, a MIDI file consists of a series of messages, and each message is separated by a time interval of $\Delta \in \mathbb{N}$ ticks from the previous one.

To convert ticks into seconds for playing a MIDI file, two parameters are involved:

1. The `ticks_per_beat` parameter: an integer value that represents the number of ticks per beat for the entire MIDI file. The term "beat" in this context refers to the quarter note value.

2. The `microseconds_per_beat` parameter: an integer value that can be changed by a `set_tempo` message within the file. It represents the number of microseconds per beat.

Using these two parameters, ticks can be converted into seconds to play the MIDI file. The `microseconds_per_beat` parameter allows for tempo changes within the file, providing flexibility in the playback speed.

Let us present the conversion formulas from ticks to seconds. We call $\Delta t_{\text{tk}} \in \mathbb{N}$ a time interval expressed in ticks and $\Delta t_{\text{s}}, \Delta t_{\mu\text{s}} \in \mathbb{R}$ a time interval expressed in seconds and microseconds, respectively. Let $\Delta t_{\text{b}} \in \mathbb{N}$ be a time interval expressed in beats (quarter notes).

If we call $\text{tpm} \in \mathbb{N}^*$ the value of `ticks_per_beat` and $\text{mpb} \in \mathbb{N}^*$ the value of `microseconds_per_beat`, we have that

$$\Delta t_{\text{b}} \cdot \text{tpb} = \Delta t_{\text{tk}} \qquad \Delta t_{\text{b}} \cdot \text{mpb} = \Delta t_{\mu\text{s}} \qquad \Delta t_{\mu\text{s}} = 10^6 \Delta t_{\text{s}}\,. \tag{2.52}$$

From these formulas, we can deduce:

$$\Delta t_{\text{s}} = \Delta t_{\mu\text{s}} \cdot 10^{-6} = \Delta t_{\text{b}} \cdot \text{mpb} \cdot 10^{-6} = \Delta t_{\text{tk}} \cdot \frac{\text{mpb}}{\text{tpb}} \cdot 10^{-6} = \text{spt} \cdot \Delta t_{\text{tk}}\,. \tag{2.53}$$

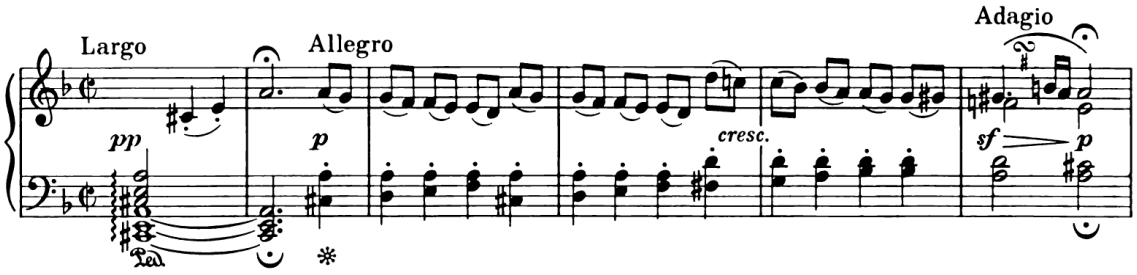
where $\texttt{seconds\_per\_tick} = \frac{\text{mpb}}{\text{tpb}} 10^{-6} \in \mathbb{Q}$ is the conversion parameter.

This combination of parameters results in two methods for measuring time within a MIDI file: using the tick unit or measuring it in seconds after conversion. These two approaches may not always yield the same results due to the possibility to change the tempo.

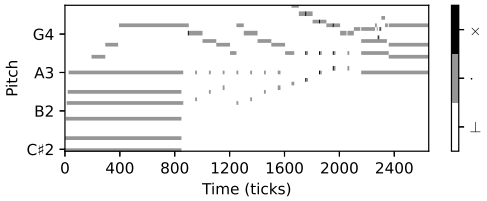
To illustrate this phenomenon, Figure 2.4 shows a musical score, which has been converted into a MIDI file using a music editor software. The MIDI file is then represented as two different piano rolls: one with time measured in ticks (Figure 2.4b) and the other with time measured in seconds (Figure 2.4c). We observe that the ***Largo*** and ***Adagio*** (with tempos set to ♩ = 48 and ♩ = 42, respectively) occupy much more time in the seconds version than in the ticks version, in contrast to the ***Allegro*** (with a tempo of ♩ = 242), which occupies much less time.

As a result, the choice of measuring time in ticks or seconds can significantly impact the representation of the MIDI file in piano roll format. Depending on the application's requirements, either method may be preferred.

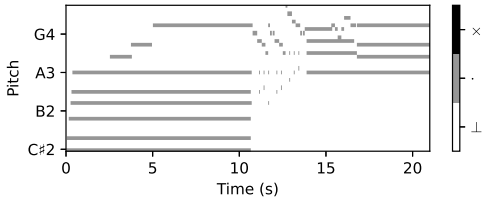
We associate the corresponding spaces and groups for each representation:



(a) Score



(b) Piano roll from a MIDI file with time expressed in ticks.



(c) Piano roll from a MIDI file with time expressed in seconds.

Figure 2.4: First bars of the first movement of Beethoven's Piano Sonata No.17, Op.31 No.2 in different representations.

- If we measure the time in ticks, we use $\mathcal{T} = \mathbb{Z}$ and $G_{\mathcal{T}} = (\mathbb{Z}, +)$.

- If we measure the time in seconds, we use $\mathcal{T} = \mathbb{Q}$ and $G_{\mathcal{T}} = (\mathbb{Q}, +)$.

2.3.2.2 Frequency

The space we choose for frequency is $\mathcal{F} = \mathcal{N}$ (the space of pitches) since it is trivially related with the MIDI numbers; we associate 60 to C4, 69 with A4, etc. This space is associated with the group $(\mathbb{Z}, +)$, where the shift 1 means shift the note one semitone up. For instance, $\mathrm{A4} + 1 = \mathrm{B}\flat 4$.

In actual MIDI files, the pitches are comprised between 0 and 127, i.e., from C-1 to G9.

Notice that, under these conditions, $\mathrm{A}\sharp = \mathrm{B}\flat$. This is a consequence of encoding frequencies by MIDI numbers inside a MIDI file.

2.3.2.3 Amplitude

In a MIDI file, the amplitude of notes is determined by the `velocity` parameter associated with each note. Additionally, the use of `note_on` and `note_off` messages specifies when a note starts and stops playing. Some MIDI files may omit the `note_off` message and instead use a `note_on` message with a velocity of 0 to represent the end of a note. For our representation, we can easily handle both cases by treating a `note_on` message with velocity 0 as equivalent to a `note_off` message.

Regarding the amplitude representation $\mathcal{A}$ for a MIDI file, there are three options:

- Using only the rhythmic range $\mathcal{A}_3$ (as discussed in Section 2.1.3.2), which omits the velocity information.
- Using a lattice that considers dynamics, as explained in Section 2.1.3.4, with the dynamics lattice being $\mathcal{D}_{128}$ (as presented in Section 2.1.3.3). There are two choices: $\mathcal{A}^P_{\mathcal{D}_{128}}$ or $\mathcal{A}_{\mathcal{D}_{128}}$.

We recall the definitions, exposed in Equations (2.12) and (2.15), of the lattices $(\mathcal{A}^P_{\mathcal{D}_{128}}, \leq^P_{128})$ and $(\mathcal{A}_{\mathcal{D}_{128}}, \leq_{128})$:

$$\mathcal{A}^P_{\mathcal{D}_{128}} = \mathcal{D}_{128} \cup \{\cdot\}$$
$$a \leq^P_{128} b \Leftrightarrow \begin{cases} a = \bot, \text{ or} \\ a = \cdot \text{ and } b \in \mathcal{D}_{128}, \text{ or} \\ a \leq_{128} b \text{ with } a, b \in \mathcal{D}_{128}\,. \end{cases}$$

$$\mathcal{A}_{\mathcal{D}_{128}} = \{\bot\} \cup \big(\{\cdot, \times\} \times \mathcal{D}_{128}\big)$$
$$a \leq_{128} b \Leftrightarrow \begin{cases} a = \bot, \text{ or} \\ a_1 \leq_3 b_1 \text{ and } a_2 \leq_{128} b_2 \end{cases}\,.$$

In Figure 2.5, we present the piano roll of a MIDI file generated from the score in Figure 2.5a with two different amplitude representations: $\mathcal{A} = \mathcal{A}_3$ (which we will use most often due to its simplicity) represented in Figure 2.5b, and $\mathcal{A} = \mathcal{A}_{\mathcal{D}_{128}}$ (a more refined representation, left for future research), represented in Figure 2.5c.

Since there are four instruments, we might represent this excerpt as a piano roll stack with the set of indexes $I = \{\textbf{Bass}, \textbf{Vla.}, \textbf{Vln. 2}, \textbf{Vln. 1}\}$, one per instrument. While the instruments are different and exhibit different timbers, they are all bowed string instruments and thus have a similar sound. This is why, we might want

to represent all of them in the same piano roll. In order to do that, we use the contraction[18] in the set of indices of $\boldsymbol{S}$, i.e.,

$$\bigvee_{i\in I} \boldsymbol{S} : \begin{array}{ccc} \mathcal{T}\times\mathcal{F} & \to & \mathcal{A} \\ (t,\xi) & \mapsto & \bigvee_{i\in I} P_i(t,\xi) \end{array} . \tag{2.54}$$

2.3.3 Representing Scores as Piano Rolls

Representing scores as piano rolls involves converting the musical data of a score into functions $P : \mathcal{T} \times \mathcal{F} \to \mathcal{A}$ in a musical space. However, it should be noted that scores can contain a vast amount of information, and not all of it can be fully rendered under this representation. The information that we choose to retain in the piano roll representation includes:

- the number of instruments, which may be merged for instruments of the same family (e.g., merging all the strings into a single representation),
- the pitch information, with potential loss of enharmonic notes,
- the rhythm, up to a certain realization, as some elements such as trills or grace notes loose their generality after being rendered,
- the dynamics information, if we choose an amplitude space such as $\mathcal{A}^P_{\mathcal{D}}$ or $\mathcal{A}_{\mathcal{D}}$.

Some of the information that we typically drop in the piano roll representation includes: the key, the bars, the time signature, the articulation, the tempo, the timber, the *divisi*, the moods, etc.

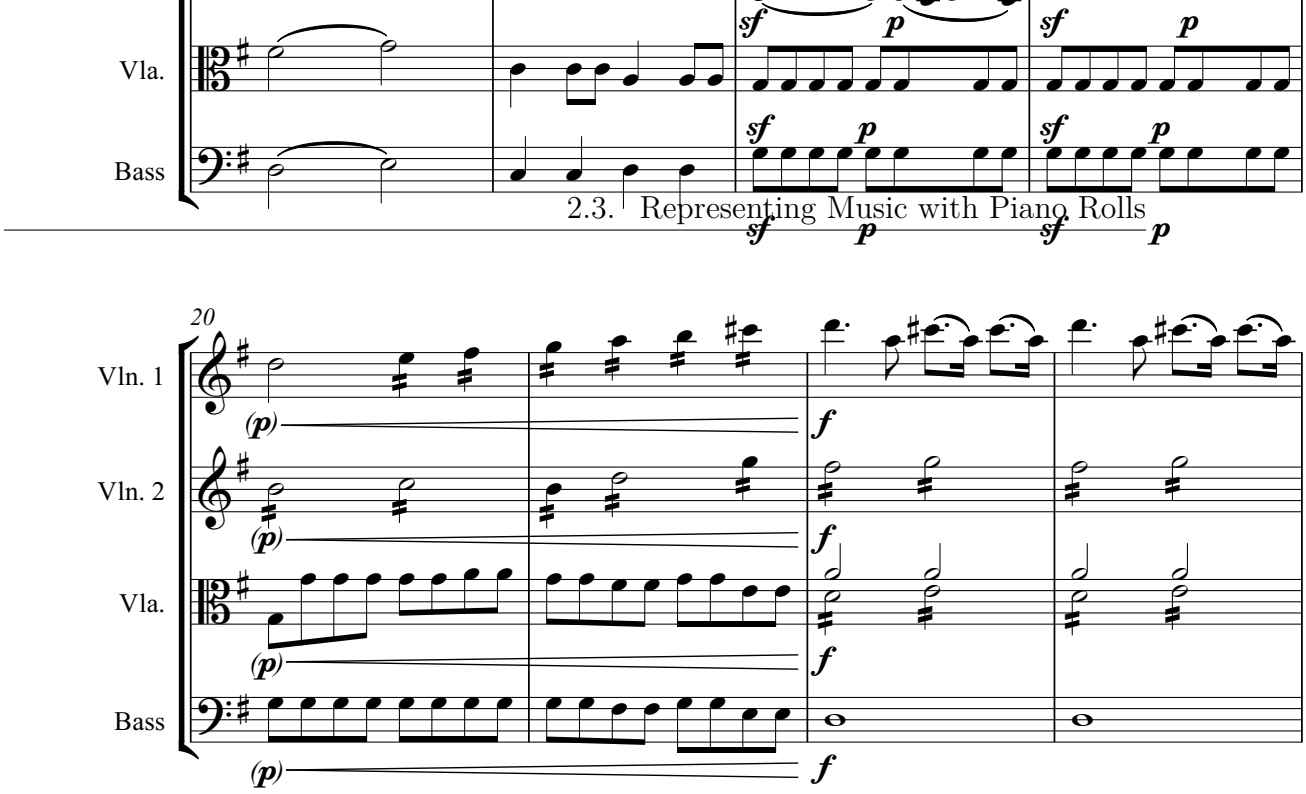
As an example, Figure 2.6 shows an actual score and a score with only the features we keep.

2.3.3.1 Time

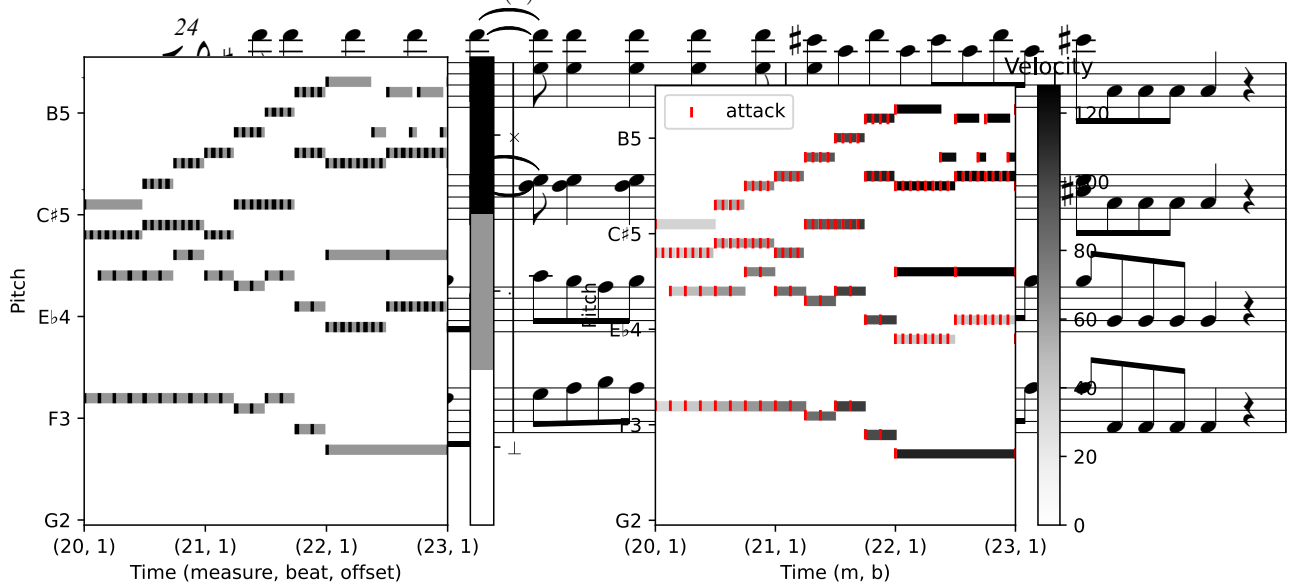
The time inside a score is measured in wholes as exposed in Section 2.1.1.3. We will then use $\mathcal{T}_{\mathbf{o}}^{\frac{p}{q}}$ as space of timestamps, with $\frac{p}{q}$ being a time signature[19].

[18]The term "contraction" is borrowed from the terminology of tensors, where the contraction of one index involves summing over the values of this index. In the context of piano roll stacks, it signifies merging multiple piano rolls into a single one, taking the supremum of all the indexed elements.

[19]We do not consider the possibility of changing the time signature, even if this is a common practice, to remain simple.



(a) Score

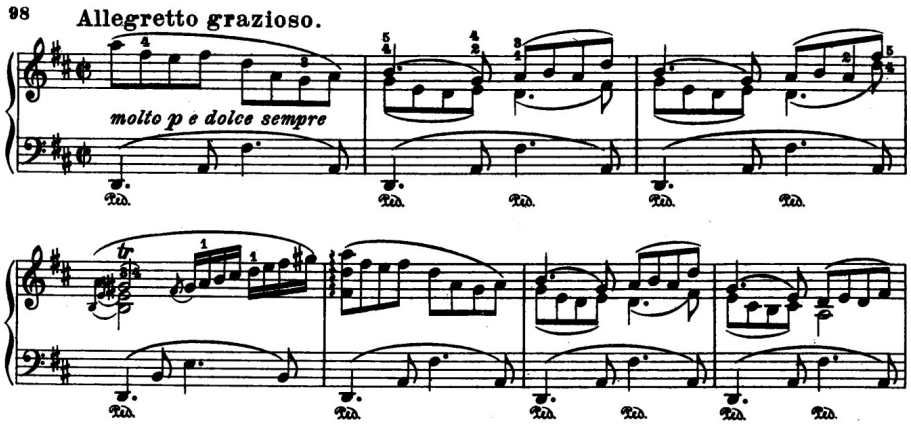


(b) Piano roll with $\mathcal{A} = \mathcal{A}_3$

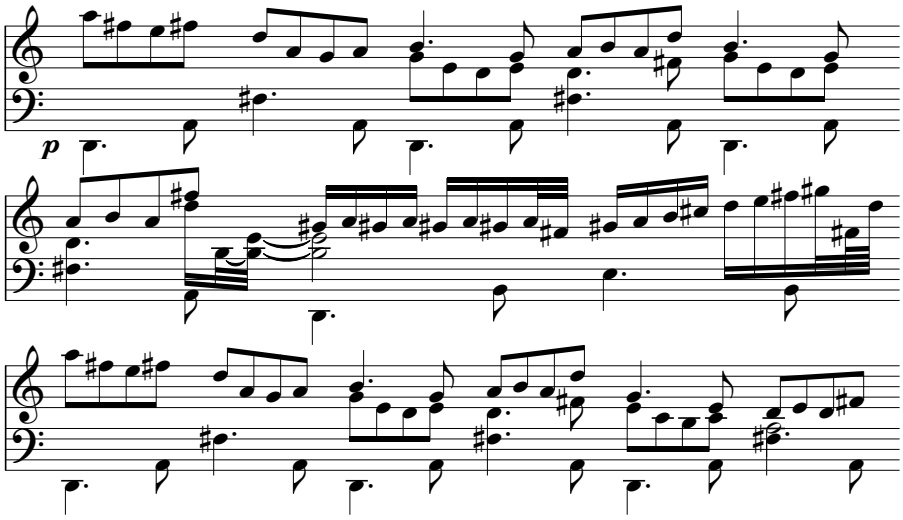
(c) Piano roll with $\mathcal{A} = \mathcal{A}_{\mathcal{D}_{128}}$

Figure 2.5: Piano roll representation of mm. 20-22 of Mozart's *Eine kleine Nachtmusik*, K.525.

To create a computational model, we need to discretize time. One approach is to arbitrarily select a minimum note value, for example, 𝅘𝅥𝅯, to serve as the smallest time unit. However, this approach may not work well when dealing with triplets or



(a) Original Score



(b) Resulting score with kept features

Figure 2.6: Difference between the score of mm. 17-23 from Brahms' *Romanze*, Klavierstücke, Op.118, N°5 and the score with only the kept features.

other complex rhythmic patterns.

To address this issue, we can determine the *tatum* of the score, which is a common notion in music theory that can be thought of as the greatest common divisor of all note values present in the score. The tatum serves as the minimal note value for our computational model, allowing us to handle various rhythmic patterns accurately.

In the representation of scores as piano rolls, we may encounter grace notes or

trills, which require determining their rhythmic realization. This means establishing how these ornamentations are performed within the time structure of the music.

2.3.3.2 Frequency

In scores, the frequency is typically quantized, except in some particular cases like the *glissando* where there may be a continuous pitch change. Despite having more information in scores compared to MIDI files (since enharmonic pitches are distinguished), we still quantize the frequency and thus have $\mathcal{F} = \mathcal{N}$.

2.3.3.3 Amplitude

In the case of scores, we do not have a velocity range like in MIDI files. However, we have a range of dynamics denoted by $\mathcal{D}_{\boldsymbol{pf}}$, as presented in Section 2.1.3.3. Similarly to MIDI files, we can choose to keep only the rhythmic component and use $\mathcal{A}_3$ to represent the amplitude lattice. This simplification allows us to focus on the essential rhythmic aspects of the score while disregarding more detailed dynamic variations. Figure 2.7 shows a representation of the first bars of Beethoven's *Pathetic* sonata in both representations.

2.3.4 Chroma Roll

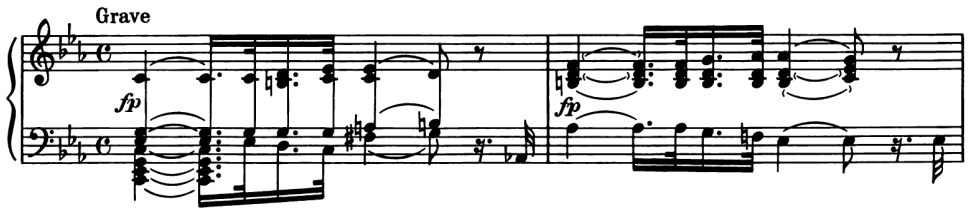
In this section, we explore the transformation of the space of pitches $\mathcal{N}$ into the space of chromas $\mathcal{N}_{12}$ to create another piano roll representation that we call *chroma roll*. The chroma roll is particularly useful for analysis, as it takes advantage of the concept of *equivalence up to the octave* commonly used in music theory.

To achieve the chroma roll representation, we consider the description of pitches given in Equation (2.6), where $\mathcal{N} = \mathcal{N}_{12} \times \mathbb{Z}$. We then project this space onto the chromas component using the first projection, denoted as π_1. This projection is defined as follows:

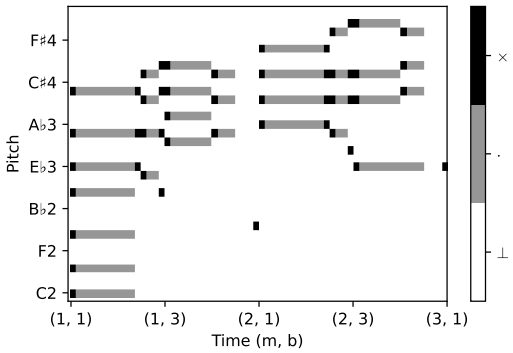
$$\begin{array}{rccc} \pi_1 : & \mathcal{N}_{12} \times \mathbb{Z} & \to & \mathcal{N}_{12} \\ & (N, n) & \mapsto & N \end{array} . \tag{2.55}$$

Next, we extend this projection to the entire musical space $\mathcal{A}^{\mathcal{T}\times\mathcal{N}}$ using the supremum operator. The extended projection, denoted as π_{12}, is given by:

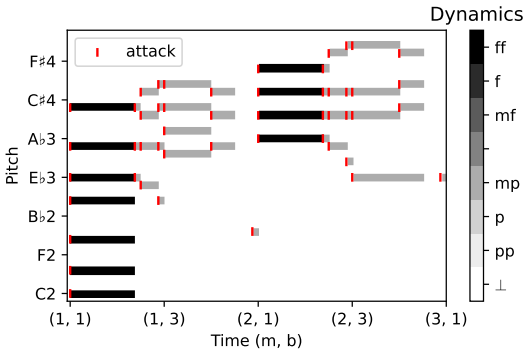
$$\begin{array}{rccccc} \pi_{12} : & \mathcal{A}^{\mathcal{T}\times\mathcal{N}} & \to & \mathcal{A}^{\mathcal{T}\times\mathcal{N}_{12}} & & \\ & P & \mapsto & \overline{P} : & \mathcal{T}\times\mathcal{N}_{12} & \to & \mathcal{A} \\ & & & & \left(t, \overline{\xi}\right) & \mapsto & \bigvee f\left(t, \pi_1^{-1}\left(\overline{\xi}\right)\right) \end{array} \tag{2.56}$$



(a) Score



(b) Piano roll with $\mathcal{A} = \mathcal{A}_3$



(c) Piano roll with $\mathcal{A} = \mathcal{A}^P_{\mathcal{D}_{\boldsymbol{pf}}}$

Figure 2.7: Piano roll representation of mm. 1-2 of Beethoven's Piano Sonata No.8, Op.13.

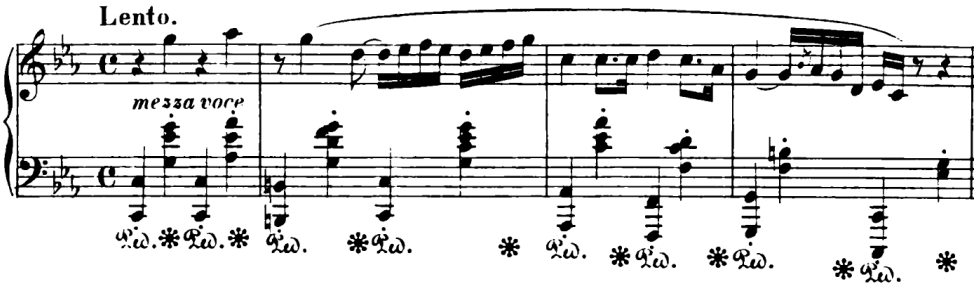
where $f\left(t, \pi_1^{-1}\left(\overline{\xi}\right)\right) = \{f(t,\eta) \in \mathcal{A} : \eta \in \pi_1^{-1}\left(\overline{\xi}\right)\}$.

Representing a chroma roll poses some challenges, as we aim to preserve the cylindrical topology of $\mathcal{T} \times \mathcal{N}_{12}$. To address this, we represent the chroma roll in two dimensions while keeping in mind that the frequency dimension wraps around.

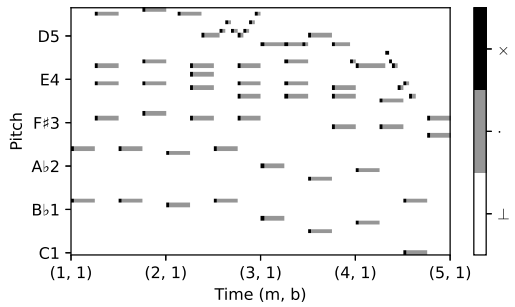
As an example, Figure 2.8 depicts the first bars of Chopin's Nocturne Op.48 Nº1 in both piano roll representation (Figure 2.8b) and chroma roll representation. The chroma roll is presented in both the flat representation (Figure 2.8c) and the cylindrical representation (Figure 2.8d). Notably, C and B are very close in the cylindrical representation (as expected) and appear far apart in the flat representation due to the limitations of plotting on two dimensions.

2.4 Conclusion

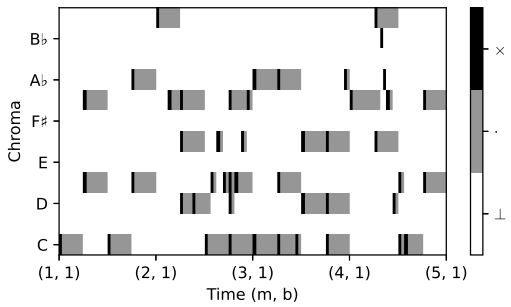
In conclusion, we have presented various time-frequency representations of music, all of which can be organized as a musical space $\mathcal{M} = \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$, with the possibility of



(a) Score



(b) Piano roll



(c) Flat chroma roll

(d) Cylindrical chroma roll

Figure 2.8: Piano and chroma roll representations of mm. 1-4 of Chopin's Nocturne Op.48 Nº1.

adding an extra dimension of indices I to represent multiple instruments or tracks.

By endowing $\mathcal{A}$ with a complete lattice structure $(\mathcal{A}, \leq)$, we establish that $(\mathcal{M}, \preceq)$ is also a complete lattice, enabling the application of Mathematical Morphology on these representations. Moreover, as the domain of functions in $\mathcal{M}$ is a space $\mathcal{T} \times \mathcal{F}$ with a group that acts on it, we can employ mathematical morphology techniques that use structuring elements.

Table 2.1 provides a summary of the representations we have presented, along with the different choices available for $\mathcal{T}$, $\mathcal{F}$, and $\mathcal{A}$. In the forthcoming chapters, we will employ some of these spaces to undertake various musicological tasks.

Representation		Space	Group	unit	Amplitude
Spectrogram	STFT	$\mathcal{T}_{\mathrm{s}} \times \mathcal{F}_{\mathrm{Hz}}$	$(\mathbb{R} \times \mathbb{R}, +)$	s × Hz	$\overline{\mathbb{R}}^{+}$ or $[0,1]$
					$\overline{\mathbb{R}}$ or $\overline{\mathbb{R}}^{-}$ (in dB)
	CQT	$\mathcal{T}_{\mathrm{s}} \times \mathcal{F}_{\mathrm{Hz}}^{\log}$	$(\mathbb{R}, +) \times (\mathbb{R}^{+*}, \cdot)$	s × Hz	$\overline{\mathbb{R}}^{+}$ or $[0,1]$
					$\overline{\mathbb{R}}$ or $\overline{\mathbb{R}}^{-}$ (in dB)
		$\mathcal{T}_{\mathrm{s}} \times \mathcal{F}_{\mathrm{st}}$	$(\mathbb{R} \times \mathbb{R}, +)$	s × st	$\overline{\mathbb{R}}^{+}$ or $[0,1]$
					$\overline{\mathbb{R}}$ or $\overline{\mathbb{R}}^{-}$ (in dB)
Piano roll	MIDI file	$\mathcal{T}_1 \times \mathcal{N}$	$(\mathbb{Z} \times \mathbb{Z}, +)$	tick × st	$\mathcal{A}_3$, $\mathcal{A}_{128}$ or $\mathcal{A}_{128}^{P}$
	Score	$\mathcal{T}_{\mathbf{o}}^{\frac{p}{q}} \times \mathcal{N}$	$(\mathbb{Q} \times \mathbb{Z}, +)$	$\mathbf{o}$ × st	$\mathcal{A}_3$, $\mathcal{A}_{\mathcal{D}_{\boldsymbol{pf}}}^{P}$ or $\mathcal{A}_{\mathcal{D}_{\boldsymbol{pf}}}$
Chroma roll	MIDI file	$\mathcal{T}_1 \times \mathcal{N}_{12}$	$(\mathbb{Z} \times \mathbb{Z}_{12}, +)$	tick × st	$\mathcal{A}_3$, $\mathcal{A}_{128}$ or $\mathcal{A}_{128}^{P}$
	Score	$\mathcal{T}_{\mathbf{o}}^{\frac{p}{q}} \times \mathcal{N}_{12}$	$(\mathbb{Q} \times \mathbb{Z}_{12}, +)$	$\mathbf{o}$ × st	$\mathcal{A}_3$, $\mathcal{A}_{\mathcal{D}_{\boldsymbol{pf}}}^{P}$ or $\mathcal{A}_{\mathcal{D}_{\boldsymbol{pf}}}$

Table 2.1: Representations of music depending on the choices for time and frequency.

Chapter 3

Analyzing Spectrograms with Mathematical Morphology

In the preceding chapters, we established a framework for applying mathematical morphology to various time-frequency representations of music. Particularly, we demonstrated that a spectrogram can be viewed as a representation of the form $\mathcal{A}^{\mathcal{T}\times\mathcal{F}}$, enabling the application of morphological operators based on structuring elements.

While MM has been applied to analyze spectrograms (Steinberg & O'Shaughnessy, 2008; Cadore et al., 2011; Xu et al., 2014; Zhang et al., 2015) of speech, it is not up to (Romero-García, Agón, et al., 2022) that it was first applied to analyze spectrograms of music.

The primary objective of this chapter is to synthesize an audio signal $y(t)$ that closely resembles an input signal $x(t)$ from a musical instrument. To achieve this, we need a method to analyze the input signal $x(t)$ and a method to synthesize the output signal $y(t)$.

The approach we take is to transform $x(t)$ into a spectrogram $S(\tau, \omega)$ and then apply MM operators to extract features that can be used to synthesize the output signal $y(t)$. The features we need to extract are determined by the synthesis model we use.

In this thesis, we employ the *spectral modeling synthesis* (SMS) (X. Serra & Smith, 1990) for generating sounds of musical instruments. SMS involves analyzing the spectrum of a signal to extract its spectral features and subsequently synthesizing a new signal that closely resembles the original one. We adopt an extension of the SMS model, known as the STN model (Sines plus Transients plus Noise) (T. S. Verma & Meng, 2000), which incorporates an additional component for transients.

The original SMS model, which consists of sines plus noise, is well-suited for many musical instrument signals, as it captures the significant sinusoidal component (e.g., from bowed strings or winds) and the accompanying noise component (e.g., the sound of the bow or the blow). However, some instruments possess a transient component that significantly contributes to their sound (e.g., plucked or struck strings and idiophones like wood blocks, marimbas, or vibraphones). As a result, the STN model proves to be more accurate in such cases.

The task we want to perform can be summarized as follows:

1. Transform the input signal $x(t)$ into a spectrogram $S(\tau, \omega)$.

2. Apply MM operators to estimate the parameters of the STN model.

3. Synthesize the output signal $y(t)$ with the STN model.

The overall pipeline is depicted in Figure 3.1.

In the following, we introduce the STN model in Section 3.1, and we elaborate on the MM pipeline for parameter extraction in Section 3.2. Finally, in Section 3.3, we showcase and analyze the results and limitations of our method when applied to various musical instruments.

3.1 Sines, Transients and Noise Model

The process of generating synthetic sounds requires a suitable synthesis model. The earliest model for sound synthesis was additive synthesis, which can be traced back to the work of Fourier, 1888 and Helmholtz, 1865.

However, a significant improvement in sound quality was achieved with the introduction of Sines + Noise synthesis (X. Serra & Smith, 1990), particularly for music instrument sounds.

The latest evolution in sound synthesis came with the incorporation of a transient component, resulting in the Sines + Transients + Noise (STN) synthesis. This breakthrough emerged from a series of papers published towards the end of the century (T. S. Verma et al., 1997; T. Verma & Meng, 1998; T. S. Verma & Meng, 2000). The STN model has since garnered significant research interest and has seen further developments in recent times (Driedger et al., 2014; Füg et al., 2016; Fierro & Välimäki, 2023).

The STN model is based on decomposing a signal $y(t)$ into three components: the sines $s(t)$, the transients $h(t)$, and the noise $w(t)$. The equation representing this

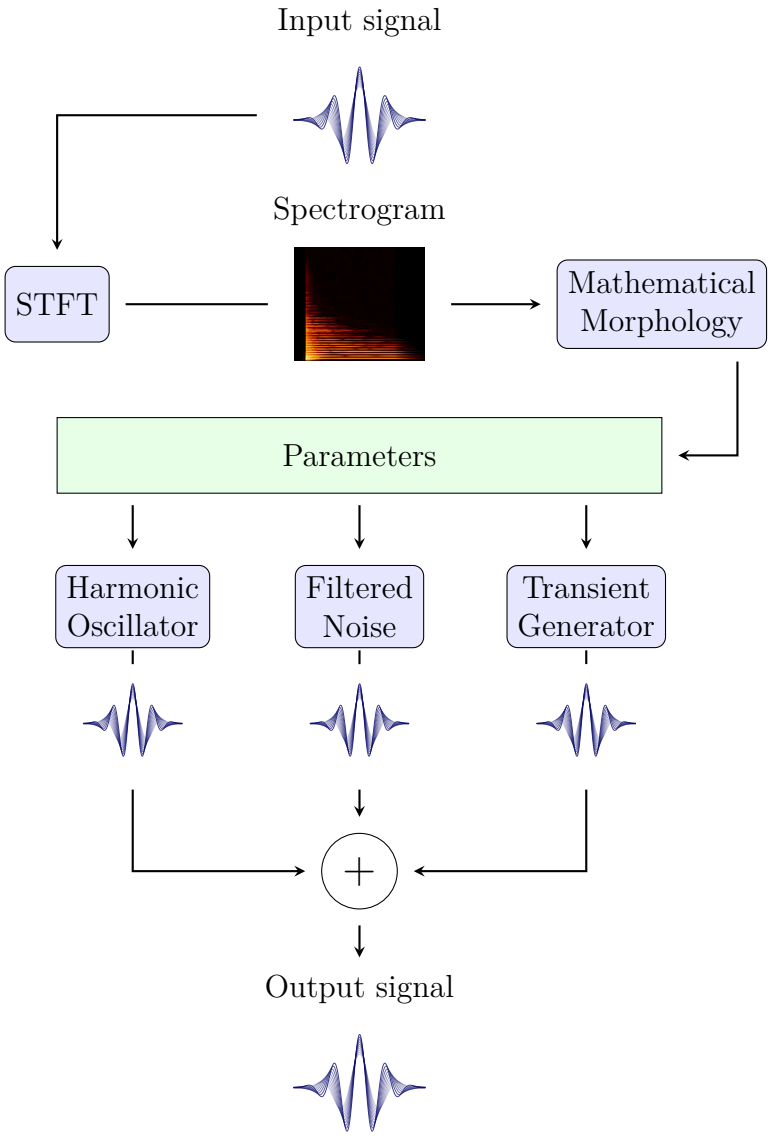


Figure 3.1: Pipeline for the synthesis of signals.

decomposition is given by:

$$y(t) = s(t) + h(t) + w(t), \tag{3.1}$$

where t is a variable representing time measured in seconds. The domain of t will depend on whether we are in the continuous case ($t \in \mathbb{R}$) or in the discrete case

($t = T_s n \in T_s\mathbb{Z}$), where T_s is the sampling period[1].

3.1.1 Sinusoidal Oscillators

To generate the sines, we use a sinusoidal oscillator, a well-known technique in the literature (see (Smith, 2011)). The formula for generating the sines is as follows:

$$\begin{aligned} s(t) &= \sum_{i=1}^{I} s_i(t) \\ &= \sum_{i=1}^{I} a_i(t) \sin\big(2\pi\Phi_i(t)\big) \end{aligned} \tag{3.2}$$

where $I \in \mathbb{N}^*$ is the number of sinusoidal components, $a_i(t)$ is the time-varying amplitude of the i^{th} component s_i, and $\Phi_i(t) = \int_0^t \xi_i(u)\,du + \Phi_0$, with $\xi_i(t)$ and Φ_0 are the instantaneous frequency at the time t and the starting phase of the i^{th} component, respectively.

The parameters to be estimated are then $a_i(t)$ and $\xi_i(t)$ (we drop the starting phase information Φ_0 and set it to 0), from which we can deduce $\Phi_i(t) = \int_0^t \xi_i(u)\,du$.

3.1.2 Filtered Noise

To generate the stochastic part $w(t)$, we create a white noise signal and filter it using a linear time-varying filter (LTV filter).

First, we generate the white noise by using a random process distributed as a normal distribution $\mathcal{N}(0, \sigma)$ with a standard deviation $\sigma \in \mathbb{R}^+$. We select a value of σ that results in an overall power density of 0 dB. The process of finding the appropriate σ is described in Section 3.2.2.2.

Next, we apply the LTV filter to the white noise. To do this, we follow the approach known as the *STFT filter* (Boashash, 2016), which involves the following three steps:

1. Calculate the short-time Fourier transform (STFT) $S(\tau, \omega)$ of the input signal $x(t)$.

2. Multiply $S(\tau, \omega)$ by a weight function $\Theta(\tau, \omega)$.

[1]The sampling period T_s is the inverse of the sampling frequency ξ_s, i.e., $T_s = \frac{1}{\xi_s}$.

3. Synthesize the output signal $y(t)$ by performing an inverse STFT of $S(\tau, \omega) \cdot \Theta(\tau, \omega)$.

The filter itself is represented by the parameter $\Theta(\tau, \omega)$, that we call *mask*, which is the parameter we aim to estimate using mathematical morphology as discussed in Section 3.2.

3.1.3 Transient Generation

To generate the transient component, we follow the approach presented in (T. S. Verma & Meng, 2000). However, instead of using the inverse cosine transform as in the original paper, we use the usual Fourier transform and adapt the formulas accordingly.

The transient is considered the dual of a sinusoid, as sinusoids appear as horizontal lines in spectrograms, while transients appear as vertical lines. To achieve the rotation in the time-frequency plane, we use the Fourier transform. Specifically, we make use of the property stated in Proposition 2.5:

$$\mathrm{STFT}_g[x](\tau, \omega) = e^{-2\pi\imath\omega\cdot t}\mathrm{STFT}_{\mathscr{F}[g]}[\mathscr{F}[x]](\omega, -\tau)\,. \tag{3.3}$$

To generate the transient, suppose we want a transient with amplitude $a(\xi)$ and time $t(\xi)$ (where frequency ξ is the variable). First, we generate the signal

$$s(\xi) = a(\xi)\big(e^{2\pi\imath\Phi(\xi)} + e^{2\pi\imath\Phi(-\xi)}\big) \tag{3.4}$$

where $\Phi(\xi) = \int_0^\xi t(\nu)\,d\nu$. Then, we obtain the transient component $h(t)$ by applying the Fourier transform to $s(\xi)$:

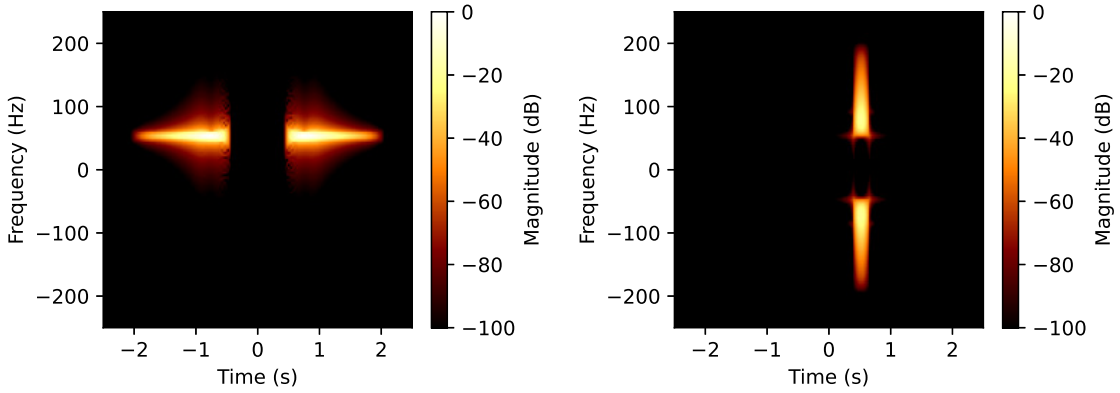
$$h(t) = \mathscr{F}[\xi \mapsto s(\xi)](t)\,. \tag{3.5}$$

The result of this process is illustrated in Figure 3.2.

When transients exhibit significant variations in amplitudes across frequencies, they are prone to experiencing temporal leakage. This temporal leakage poses a problem as it introduces sinusoids before and after the transient, which interferes with the main feature of the transient - its time concentration. Such artifacts are undesirable.

Although this issue was not explicitly addressed in (T. S. Verma et al., 1997; T. Verma & Meng, 1998; T. S. Verma & Meng, 2000), we propose a solution to mitigate this problem. We apply a window to the transient that is equal to 1 between $\min_{\xi\in\mathbb{R}} t(\xi)$ and $\max_{\xi\in\mathbb{R}} t(\xi)$, and zero outside of this interval[2].

[2]To further avoid artifacts, we actually apply a filter to this window with a Hann kernel of 5 ms.



(a) Spectrogram of signal $s(\xi)$

(b) Spectrogram of signal $h(t)$

Figure 3.2: Rotation of the time-frequency plane by $\mathscr{F}$.

Similarly as for sinusoids, we create multiple transient components and add them together. While many common musical instruments have only one transient, the STN model allows us to generate an arbitrary number of transients, offering flexibility for other applications.

3.2 Mathematical Morphology Analysis

In this section, we will provide a detailed explanation of the process of estimating the parameters for the given signal $x(t)$ using MM. The parameters we need to estimate are as follows:

1. For the sinusoidal oscillators, we need to estimate the number of components $I \in \mathbb{N}$, the amplitudes $a_i^s(t)$ and the frequencies $\xi_i(t)$, $\forall i \in I$.

2. For the filtered noise, we need to estimate the mask $\Theta(\tau, \omega)$.

3. For the transient generator, we need to estimate the number of transient components $P \in \mathbb{N}$, the amplitudes $a_p^h(\xi)$ and the times $t_p(\xi)$, $\forall p \in P$.

3.2.1 Discrete Version of the Problem

Let us now expose the discrete version of the problem. We consider an input signal $x(t) \in [-1, 1]^{\mathbb{R}}$ with the time measured in seconds and finite support $[0, T]$, with

$T \in \mathbb{R}^+$. To transform it into a discrete array, we sample it at a sampling frequency $\xi_s \in \mathbb{R}^+$ measured in Hertz (with corresponding sampling period $T_s = \frac{1}{\xi_s}$ measured in seconds), and we quantize the amplitudes in floating point values of 32 bits.

We get then a number of samples $N = \lceil T \cdot \xi_s \rceil \in \mathbb{N}^*$, corresponding to the timestamps $t_n = nT_s \in \mathbb{R}$ measured in seconds. We define the discrete input signal as

$$\{\boldsymbol{x}[n]\}_{n=0}^{N-1} \in [-1, 1]^N,$$

with $\boldsymbol{x}[n] = x(t_n)$, $\forall n \in \{0, 1, ..., N-1\}$.

To transform the discrete signal $\{\boldsymbol{x}[n]\}_{n=0}^{N-1}$ into a time-frequency representation, we use the discrete STFT exposed in Section 2.2.2.1; we leave for future research the use of the CQT and the TFST.

We choose a STFT with a window $\boldsymbol{g}$ of size $J \in \mathbb{N}^*$ and center sample $j_0 \in \{0, 1, ..., J-1\}$. We choose a hop size of $H \in \mathbb{N}^*$ and a number of frequency bins $K \in \mathbb{N}^*$.

By setting $M = \lceil \frac{N}{H} \rceil \in \mathbb{N}^*$ and adapting Equation (2.39) to our case, we obtain:

$$\boldsymbol{Z}[m, k] = \sum_{j=0}^{J-1} \boldsymbol{x}[m + j - j_0]\overline{\boldsymbol{g}[j]} e^{-2\pi\imath(m+j-j_0)\frac{k}{K}} .$$

The STFT array $\{\boldsymbol{Z}[m, k]\}_{m=0,\,k=0}^{M-1,\,K-1} \in \mathbb{C}^{M \times K}$ corresponds to the time-frequency points $\{(\boldsymbol{\tau}[m], \boldsymbol{\omega}[k]\}_{m=0,k=0}^{M-1,K-1}$ that are given by:

- $\boldsymbol{\tau}[m] = t_{mH} \in \mathbb{R}$ with time precision $T_p = HT_s \in \mathbb{R}$.
- $\boldsymbol{\omega}[k] = k\frac{\xi_s}{K} \in \mathbb{R}^+$ with frequency precision $\xi_p = \frac{\xi_s}{K} \in \mathbb{R}^+$.

$\boldsymbol{\tau}$ and T_p are measured in seconds and $\boldsymbol{\omega}$ and ξ_p are measured in Hertz.

The window function $\boldsymbol{g}$ is chosen to be real, positive and symmetric, having then $\boldsymbol{g}^* = \boldsymbol{g}$. Its length is $J \in \mathbb{N}^*$ which gives a time observation $T_o = JT_s \in \mathbb{R}$ seconds. We have then an array $\{\boldsymbol{g}[j]\}_{j=0}^{J-1} \in (\mathbb{R}^+)^J$. We set $j_0 = \lfloor \frac{J}{2} \rfloor$ as the center sample. We normalize it such that $||\boldsymbol{g}||_1 = 1$ in order to satisfy the conditions of Equation (2.21).

The spectrogram $\{\boldsymbol{S}[m, k]\}_{m=0,\,k=0}^{M-1,\,K-1} \in [-\infty, 0]^{M \times K}$ is calculated from the STFT array using the formula:

$$\boldsymbol{S}[m, k] = 10 \log_{10} |\boldsymbol{Z}[m, k]|^2 . \tag{3.6}$$

The values used for the parameters are given in Table 3.1. The parameters T, N, and M depend on the input signal and are not explicitly listed in the table.

Parameter	Value	Parameter	Value
ξ_s	44 100 Hz	H	44
T_s	2.2×10^{-5} s	K	4096
$\boldsymbol{g}$	Blackman window	T_p	1 ms
J	2048	ξ_p	10 Hz
j_0	1024	T_o	46 ms

Table 3.1: Choice of the parameters for the computations.

3.2.2 Processing Pipeline

The main contribution of this chapter is the use of MM for extracting the following parameters:

1. For the harmonic oscillator:
   (a) the number of sines I,
   (b) the amplitudes $\{\boldsymbol{a}_i^s[m]\}_{m=0}^{M-1}$ of each sine,
   (c) the frequencies $\{\boldsymbol{\xi}_i[m]\}_{m=0}^{M-1}$ of each sine.
2. For the filtered noise:
   (a) the mask $\{\Theta[m,k]\}_{m=0,\,k=0}^{M-1,\,K-1}$.
3. For the transients:
   (a) the number of transients P,
   (b) the amplitudes $\{\boldsymbol{a}_p^h[k]\}_{k=0}^{K-1}$ of each transient,
   (c) the times $\{\boldsymbol{t}_p[k]\}_{k=0}^{K-1}$ of each transient.

To estimate these parameters, we input the spectrogram $\boldsymbol{S}$ into the morphological pipeline described in Figure 3.3. In the next sections, we will provide a detailed explanation of each step.

For illustrative purposes, we use an input signal of a woodblock, as it exhibits all three features required for our analysis: the sinusoidal component, the noise component, and a transient. The spectrogram of the woodblock sound is shown in Figure 3.4. This spectrogram will serve as the input to the morphological pipeline.

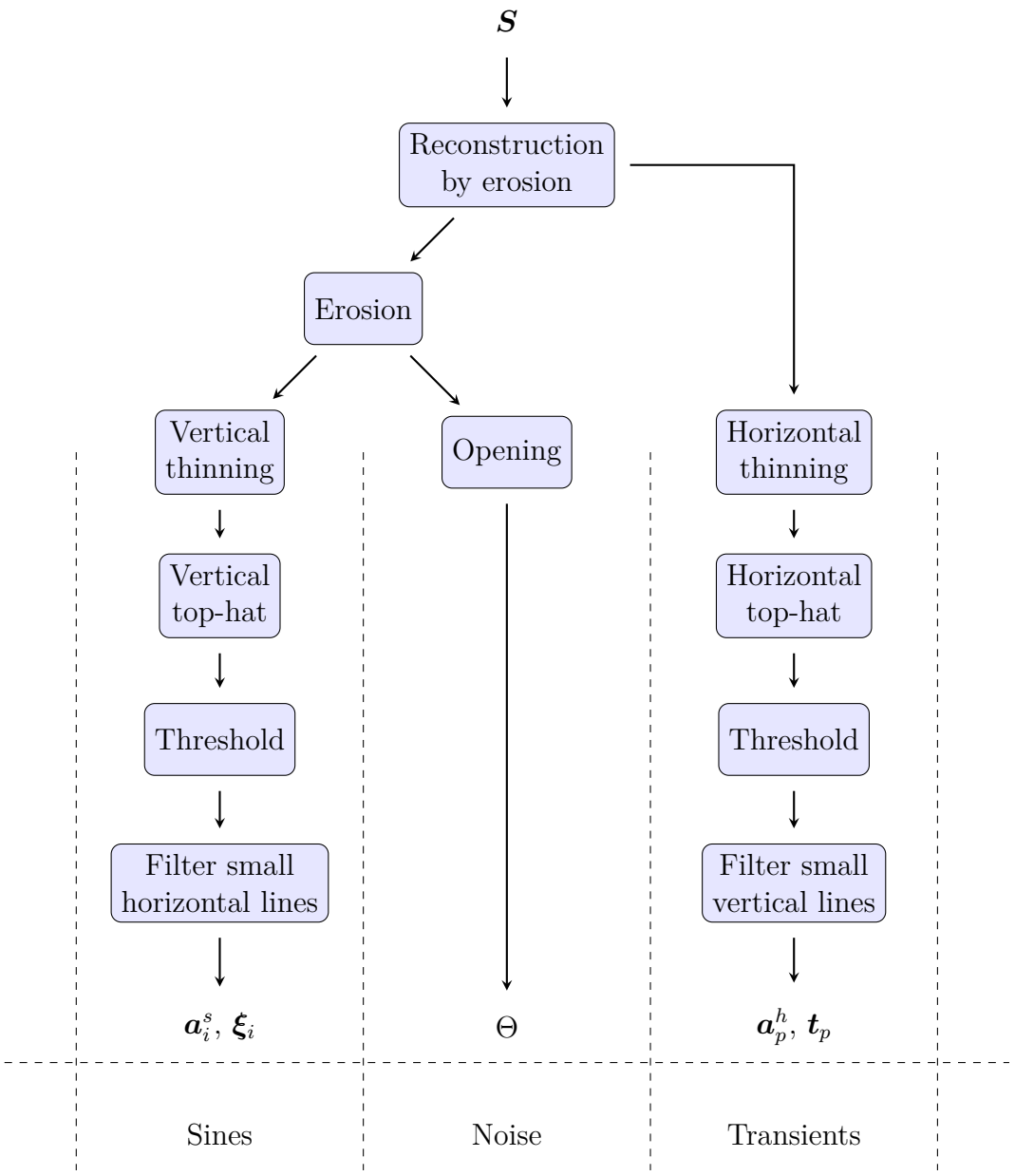


Figure 3.3: Pipeline for the morphological processing.

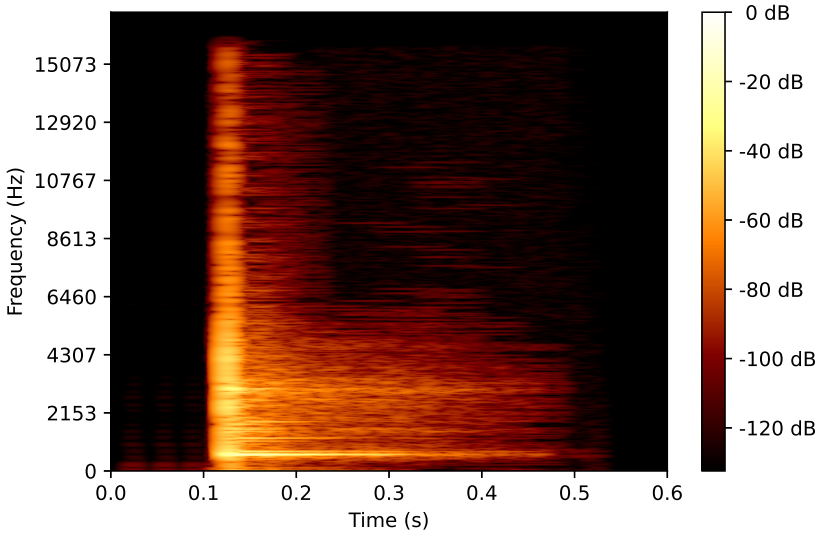


Figure 3.4: Spectrogram a woodblock.

3.2.2.1 Pre-processing

To prepare the input spectrogram for further processing, we apply two consecutive morphological operations: reconstruction by erosion and erosion.

Reconstruction by erosion

The first step of our processing is to "fill the holes" in the spectrogram. The noisy part of a spectrogram often contains holes and hills, as shown in Figure 3.5a. To ensure that the subsequent operations are not biased by the presence of holes, we use the reconstruction by erosion technique, as explained in Section 1.2.5.3.

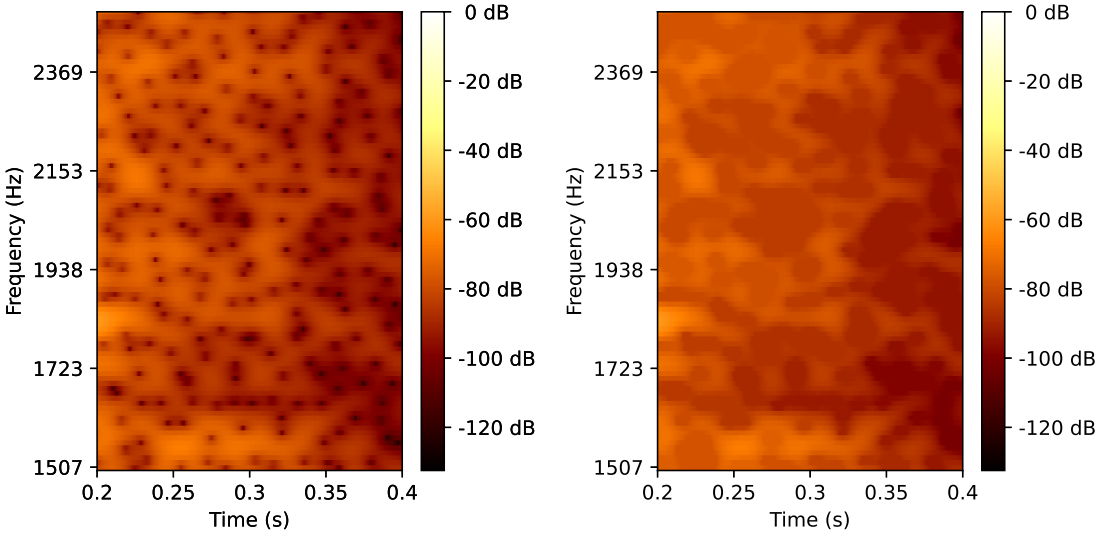
The marker function for the reconstruction by erosion is the zero function, which is the top element of the space of functions $[-\infty, 0]^{\mathbb{R}\times\mathbb{R}}$ (or, in its discrete version, the space of arrays $[-\infty, 0]^{M\times K}$).

Erosion

Once we obtain the result of the reconstruction by erosion, we proceed to apply a greyscale erosion. The structuring element $\boldsymbol{b}$ we use for this operation is the window function in dB, i.e., if $\boldsymbol{g}$ is the window, and it is given by:

$$\boldsymbol{b} = 20\log_{10}(\boldsymbol{g})\,. \tag{3.7}$$

This step is crucial as it helps to reduce the temporal leakage of the spectrogram and ensures that the masks for the noise and sinusoids are accurately aligned with



(a) Noisy spectrogram

(b) Reconstruction by erosion

Figure 3.5: Noisy part of a spectrogram and its reconstruction by erosion.

the actual onsets. The erosion operation is shown in Figure 3.6. The resulting eroded spectrogram serves as the input for the subsequent steps in our processing pipeline.

3.2.2.2 Processing for the noise component

To obtain the mask Θ for filtering the noise, we simply need to apply an opening operation to the erosion obtained in the previous step.

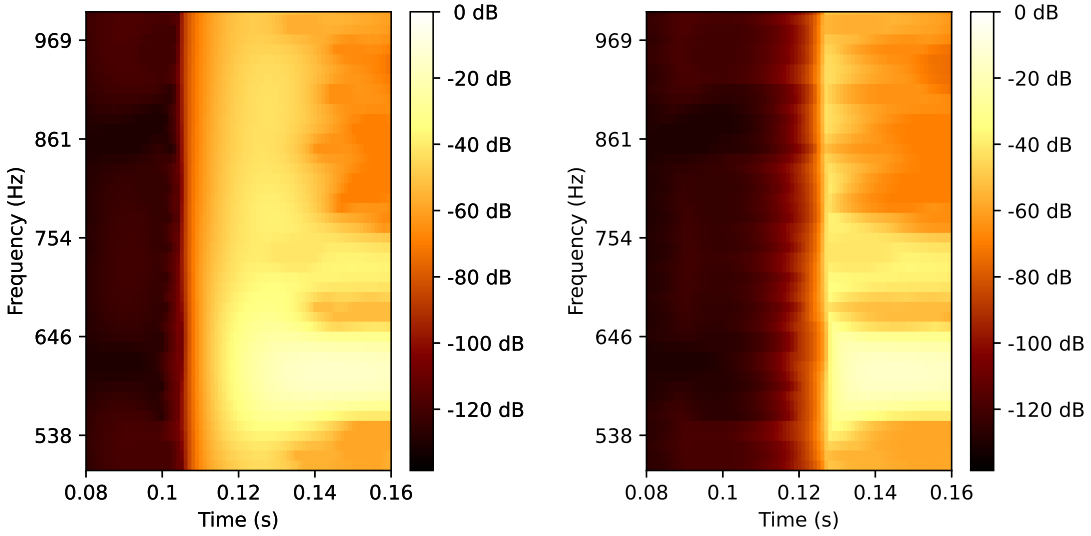
Opening

The structuring element used for the opening is a square with the sizes t_w and ξ_w, which satisfy the following conditions:

$$20\log_{10}(w(0)) - 20\log_{10}(w(t)) > 60\,\text{dB},\ \forall t \in \mathbb{R} : |t| > \frac{t_w}{2} \tag{3.8}$$

$$20\log_{10}(\hat{w}(0)) - 20\log_{10}(\hat{w}(\xi)) > 60\,\text{dB},\ \forall \xi \in \mathbb{R} : |\xi| > \frac{\xi_w}{2}\,, \tag{3.9}$$

This means that the width of the square ensures a 60 dB drop both in time and frequency. With the parameters provided in Table 3.1, we get $t_w = 44\,\text{ms}$ and $\xi_w = 193\,\text{Hz}$ for the time and frequency dimensions of the rectangular structuring element, respectively.



(a) Reconstruction by erosion

(b) Erosion

Figure 3.6: Erosion of the reconstruction by erosion for avoiding temporal leakage.

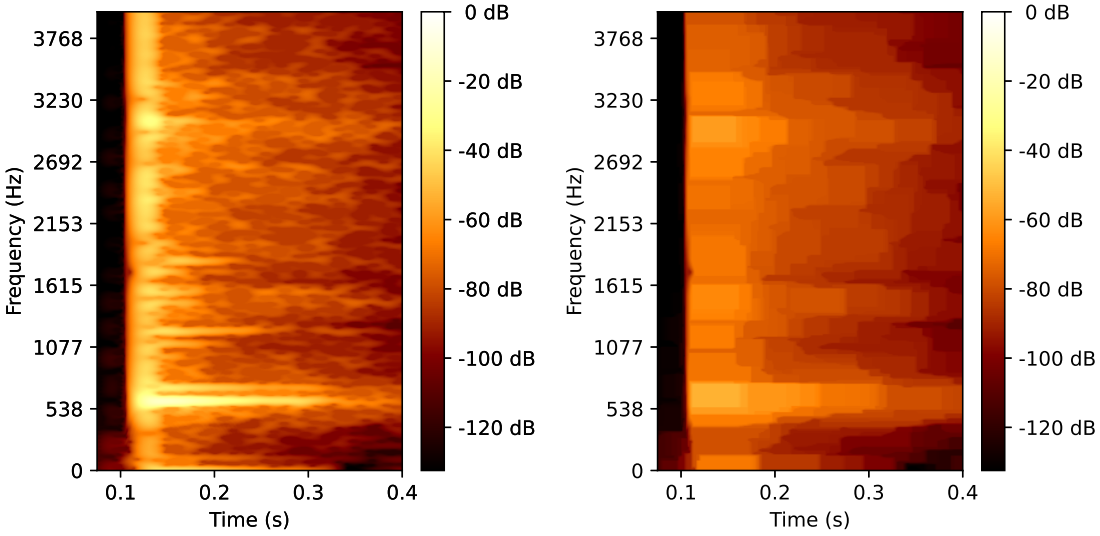
The result of applying the opening operation to the erosion result is shown in Figure 3.7. The resulting image now resembles the overall shape of the noise component.

The importance of the reconstruction by erosion step becomes visible in this process. Its absence would lead the opening operation to propagate the values of the holes in the spectrogram rather than achieving the desired average value of 0 dB. This effect is illustrated in Figure 3.8, where we observe that applying the opening before the reconstruction by erosion (Figure 3.8b) results in a not constant image. Conversely, applying the opening after the reconstruction yields a uniform image with an average value of 0 dB.

In order to achieve an average value of 0 dB after the reconstruction by erosion, we tested several values of σ and determined that setting $\sigma = 30$ achieves this desired outcome[3], as illustrated in Figure 3.8.

The application of an erosion to the reconstruction by erosion is a key step, as it ensures that the noise component aligns accurately with the transient. Without this erosion step, the noise component could start before the transient. In Figure 3.9, we display the mask and the filtered noise side by side, highlighting the effects of

[3]It is worth noting that σ is dimensionless, as the amplitude range is also dimensionless.



(a) Reconstruction by erosion

(b) Opening

Figure 3.7: Opening of the reconstruction by erosion.

temporal leakage that can occur if the erosion is not applied.

We have now obtained the mask Θ, which allows us to effectively filter the white noise. The result is presented in Figure 3.10, where we can observe both the original spectrogram and the spectrogram of the filtered noise. Notably, the noisy part is accurately recovered, showcasing the success of the filtering process in faithfully reconstructing the noise component.

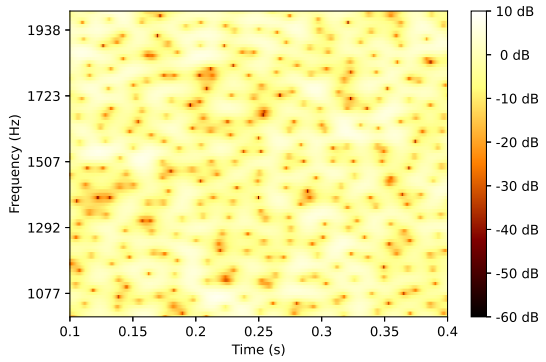
3.2.2.3 Processing for the sinusoidal component

We now explain how we use morphological operators to estimate the parameters for the harmonic oscillator, which are I, $\boldsymbol{a}_i^s$ and $\boldsymbol{\xi}_i$. We use as input for the processing the erosion since it has no holes and no temporal leakage.

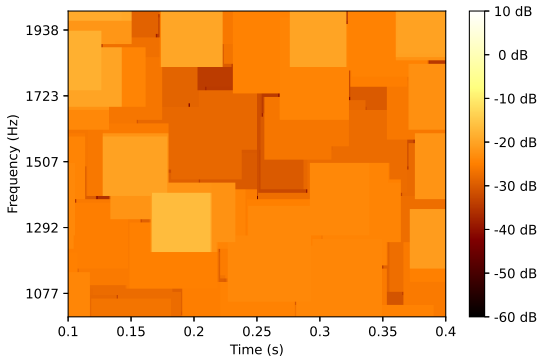
Vertical thinning

The first operator we apply is a vertical thinning; the thinning operator is explained in Definition 1.33. For obtaining a vertical thinning, i.e., for contracting the image in the vertical direction to obtain horizontal components, we need to remove the north, south, north-east, south-west, north-west and south-east points[4].

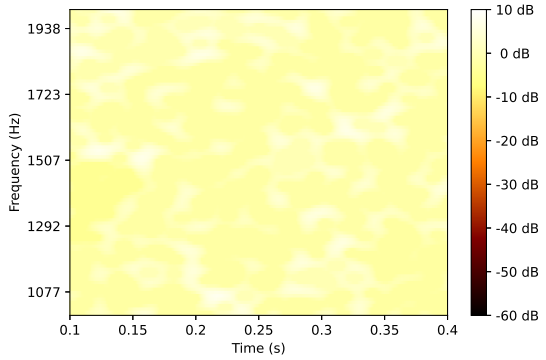
[4]Since we shall make a choice in the order of removal of the points, we choose this order, i.e., N,



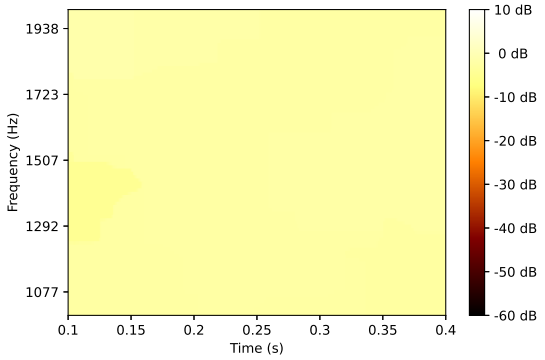
(a) Spectrogram of white noise



(b) Opening before the reconstruction by erosion



(c) Reconstruction by erosion



(d) Opening after the reconstruction by erosion

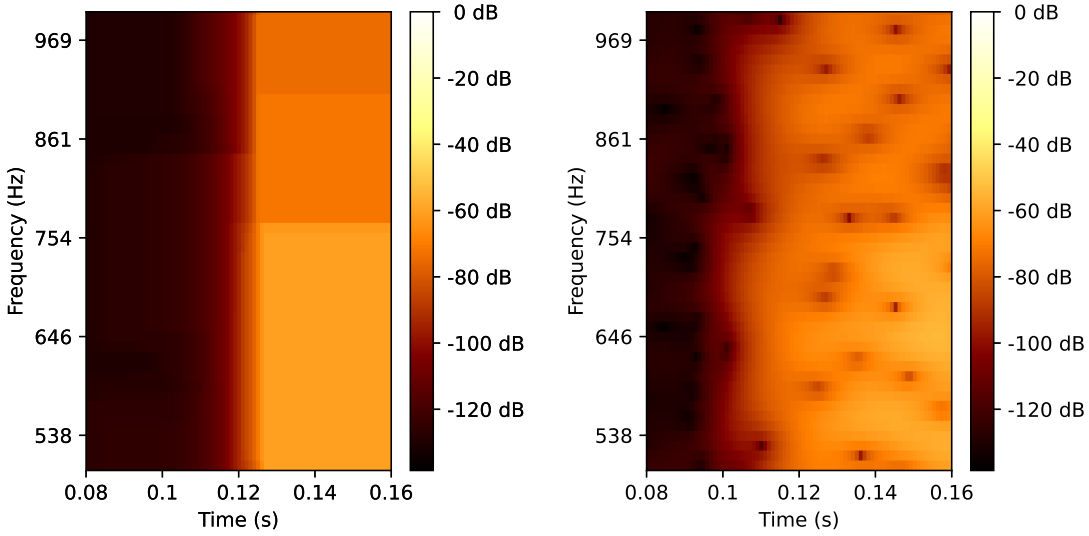
Figure 3.8: The result of applying the morphological processing to white noise with $\sigma = 30$.

To remove these points, we select the following pair of structuring elements:

$$(C,D)_N = \begin{bmatrix} 0 & 0 & 0 \\ - & 1 & - \\ - & 1 & - \end{bmatrix} \quad (C,D)_{NE} = \begin{bmatrix} - & 0 & 0 \\ 1 & 1 & 0 \\ - & 1 & - \end{bmatrix} \quad (C,D)_{NW} = \begin{bmatrix} 0 & 0 & - \\ 0 & 1 & 1 \\ - & 1 & - \end{bmatrix}$$

$$(C,D)_S = \begin{bmatrix} - & 1 & - \\ - & 1 & - \\ 0 & 0 & 0 \end{bmatrix} \quad (C,D)_{SW} = \begin{bmatrix} - & 1 & - \\ 0 & 1 & 1 \\ 0 & 0 & - \end{bmatrix} \quad (C,D)_{SE} = \begin{bmatrix} - & 1 & - \\ 1 & 1 & 0 \\ - & 0 & 0 \end{bmatrix}$$

S, NE, SW, NW, SE.



(a) Opening

(b) Spectrogram of the filtered noise

Figure 3.9: Temporal leakage.

These patterns should be interpreted in the following manner: ones correspond to the elements of set C, zeroes correspond to the elements of set D, and $-$ means that the point is not considered in the structuring element. Moreover, we assume that the origin is located at the center pixel. More formally, if we denote the matrix associated with the pattern as $(a_{ij}) : i, j = 1, 2, 3$, the corresponding sets C and D are given by:

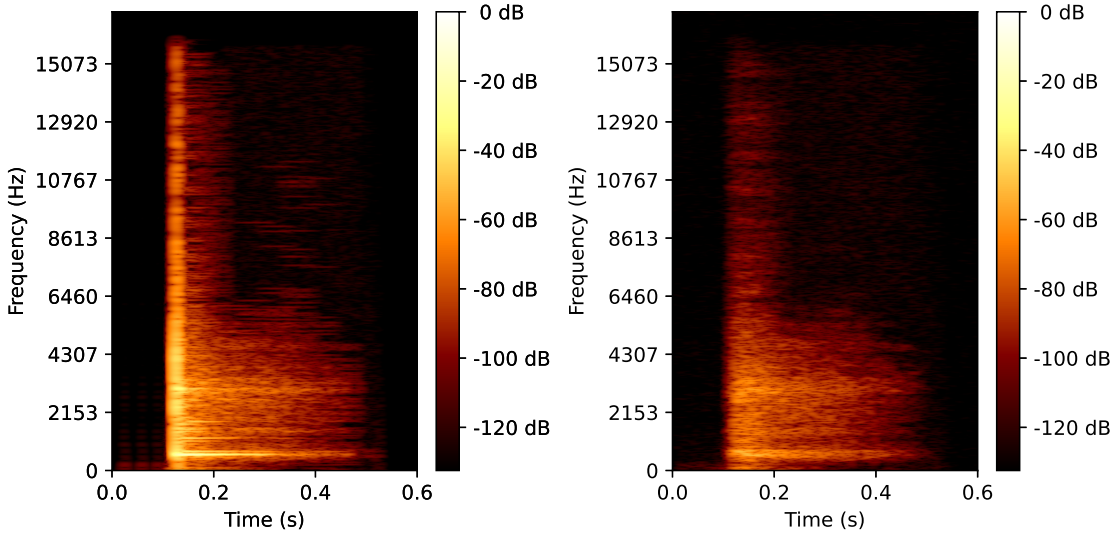
$$C = \{(i-2, j-2) \in \mathbb{Z}^2 : a_{ij} = 1\} \tag{3.10}$$

$$D = \{(i-2, j-2) \in \mathbb{Z}^2 : a_{ij} = 0\} \tag{3.11}$$

This vertical thinning process transforms the ridges of the input into lines of one-pixel thickness. However, as seen in Figure 3.11, these lines cannot be directly used to obtain our parameters. We still need to remove the background information to obtain precise lines. This is achieved by using the top-hat operation.

Vertical top-hat

To isolate the lines and remove the background, we apply a top-hat operation to the thinned image. Top-hat is explained in Section 1.2.4.3. Since our objective is to retrieve the horizontal lines, we use a vertical top-hat by using a structuring element



(a) Spectrogram of the input

(b) Spectrogram of the filtered noise

Figure 3.10: Result of extracting the stochastic component of the signal.

with a size of 1×3 pixels (1 pixel in time and 3 pixels in frequency). The output of this process is shown in Figure 3.12.

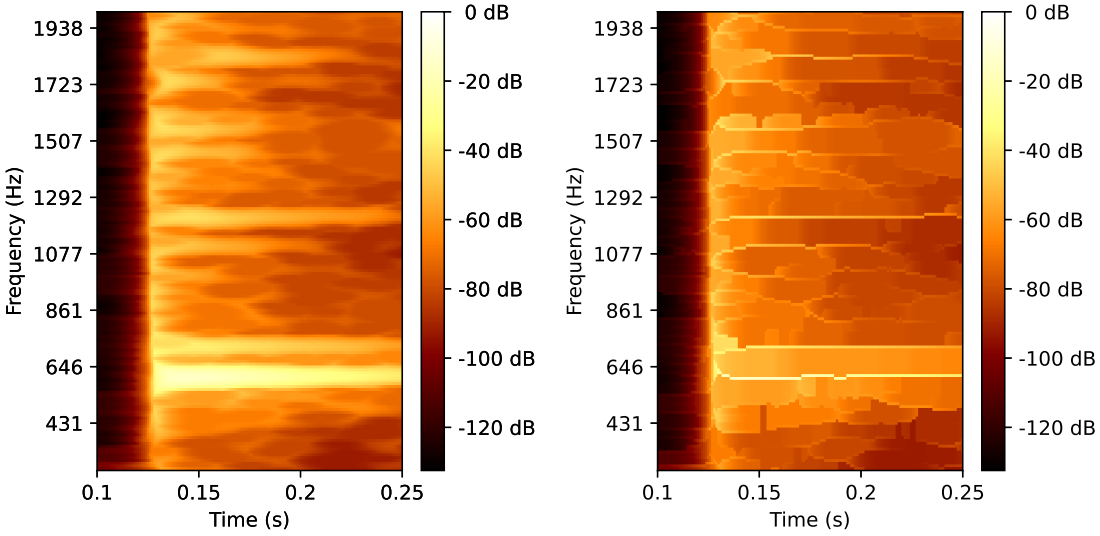
Threshold vertical top-hat

While horizontal lines appear neat in the top-hat image, various artifacts arise due to the nature of spectrograms. To mitigate some of these artifacts[5] we apply a threshold. Moreover, this threshold operation is done on the reconstruction by erosion but with the values of the top-hat i.e., if we denote the output of the threshold by $S_>$, the reconstruction by erosion as S_0 and the result of the top-hat as $S_{\mathbf{Id}-\gamma}$, we have

$$S_>(\tau,\omega) = \begin{cases} S_0(\tau,\omega) & \text{if } S_{\mathbf{Id}-\gamma}(\tau,\omega) > \tau_\mathrm{v} \\ -\infty & \text{if } S_{\mathbf{Id}-\gamma}(\tau,\omega) \leq \tau_\mathrm{v} \end{cases} \tag{3.12}$$

with τ_v being the threshold for the vertical top-hat. The value we chose is $\tau_\mathrm{v} = 5\,\mathrm{dB}$. The result is shown in Figure 3.13.

[5]It is very difficult to remove all of them, if not impossible, in part because the border between signal and noise is not always well defined.



(a) Reconstruction by erosion

(b) Vertical thinning

Figure 3.11: Vertical thinning transforming the ridges into one-pixel-thick lines.

Filter small horizontal lines

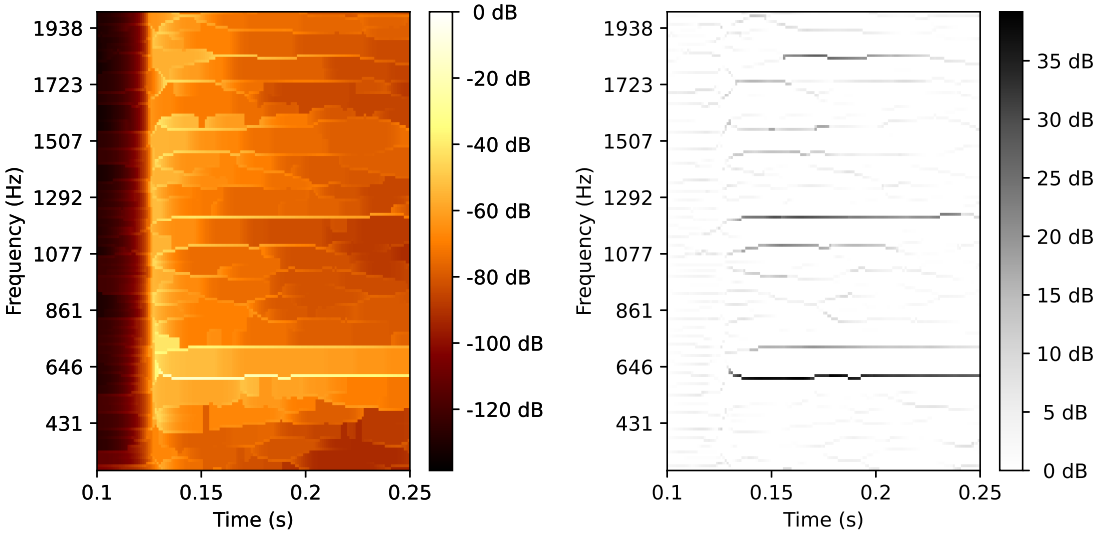
This step is intended to eliminate lines that are too small to be considered genuine sinusoids and are more likely to be artifacts. While these lines might represent actual signals, they tend to contribute less to the desired output and can introduce unwanted sounds that are perceived as artifacts.

To address this issue, we employ a two-step processing approach. First, we shrink the lines that are below a certain length threshold, causing them to disappear if they are too small. Then, we use a reconstruction by dilation to recover the parts that were previously shrunk.

For shrinking the lines, we remove the west and east points by applying a thinning that uses the following patterns:

$$C_W = \begin{bmatrix} 0 & - & - \\ 0 & 1 & - \\ 0 & - & - \end{bmatrix} \qquad C_E = \begin{bmatrix} - & - & 0 \\ - & 1 & 0 \\ - & - & 0 \end{bmatrix}. \tag{3.13}$$

Following the shrinking operation, we perform a reconstruction by dilation. We use the shrunken image as the marker and the output of the threshold as the mask. The minimum length that we allow for a line to be retained serves as a parameter



(a) Vertical thinning

(b) Vertical top-hat

Figure 3.12: Vertical top-hat for recovering the horizontal lines.

for this process. In this case, a minimum length of 100 ms was chosen. The result of this operation is shown in Figure 3.14.

Retrieving the parameters for sinusoids

The last image of the process serves as input to for parameter recovery. This recovery is not a processing in itself, but rather a "transducer": it transforms an image into a list of parameters. The process works as follows:

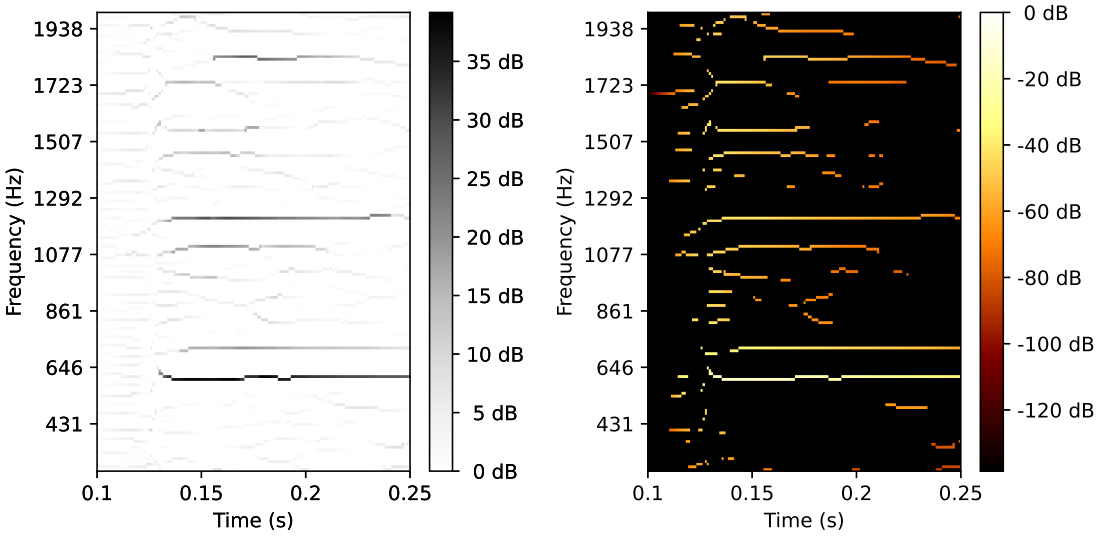
1. We recover the I connected components, representing individual lines, using the SciPy (Virtanen et al., 2020) library's functionality for this purpose.

2. For each component indexed by i, we create an array $\{(t_m^i, \xi_m^i, S_O(t_m^i, \xi_m^i)\}_{m=1}^{M_i}$ where S_O is the output of our morphological pipeline for sinusoids.

3. We sort the array with respect to the time.

This approach yields i arrays, each corresponding to a sinusoidal component. However, two potential issues arise if we synthesize directly from these arrays:

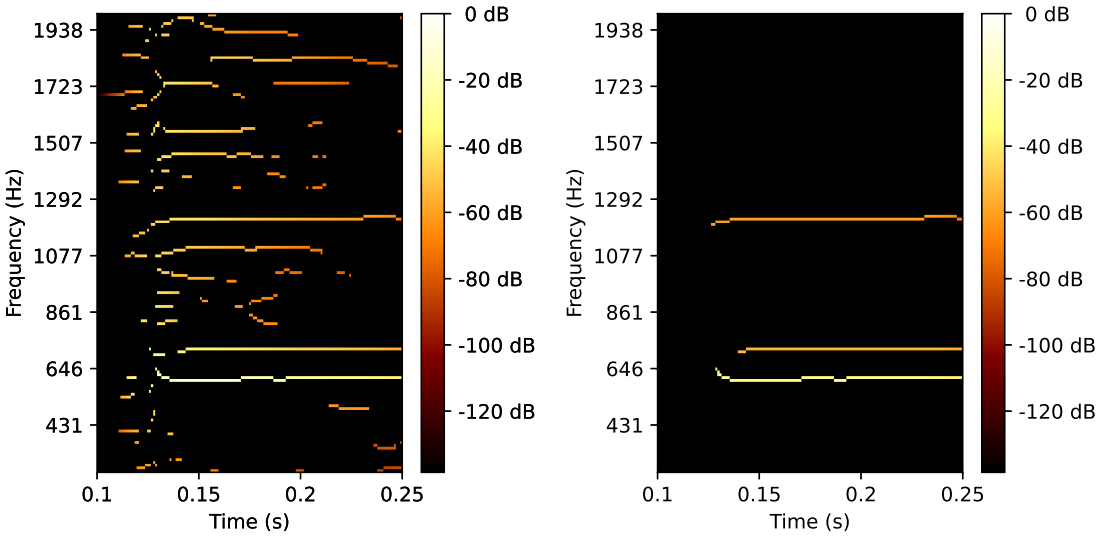
1. Multiple branches for each line may exist, leading to multiple ξ values for the same t.



(a) Top-hat

(b) Threshold

Figure 3.13: Threshold of 5 dB of the vertical top-hat.



(a) Threshold of the top-hat

(b) Thinning and reconstruction by dilation

Figure 3.14: Removal of the small lines with a thinning followed by a reconstruction by dilation.

2. Working with images composed of pixels implies frequency quantization, resulting in significant steps that could generate artifacts, particularly in lower frequencies.

To address both issues simultaneously, we apply a filter to the array of frequencies. This smooths the frequencies and eliminates artifacts. Specifically, we use a Butterworth filter (Butterworth, 1930) of order 3, with a critical frequency set to 0.05 times of the Nyquist frequency[6]. To avoid border problems, we employ Gustafsson's method (Gustafsson, 1996). The result is shown in Figure 3.15.

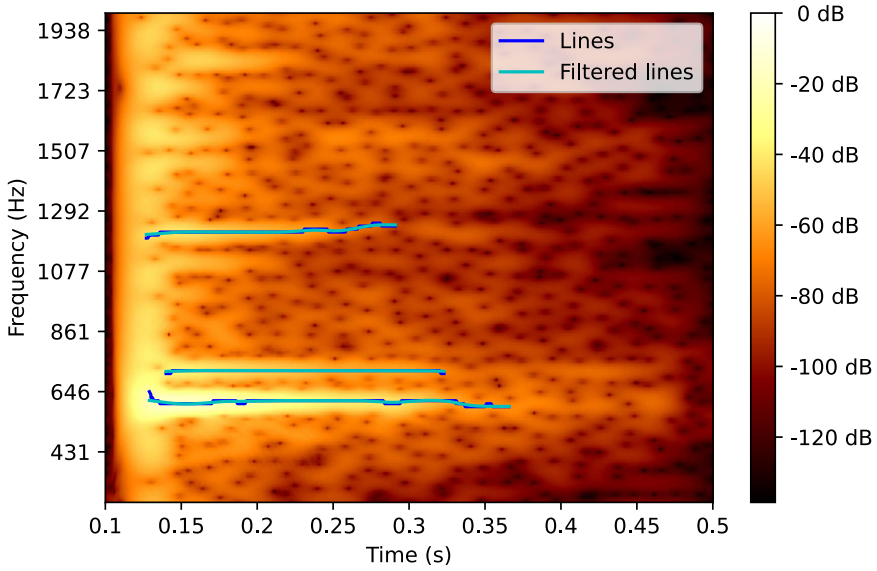


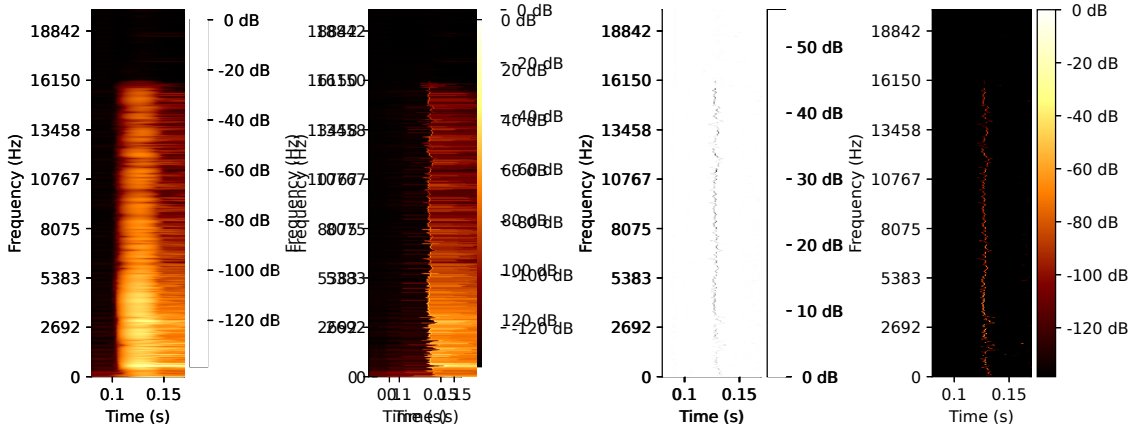
Figure 3.15: Lines and filtered lines recovered in the process superimposed to the input spectrogram.

3.2.2.4 Processing for transient component

The approach used for retrieving transient parameters is a dual of the one employed for sinusoids, with the exception that the reconstruction by erosion itself (not its eroded version) is used. The corresponding steps are as follows:

[6]In our example, since the time resolution for the array $\boldsymbol{\xi}$ is 0.001 s, the sampling frequency is 1000 Hz and the Nyquist frequency is 500 Hz, which gives a critical frequency of $0.05 \times 500 = 25$ Hz.

- **Horizontal thinning**: instead of a vertical thinning we use an horizontal thinning, with the templates corresponding to the east, west, north-west, south-east, north-east, south-west points. The result is shown in Figure 3.16b.
- **Horizontal top-hat**: instead of a vertical top-hat, we employ an horizontal top-hat operation, with a structuring element of size 3×1 pixels (3 in time and 1 in frequency). The result is shown in Figure 3.16c.
- **Threshold**: the threshold operation is performed on the output of the horizontal top-hat, with the same threshold value $\tau_{\mathrm{v}} = 5\,\mathrm{dB}$. The result is shown in Figure 3.16d.
- **Filter small vertical lines**: we use the same approach as before, but with a vertical thinning (using templates C_N and C_S) and with minimal length we allow being 100 Hz. However, this step did not affect the image in this case as there are two long lines.
- **Retrieving the parameters for transient**: the process for retrieving the lines and applying the filter (in this case to the times array) is the same as used for sinusoids. The result is shown in Figure 3.17.



(a) Input (b) Horizontal thinning (c) Horizontal top-hat (d) Threshold

Figure 3.16: Morphological steps for recovering the transient lines.

Once we recover the lines, we use the method exposed in Section 3.1.3 to generate a transient. The comparison between the input and the spectrogram of the generated

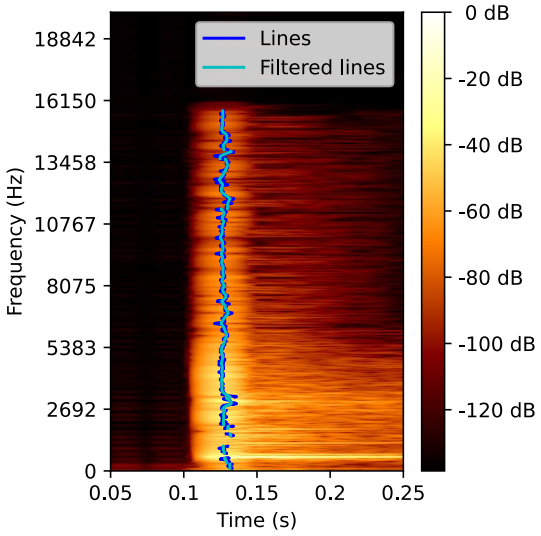


Figure 3.17: Lines and filtered lines recovered in the process superimposed to the input spectrogram.

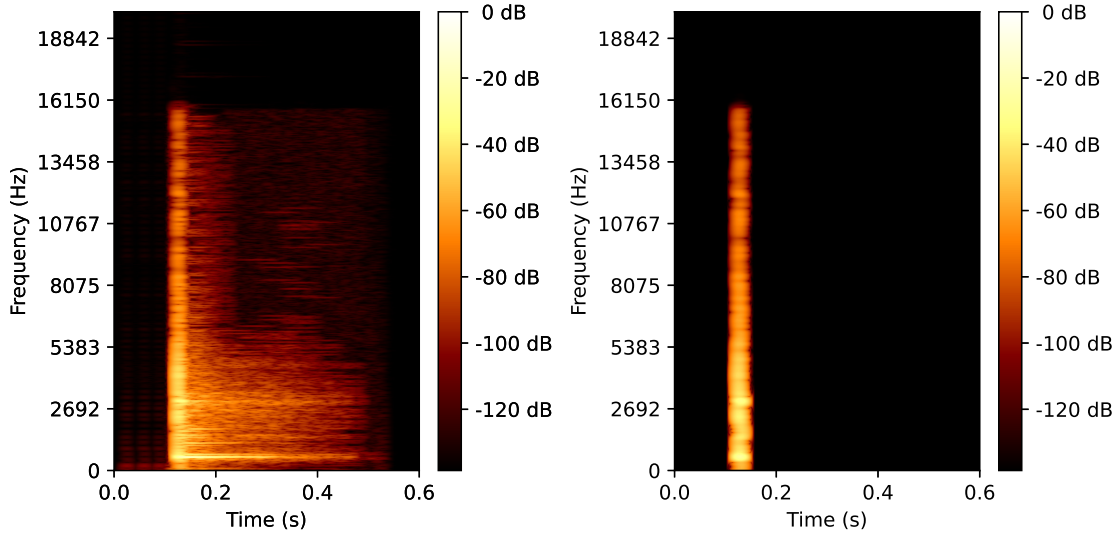
transient is shown in Figure 3.18. We see that the shape of the transient is very similar to the input one.

3.3 Application to Music Instruments

In this section, we evaluate the performance of the proposed method by applying it to different musical instruments. We use the same set of parameters and test the method on sounds produced by various instruments: marimba (with a pronounced transient component), violin (with a prominent sinusoidal component), gong (primarily consisting of a noise component), and piano (featuring a balanced combination of three components).

All of the sounds used in this chapter are sourced from the University of Iowa Musical Instrument Samples[7] and Studio-On-Line Database (Ballet et al., 1999) sound libraries.

[7]https://theremin.music.uiowa.edu/MIS.html.



(a) Spectrogram of the input

(b) Spectrogram of the generated transient

Figure 3.18: Comparison between the spectrograms of the input and the generated transient.

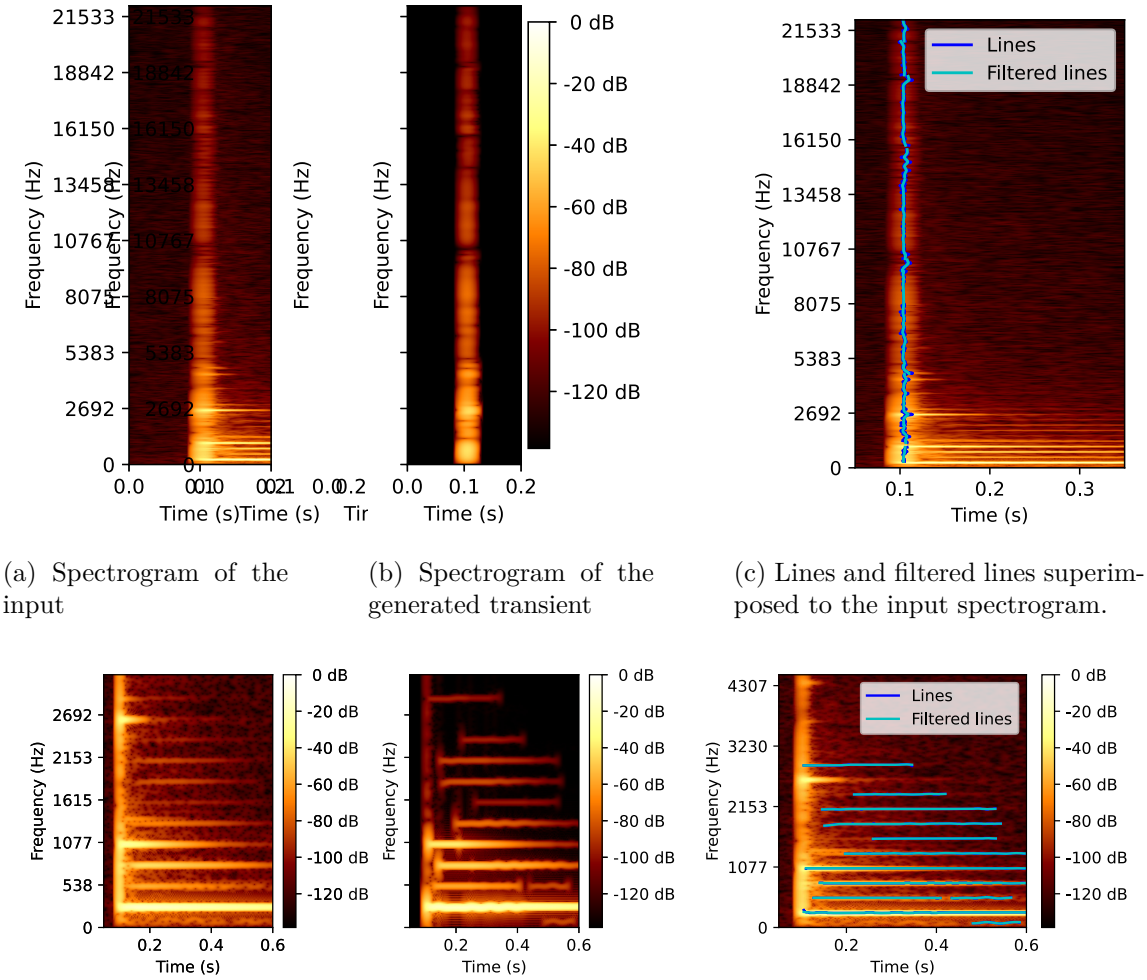
3.3.1 Marimba

The marimba sound features a significant transient part and also a notable sinusoidal component. The processing results are displayed in Figure 3.19. The transient part has been recovered with high fidelity. However, there are some challenges in the recovery of the sinusoidal component.

One major issue is the failure to capture an important sine wave with a frequency of around 2600 Hz. This is a critical concern as, despite its brevity, the sine wave is clearly intense and prominent, as shown in the spectrogram. The problem occurred due to the threshold operation, where the sine was split into two parts, both of which were subsequently suppressed because they were too small.

Another issue is the presence of interference between two sine waves around 1300 Hz causing both of them to go undetected. Additionally, interference effects are the cause of the break of the sine wave with a frequency of around 500 Hz.

Overall, the retrieval of the sinusoidal component from the marimba sound is considered to be of mediocre quality.



(a) Spectrogram of the input

(b) Spectrogram of the generated transient

(c) Lines and filtered lines superimposed to the input spectrogram.

(d) Spectrogram of the input

(e) Spectrogram of the generated sinusoids

(f) Lines and filtered lines superimposed to the input spectrogram.

Figure 3.19: Marimba.

3.3.2 Violin

The results from the violin sound exhibit improvements over those from the marimba, as demonstrated in Figure 3.20. Multiple sinusoidal components were successfully detected, and the noise component generated by the bowing of the string was accurately recovered. The input and output spectrograms closely resemble each other.

However, a notable limitation of the process is related to the attack of the sound. In the case of the violin, particularly when the attack is strong, it produces a crackling sound rather than a traditional transient. This unique nature of the attack prevents its simulation through a transient component, resulting in an unsuccessful recovery using the noise component.

The method was also applied to a violin sound featuring vibrato, and the results are depicted in Figure 3.21. The majority of the lines were reasonably recovered, even if they are not straight. Some lines, due to their brevity and lack of connection to others, were not recovered. Additionally, interference, this time with the noise, led to the disconnection of certain lines.

3.3.3 Gong

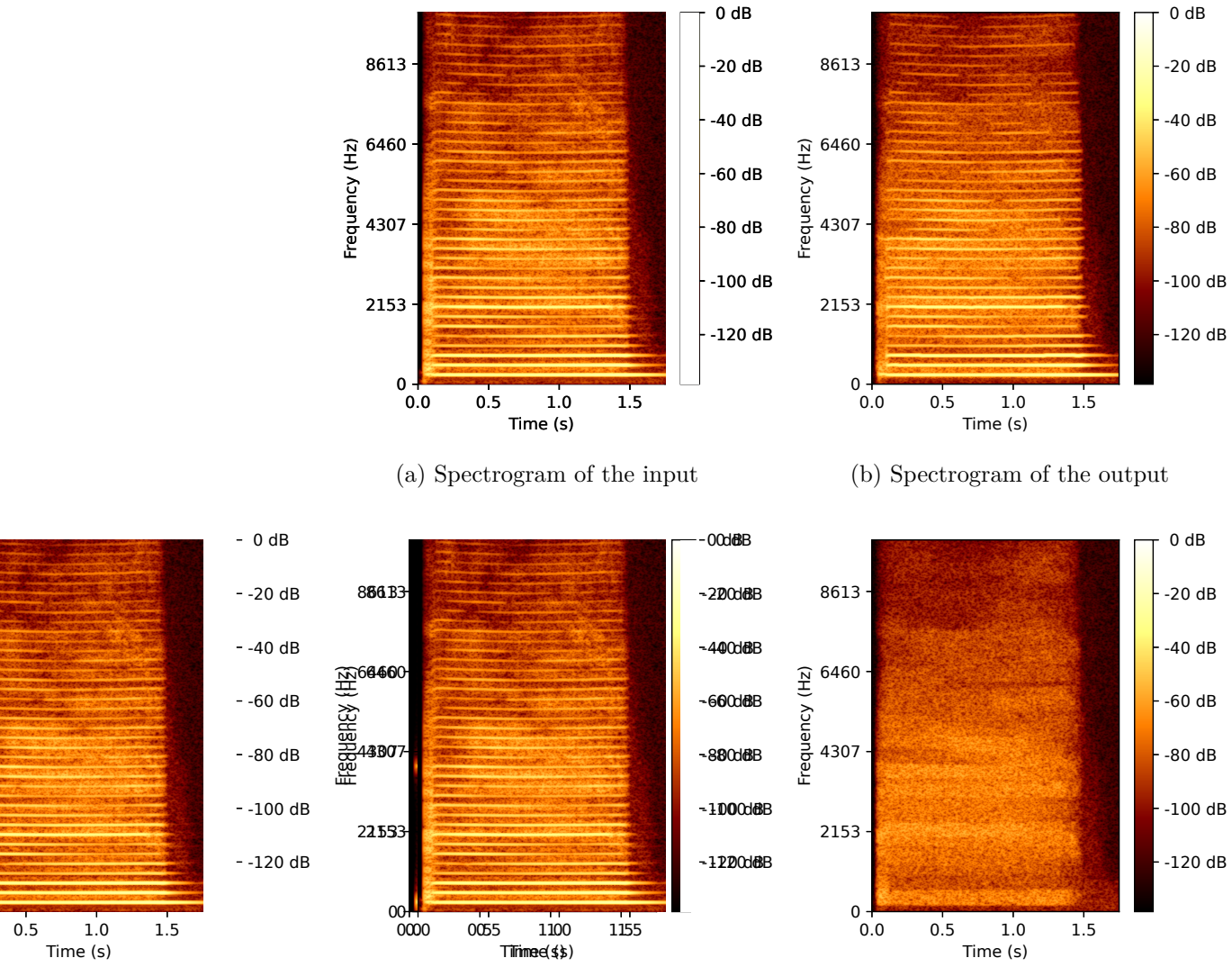
Among the tested instruments, the gong showcases the best results. The noise component is accurately recovered, demonstrating high fidelity in reproducing the original noise characteristics. While some minor sinusoidal components are present, their influence on the output is largely masked by the dominant noise component. However, the fundamental bass sine, the only sine that is clearly perceptible, is successfully recovered. The transient elements, although not extensively pronounced in the gong sound, are still moderately captured by the method.

3.3.4 Piano

The piano results are probably the more disappointing: while the sinusoidal and noise components are accurately recovered, the transient component retrieval is notably inadequate. This outcome contradicts initial expectations, given that the piano sound is renowned for its significant transient component during its attack.

Despite this unexpected result, several insights might shed light on the situation. Close examination of the piano spectrogram reveals that the transient is not present as a distinct, isolated vertical line, as is typical in some other cases. Instead, there exist small vertical lines at the onset of each sine wave. These shorter vertical lines correspond to high variations of the amplitude of the sine, experiencing spectral leakage.

This observation prompts the question of whether the transient component alone is sufficient or if it should be use as a sub-parameter of the sinusoidal component, or perhaps even use a hybrid approach tailored to each specific case.



(a) Spectrogram of the input

(b) Spectrogram of the output

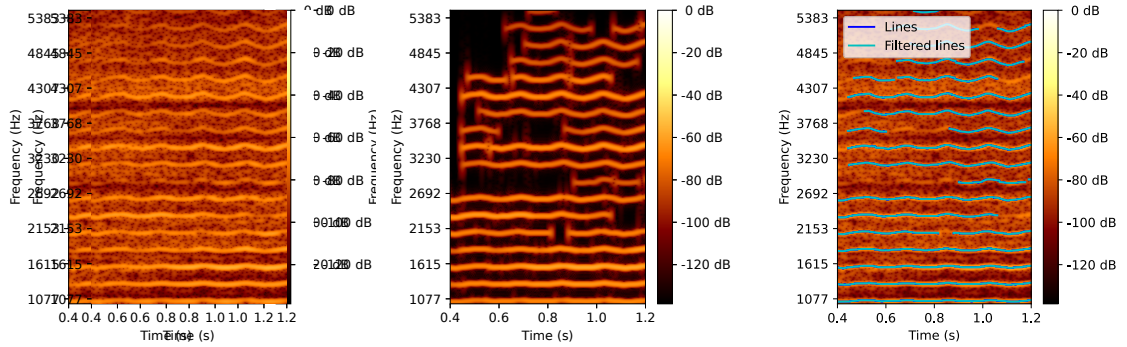
(c) Spectrogram of the sinusoids

(d) Spectrogram of the noise

Figure 3.20: Violin.

3.4 Discussion and Conclusion

Throughout this chapter, we have explored how MM can be applied to the analysis of spectrograms of music instrument sounds. These sounds often exhibit distinct ge-



(a) Spectrogram of the input (b) Spectrogram of the sinusoids (c) Spectrogram and lines superimposed

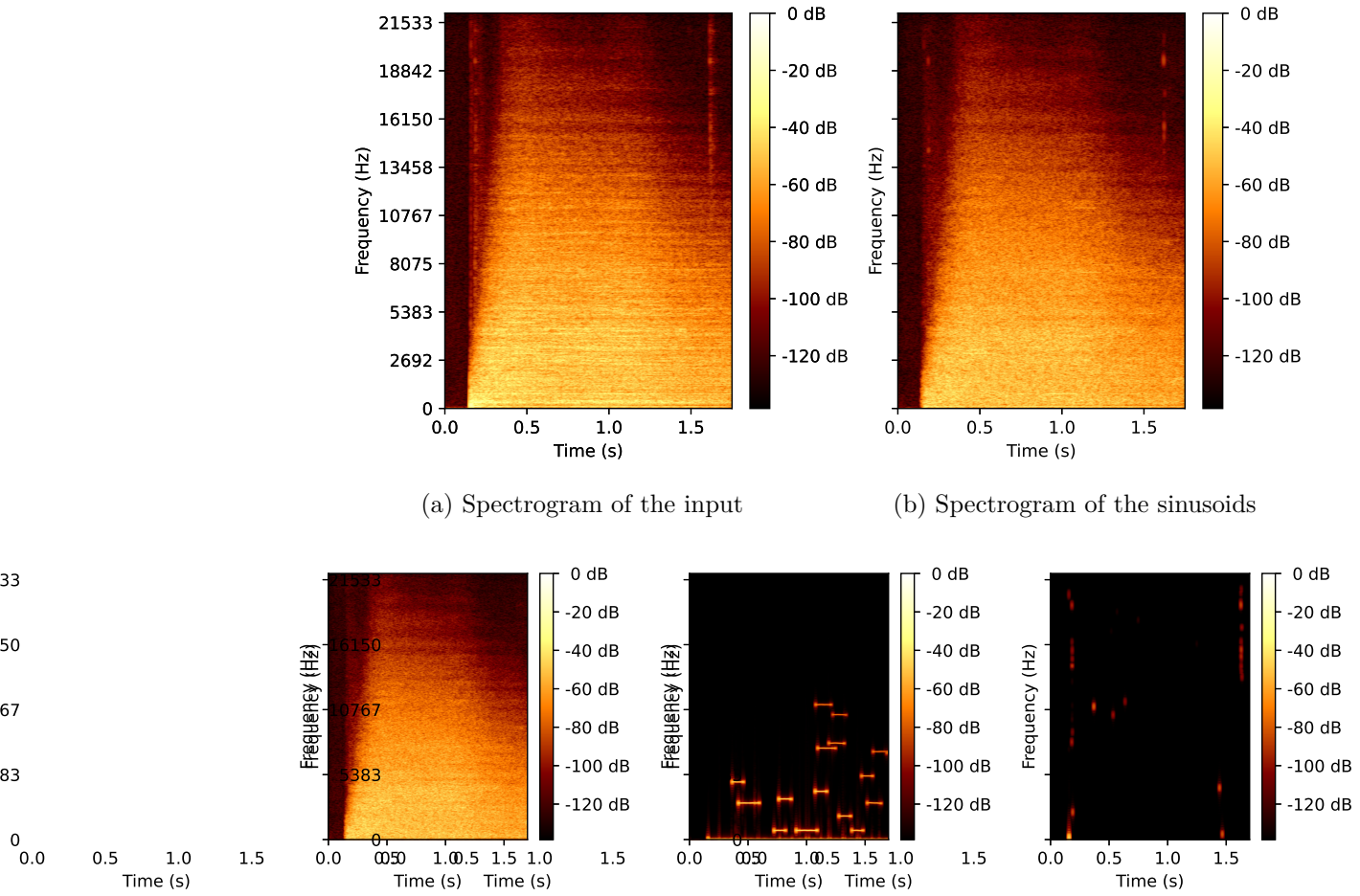
Figure 3.21: Violin vibrato.

ometric patterns, such as lines and holes, making MM well-suited for their detection.

However, the performance of the proposed method falls short of our expectations. While it does reasonably well in detecting the desired geometric patterns, it lacks the robustness required for such signal processing applications. This limitation is evident from the impact of the thresholding process and the presence of interferences. The method excels in recovering the noise component, which involves fewer steps. Nonetheless, it is important to acknowledge that the core of noise component recovery lies in the reconstruction by erosion, a computationally expensive operation that demands several seconds of GPU computation for only a few seconds of sound.

Many of the challenges in our method are not exclusively rooted in the domain of mathematical morphology. The shortcomings also extend to the STN synthesis model. This model, while conceptually appealing, does not perform as well as expected. In particular, the transient generation has notable issues. For instance, the synthesized signals are symmetric, an unrealistic feature in musical sounds. Additionally, the lack of coherent synchronization between transients and sines results in audible discrepancies.

It is important to note that the distinction between the signal and the noise can often be ambiguous, with each potentially obscuring the other and yielding undesirable outcomes. When the signal overshadows the noise, the situation is less problematic since the opening operation effectively transforms it into noise, preserving the coherence of the noise component. However, when the noise masks the signal, it results in fragmented lines with fading effects to manage, as well as the issue of small lines that might be overlooked.



(a) Spectrogram of the input

(b) Spectrogram of the sinusoids

(c) Spectrogram of the noise

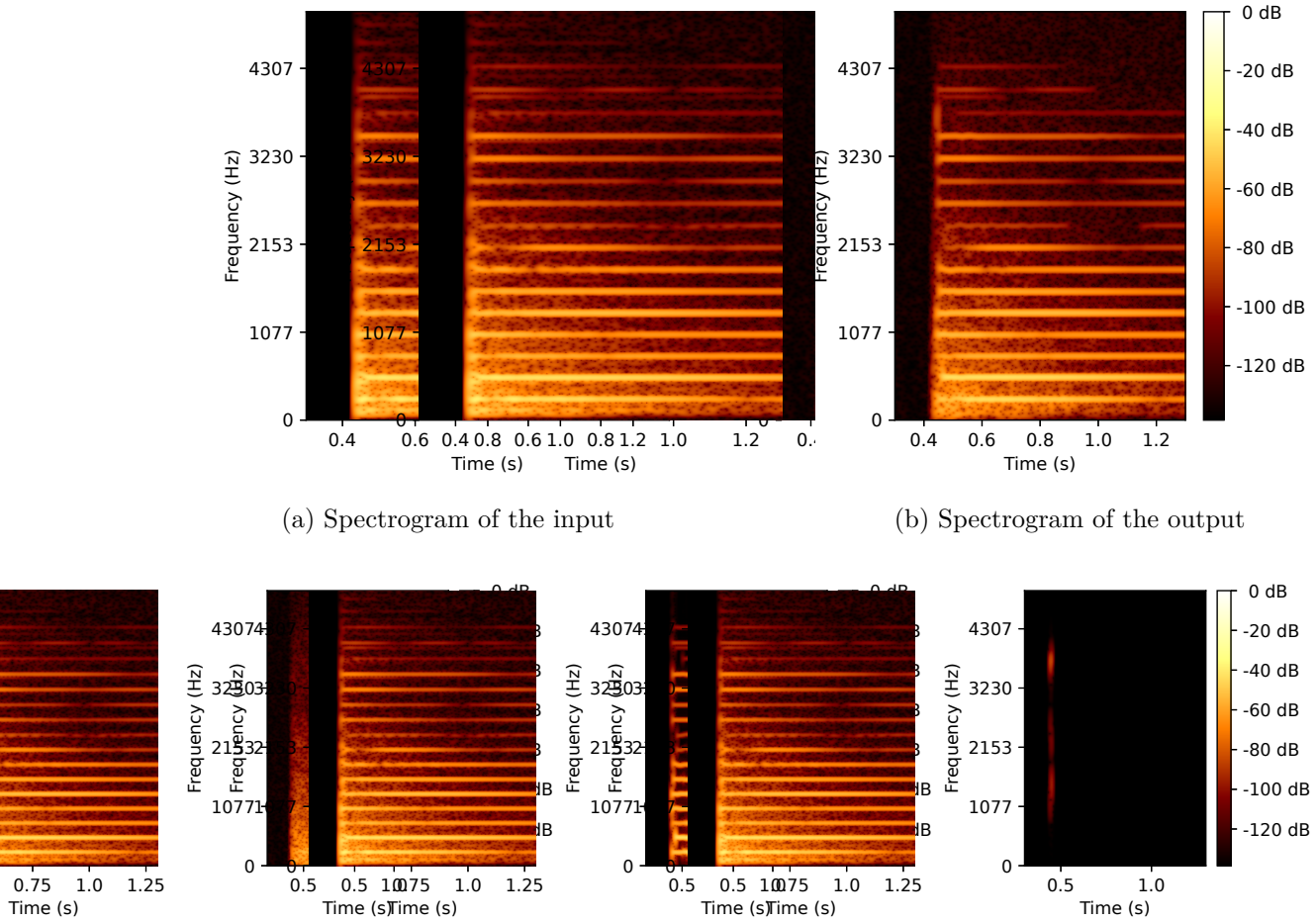
(d) Spectrogram of the output

(e) Spectrogram of the output

Figure 3.22: Gong.

In conclusion, we believe that significant room for improvement exists both in the MM pipeline itself to enhance its robustness and in the synthesis method to achieve smoother component integration.

Additionally, we acknowledge the significant influence that parameter tuning can have on the outcomes. This emphasizes the importance of incorporating human expertise into the process. While this might pose challenges for full automation of the pipeline, it presents an opportunity for meaningful collaboration between humans and machines.



(a) Spectrogram of the input

(b) Spectrogram of the output

(c) Spectrogram of the noise

(d) Spectrogram of the sinusoids

(e) Spectrogram of the transient

Figure 3.23: Piano.

Specifically, we envision the development of a desktop application that empowers users to interact with the system. Users could select specific regions of a spectrogram and use MM techniques to effectively separate noise from the desired signal. This interactive approach would not only harness the strengths of mathematical morphology but also leverage human intuition and domain knowledge to enhance the overall accuracy and quality of the results.

Chapter 4

Mathematical Morphology Applied to Generate Piano Rolls

In the previous chapter, we have shown how to use MM to analyze spectrograms. In this chapter and the following, we apply morphological operators to another kind of time-frequency representation of music: piano rolls. Piano rolls have been exposed in Section 2.3 as a useful representation of MIDI files and scores. In this chapter, we focus on generating music in this format through the use of MM.

In the following, we use the notations exposed in Chapter 2. In particular, we consider the complete lattice $(\mathcal{A}^{\mathcal{T}\times\mathcal{F}}, \preceq)$, where $\mathcal{T} \times \mathcal{F}$ is a time-frequency space with $(G_{\mathcal{T}\times\mathcal{F}}, +)$ a group acting on it, and $(\mathcal{A}, \leq)$ is one of the amplitude ranges exposed in Section 2.1.3, endowing $\mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ with a structure of complete lattice given by the pointwise order. With this structure, we can apply MM based on structuring elements as exposed in Section 1.2.

The choices of $\mathcal{T}$, $\mathcal{F}$ and $\mathcal{A}$ may vary depending on the specific cases, as exposed in Chapter 2. The selection of $\mathcal{T}$ depends on whether the input is derived from a MIDI file or a musical score. $\mathcal{F}$ will be either $\mathcal{N}$ or $\mathcal{N}_{12}$. While the amplitude range might also differ, for the sake of simplicity, we use $\mathcal{A}_3 = \{\bot, \cdot, \times\}$ in the examples provided. Additionally, we employ the Boolean lattice $\mathcal{A}_2 = \{0, 1\}$, which results in a residuated triplet through the lattice multiplication operation $\bullet : \mathcal{A}_3 \times \mathcal{A}_2 \to \mathcal{A}_3$ exposed[1] in Section 2.1.3.2.

The chapter is structured as follows: in Section 4.1, we begin by establishing the definitions of *texture* and *harmony*, which we subsequently employ to construct a concept that we call *harmonic texture*. Then, in Section 4.2, we expose the process

[1]It is worth noting that we have modified the order of the inputs of $\bullet$ for consistency with the subsequent notations.

of generating piano rolls using MM and harmonic textures.

4.1 Texture and Harmony

The main contribution of this chapter is the establishment of a framework for the generation of music based on two concepts borrowed from music theory: *texture* and *harmony*. While these terms might exhibit varied and sometimes vague definitions across different contexts, our objective here is to provide precise mathematical definitions tailored to our specific objectives.

4.1.1 Texture

The term *texture* holds various interpretations across different domains. In the realm of music, the concept of texture lacks a universally agreed-upon definition (Moreira, 2019). In (Couturier et al., 2022b), two distinct yet interconnected (Herold, 2012) interpretations of the term texture have been identified.

The first interpretation is the *orchestral texture*, which concerns the timbral interactions between musical instruments. This aspect has been extensively studied in orchestration from the 19^{th} century (Berlioz, 1844; Piston, 1955; Nordgren, 1960; Guigue & de Paiva Santana, 2018).

The other interpretation, known as *symbolic texture*, has received less attention, possibly due to its elusive nature. However, it is this latter interpretation that forms the basis of our discussion, and we aim to provide a formal definition that captures its essence. In recent years, there has been a growing interest in investigating symbolic texture (Giraud et al., 2014; Parada-Cabaleiro et al., 2021; Soum-Fontez et al., 2021; Couturier et al., 2022a, 2022b).

To formulate our definition of texture, we draw upon the concept of *rhythm*. We provide a custom definition of rhythm for our framework, but we will see later that it is fairly compatible with the definition of rhythm based on trees (Agon et al., 2002) that is used in Computer Assisted Composition (Jacquemard et al., 2015; Jacquemard et al., 2017; Ycart et al., 2016).

Definition 4.1 (Rhythm)**.** We say that $R \in \mathcal{A}_3^{\mathbb{Q}}$ is a **rhythm** if $\exists N \in \mathbb{N}$ such that

$$R = \bigvee_{n=1}^{N} h_n \tag{4.1}$$

where the $h_n \in \mathcal{A}_3^{\mathbb{Q}}$ are such that $\exists s_n \in \mathbb{Q}$, $\exists d_n \in \mathbb{Q}^+$:

$$h_n : \begin{array}{ccc} \mathbb{Q} & \to & \mathcal{A}_3 \\ t & \mapsto & h_n(t) = \begin{cases} \times & \text{if } t = s_n \\ \cdot & \text{if } t \in]s_n, s_n + d_n[\\ \perp & \text{if } t \notin [s_n, s_n + d_n[\end{cases} \end{array}$$

and $h_n \wedge h_{n'} = \perp$ if $n \neq n'$.

Each h_n is called a *hit* and has a start s_n and a duration[2] d_n. We notate $h_n \equiv (s_n, d_n)$ and $R \equiv \{(s_n, d_n)\}_{n=1}^N$ for simplicity.

This definition links the notion of rhythm with the notion of *time span* that is presented in (Lewin, 1987); each hit is equivalent to a time span, and a rhythm is given by a set of hits.

In order to include also the possibility of a rhythm to be defined over a time space $\mathcal{T}$, we include the definition of a *placed rhythm*.

Definition 4.2 (Placed rhythm)**.** Let $\mathcal{T}$ be a time space and $\iota : \mathcal{T} \to \mathbb{Q}$ a function[3].

We call $R_0 \in \mathcal{A}_3^{\mathcal{T}}$ a **placed rhythm** if $\exists t_0 \in \mathcal{T}$, $\exists R \in \mathcal{A}_3^{\mathbb{Q}}$ such that

$$R_0 = t_0 + R : \begin{array}{ccc} \mathcal{T} & \to & \mathcal{A}_3 \\ t & \mapsto & R_0(t) = R(t - t_0) := R(\iota(t) + (-\iota(t_0))) \end{array} . \tag{4.2}$$

These definitions might appear intricate for rhythms, but they represent a notion of rhythm that can handle previous mathematical formalizations like the one exposed in (Toussaint, 2013) and the one proposed in (Agon et al., 2002) (besides the hierarchical component).

To be able to simplify the representation of rhythms, we are showing that a rhythm can be represented by a vector.

Let R be a rhythm. We are creating a vector $\boldsymbol{R} \in \mathcal{A}_3^M$ for some $M \in \mathbb{N}$ that represents this rhythm. We know that R can be written as

$$R = \bigvee_{n=1}^{N} h_n$$

[2]Notice that the duration may be zero, which is allowed for representing percussive rhythms where the duration is not featured.

[3]The requirement for a function $\iota : \mathcal{T} \to \mathbb{Q}$ arises from the need to establish an association between an element $t_0 \in \mathcal{T}$ and an element $\iota(t_0) \in \mathbb{Q}$, thereby enabling the definition of the additive inverse of t_0, denoted as $-t_0 := -\iota(t_0)$.

with h_n disjoint hit functions with corresponding starts and durations s_n and d_n, respectively. We call $e_n = s_n + d_n$ the ends.

Without loss of generality, we can order these functions such that $n < m \Rightarrow s_n < s_m$, having additionally that $n < m \Rightarrow e_n \leq s_m$ due to the h_n being disjoint (notice that they may be equal).

There is a single degenerated case that we will not address: the case where $N = 1$ and $h_1 \equiv (0, 0)$. In the other cases, let $a \in \mathbb{Q}^+$ be the greatest common divisor (GCD) in the sense of rationals[4] of the set $\{s_n, e_n\}_{n=1}^N$, which is commonly called in music the *tatum*[5].

With a, we can rewrite all the s_n, e_n as m_n^s, m_n^e with $s_n = m_n^s \cdot a$ and $e_n = m_n^e \cdot a$. Now, let

$$m_0 = \min\{m_n^s, m_n^e\}_{n=1}^N \qquad\qquad m_1 = \max\{m_n^s, m_n^e\}_{n=1}^N$$

and $M = m_1 - m_0 \in \mathbb{N}$ if $d_N \neq 0$ and $M = m_1 - m_0 + 1 \in \mathbb{N}$ if $d_N = 0$. We give then the following values to each element of the vector $\boldsymbol{R}$:

$$\forall m \in \{0, 1, ..., M-1\},\ \boldsymbol{R}_m = \bigvee_{t \in [t_0, t_1[} R(t) \tag{4.3}$$

with $t_0 = (m + m_0) \cdot a \in \mathbb{Q}$ and $t_1 = (m + m_0 + 1) \cdot a \in \mathbb{Q}$.

Let us illustrate this method with an example.

Example 4.3. *Let us consider the rhythm* 𝅘𝅥𝅯𝅘𝅥𝅯|𝅘𝅥𝅯𝅀𝅘𝅥 *where the vertical bar indicates the place of the 0. This rhythm is illustrated in Figure 4.1.*

It can be represented by the rhythm

$$\begin{aligned} R &\equiv \{(s_1, d_1), (s_2, d_2), (s_3, d_3), (s_4, d_4)\} \\ &\equiv \left\{\left(-\frac{1}{4}, \frac{1}{8}\right), \left(-\frac{1}{8}, \frac{1}{8}\right), \left(0, \frac{1}{4}\right), \left(\frac{1}{2}, \frac{1}{4}\right)\right\}. \end{aligned} \tag{4.4}$$

The set $\{s_n, e_n\}_{n=1}^4$ *is*

$$\left\{-\frac{1}{4}, -\frac{1}{8}, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}\right\}$$

[4]That is, $a \in \mathbb{Q}^+$ is the GCD of $p \in \mathbb{Q}$ and $q \in \mathbb{Q}$ if $\exists m_p, m_q \in \mathbb{Z}$ such that $p = a \cdot m_p$, $q = a \cdot m_q$, and a is the greatest rational that satisfies this property.

[5]This notion, whose name was given by Bilmes, 1993, has been studied in (Romero-García, Lascabettes, et al., 2022; Romero-García, Guichaoua, et al., 2022).

with the GCD being $a = \frac{1}{8}$. *We can rewrite then*

$$s_1 = -2 \qquad s_2 = -1 \qquad s_3 = 0 \qquad s_4 = 4$$
$$e_1 = -1 \qquad e_2 = 0 \qquad e_3 = 2 \qquad e_4 = 6$$

and have $m_0 = -2$, $m_1 = 6$ *and* $M = 6 - (-2) = 8$.

The resulting vector is given by

$$\boldsymbol{R} = (\times, \times, \times, \cdot, \perp, \perp, \times, \cdot)\,.$$

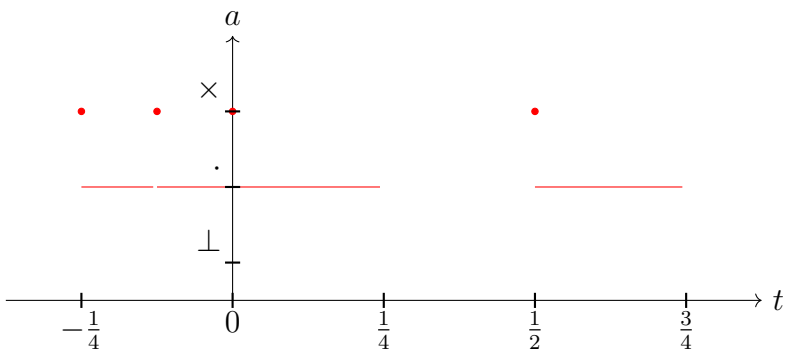
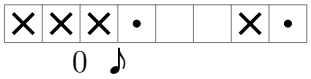


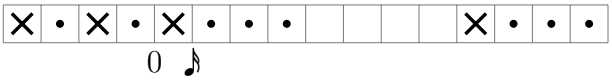
Figure 4.1: Representation of the rhythm 𝅘𝅥𝅯𝅘𝅥𝅯|𝅘𝅥𝅮𝄽𝅘𝅥𝅮 by the function from Equation (4.4).

In the following, we will plot such a vector as



where we left the square empty for ⊥ and specify the place of the 0 and the tatum.

Notice that we used the GCD (the tatum) to quantize the rhythm, but that we might have use a divisor of the tatum. In the case of the Example 4.3, have we set the value of a to 𝅘𝅥𝅯 $= \frac{1}{16}$ and we would have get the vector



Notably, we are capable of converting any rhythm tree into a rhythm. In (Jacquemard et al., 2015), it is demonstrated that each rhythm tree possesses an associated

duration sequence. From this sequence, we can deduce the sequence of hits that correspond to a rhythm in accordance with our definition[6]. This transformation does involve a loss of the hierarchical structure inherent in rhythm trees, yet it is important to acknowledge that this hierarchical nature cannot be represented within the piano roll model.

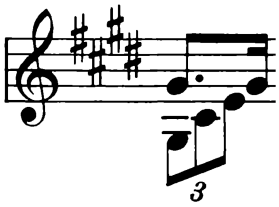
Now, let us introduce our definition of texture.

Definition 4.4 (Texture)**.** A **texture** T is a countable tuple of rhythms, i.e., $T = (R_i)_{i\in I} \in \mathcal{R}^I \subset \mathcal{A}_3^{\mathcal{T}\times I}$, where I is a countable set. A finite texture of size $|I|$ is a texture such that $|I| < \infty$.

This definition might not immediately appear closely related to the symbolic texture, but to give it meaning, we cannot separate it from the concept of *harmony*. This concept will be exposed in detail in the following section, but before let us show that a finite texture can be represented by a matrix.

Indeed, if $T = (R_i)_{i\in I}$ is a finite texture, we can represent each of its rhythms by a vector by choosing the tatum of each of them. We choose the tatum of all of them to make them quantized on the same grid, and then we stack them in a matrix in an order related to I. We show that in the following example.

Example 4.5. *Let us consider the rhythms issued from the excerpt*



from the first movement of Beethoven's Piano Sonata No.14, Op.27 No.2.

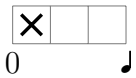
We may associate a rhythm for each pitch in the following way:

$$R_1 = \left\{\left(0, \frac{1}{12}\right)\right\} \text{ for } \mathrm{G\sharp 3} \qquad R_2 = \left\{\left(\frac{1}{12}, \frac{1}{12}\right)\right\} \text{ for } \mathrm{C\sharp 4}$$

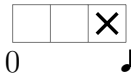
$$R_3 = \left\{\left(\frac{2}{12}, \frac{1}{12}\right)\right\} \text{ for } \mathrm{E4} \qquad R_4 = \left\{\left(0, \frac{3}{16}\right), \left(\frac{3}{16}, \frac{1}{16}\right)\right\} \text{ for } \mathrm{G\sharp 4}$$

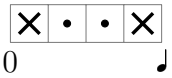
[6]This statement is made without proof, since it seems clear and the proof could be technical and take us away from our subject.

or in vector form

$R_1 =$ 

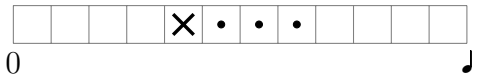
$R_2 =$ 

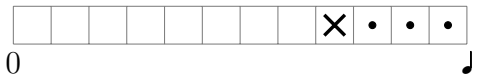
$R_3 =$ 

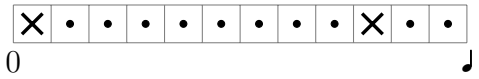
$R_4 =$ 

In order to stack them into a texture, we need to put all of them in the same rhythm quantization. Since the tatum of R_1*,* R_2 *and* R_3 *is* $\frac{1}{12}$ *and the one of* R_4 *is* $\frac{1}{16}$*, and the GCD between is* $\frac{1}{48}$*, we can rewritten the rhythms as*

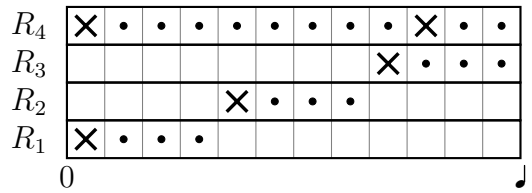
$R_1 =$ 

$R_2 =$ 

$R_3 =$ 

$R_4 =$ 

and stack them into

$T =$ 

.

Notice that we reversed the traditional order for stacking vectors into matrices in order to be more coherent with musical notation (where lower indices mean lower pitches and are place in the bottom).

4.1.2 Harmony

The term *harmony* can be even more intricate to define than texture. "Harmony" can refer to a field of study, a feature of a musical piece, or even an adjective.

In this context, *harmony* will be precisely defined in a mathematical manner to suit our purpose. In order to achieve this, akin to the process with texture, we first need to establish what we mean by a *chord*.

Definition 4.6 (Chord)**.** A **chord** C is a finite subset of $\mathcal{F}$ or $G_{\mathcal{F}}$, i.e., an element of $\mathcal{P}(\mathcal{F})$ or $\mathcal{P}(G_{\mathcal{F}})$.

This definition is intentionally broad. The absence of specification for the space $\mathcal{F}$ and the group $G_{\mathcal{F}}$ was intentional to keep the chord concept as general as possible. However, we will encounter particular chord types in specific scenarios. The following definition outlines four chord types that will be pertinent to our discussions.

Definition 4.7 (Types of chords)**.** We call

- a **positioned chord** an element of $\mathcal{P}(\mathcal{N})$,
- a **chroma chord** an element of $\mathcal{P}(\mathcal{N}_{12})$,
- a **positioned pattern** an element of $\mathcal{P}(\mathbb{Z})$,
- a **pattern** an element of $\mathcal{P}(\mathbb{Z}_{12})$.

These definitions comprise the four more general conceptions of chords; we list some examples in the following.

Examples 4.8.

1. *We usually consider the chord F major to be* $\mathrm{FM} = \{\mathrm{F}, \mathrm{A}, \mathrm{C}\} \in \mathcal{P}(\mathcal{N}_{12})$*, that is a chroma chord.*
2. *The positioned chord* $\mathrm{CM}_4^{1,3,5,8} = \{\mathrm{C4}, \mathrm{E4}, \mathrm{G4}, \mathrm{C5}\} \in \mathcal{P}(\mathcal{N})$ *is a root position of the C major chord.*
3. *A cadential six-four chord is the positioned pattern* $^6_4 = \{-5, 0, 4\} \in \mathcal{P}(\mathbb{Z})$ *where the* 0 *represents the tonic.*
4. *We say that the quality of C major is to be a major triad; the pattern of a major triad starting from its root is* $\mathrm{M} = \{\overline{0}, \overline{4}, \overline{7}\} \in \mathcal{P}(\mathbb{Z}_{12})$*.*

The examples provided illustrate that what are commonly referred to as chords are in fact one of the different types of chords we proposed. The subsequent examples highlight instances where our definition identifies entities as chords, although this may not align with conventional usage.

Examples 4.9.

1. *The silence* $\emptyset$ *is any kind of chord, i.e.,*

$$\emptyset \in \mathcal{P}(\mathcal{N}), \qquad \emptyset \in \mathcal{P}(\mathcal{N}_{12}), \qquad \emptyset \in \mathcal{P}(\mathbb{Z}), \qquad \emptyset \in \mathcal{P}(\mathbb{Z}_{12})\,.$$

2. *A note* $n \in \mathcal{N}$ *forms a (positioned) chord* $\{n\} \in \mathcal{P}(\mathcal{N})$.

3. *The major scale is the pattern* $\{\overline{0}, \overline{2}, \overline{4}, \overline{5}, \overline{7}, \overline{9}, \overline{11}\} \in \mathcal{P}(\mathbb{Z}_{12})$, *where* 0 *represents the tonic.*

4. *The third mode of limited transposition of Messiaen is the pattern* $\{\overline{0}, \overline{2}, \overline{3}, \overline{4}, \overline{6}, \overline{7}, \overline{8}, \overline{10}, \overline{11}\} \in \mathcal{P}(\mathbb{Z}_{12})$.

5. *The chromatic scale is* $\mathbb{Z}_{12} \in \mathcal{P}(\mathbb{Z}_{12})$.

6. *The* $\mathbf{V}$ *degree is the pattern* $\{\overline{2}, \overline{7}, \overline{11}\} \in \mathcal{P}(\mathbb{Z}_{12})$.

7. *A first inversion of an A minor chord is the positioned chord* $\{\mathrm{C4}, \mathrm{E4}, \mathrm{A4}\}$.

8. *The augmented sixth chords may be described by the patterns following patterns:*

$$\begin{aligned} \textit{Italian augmented sixth: } & \{-4, 0, 6\} \in \mathcal{P}(\mathbb{Z}) \\ \textit{French augmented sixth: } & \{-4, 0, 2, 6\} \in \mathcal{P}(\mathbb{Z}) \\ \textit{German augmented sixth: } & \{-4, 0, 3, 6\} \in \mathcal{P}(\mathbb{Z}) \end{aligned}$$

where the 0 *represents the tonic.*

As in the case of rhythms, every finite non empty chord can be represented by a vector belonging to $\mathcal{A}_2^M$ for some $M \in \mathbb{N}^*$. This is done as follows: let C be a chord; then, $|C| = N \in \mathbb{N}^*$. Let $C = \{c_1, c_2, ..., c_N\}$. We shall discuss two different cases: $\mathcal{F} = \mathbb{Z}$ or $\mathcal{N}$ and $\mathcal{F} = \mathbb{Z}_{12}$ or $\mathcal{N}_{12}$.

In the first case, let us consider the order in $\mathcal{F}$ given by the isomorphism with $\mathbb{Z}$. Then, we can define $c_0 = \min C \in \mathcal{F}$ and $c_1 = \max C \in \mathcal{F}$, with gives[7] us $M = (c_1 - c_0) + 1 \in \mathbb{N}^*$. We define $\boldsymbol{C} \in \mathcal{A}_2^M$ by

$$\forall m = 0, 1, ..., M,\ \boldsymbol{C}_m = \begin{cases} 1 & \text{if } c_0 + m \in C \\ 0 & \text{if } c_0 + m \notin C \end{cases}. \tag{4.5}$$

The case when $\mathcal{F} = \mathbb{Z}_{12}$ or $\mathcal{N}_{12}$ is tackled by using systematically a vector of size 12. We will also use always the index 0 corresponding either to the shift $\overline{0}$ either to the chroma C.

Chords will be plotted similarly as rhythms, but in this case, as vertical vectors, which will be exploited in the following section.

[7]The subtraction between members of $\mathcal{F}$ should be understood as the signed distance induced by the isomorphism with $\mathbb{Z}$.

Examples 4.10. *Let us plot the chords from Examples 4.8.*

$$\mathrm{FM} = \begin{matrix} \\ \\ \mathrm{A}\ 1 \\ \\ \\ \\ \mathrm{F}\ 1 \\ \\ \\ \\ \mathrm{C}\ 1 \end{matrix} \qquad \mathrm{CM}_4^{1,3,5,8} = \begin{matrix} \mathrm{C5}\ 1 \\ \\ \\ \\ \\ \mathrm{G4}\ 1 \\ \\ \\ \mathrm{E4}\ 1 \\ \\ \\ \\ \mathrm{C4}\ 1 \end{matrix} \qquad {}^{6}_{4} = \begin{matrix} 4\ 1 \\ \\ \\ \\ 0\ 1 \\ \\ \\ \\ \\ -5\ 1 \end{matrix} \qquad \mathrm{M} = \begin{matrix} \\ \\ \\ \\ \overline{7}\ 1 \\ \\ \\ \overline{4}\ 1 \\ \\ \\ \\ \overline{0}\ 1 \end{matrix}$$

While this representation of chords is adapted to our model, it takes a lot of place, so we alternate between this and the traditional presentation as sets.

Having defined chords, let us now proceed to define harmony. It is important to note that the definition of harmony we present is tailored to our specific objectives and may not align with a universal understanding of the term.

Definition 4.11 (Harmony)**.** A **harmony** H is an countable tuple of chords, i.e., $H = (C_i)_{i\in I} \in (\mathcal{P}(\mathcal{F}))^I \simeq \mathcal{P}(\mathcal{F} \times I)$, with I being a countable set. A finite harmony of size $|I|$ is a harmony where $|I| < \infty$.

We see that the notion of harmony is analogous to the notion of texture with the substitution of time by frequency and rhythm by chord. We can also establish the different kinds of harmonies depending on the different choices for the frequency space or group.

Definition 4.12 (Types of harmonies)**.** Let I be a countable set. We call

- a **positioned harmony** an element of $(\mathcal{P}(\mathcal{N}))^I$,
- a **chroma harmony** an element of $(\mathcal{P}(\mathcal{N}_{12}))^I$,
- a **positioned harmonic pattern** an element of $(\mathcal{P}(\mathbb{Z}))^I$,
- a **harmonic pattern** an element of $(\mathcal{P}(\mathbb{Z}_{12}))^I$.

Before showing how to combine textures and harmonies to create music, let us give some examples of harmonies.

Examples 4.13.

1. *A cadential six-four in major mode is the positioned harmonic pattern*

$$(\{-5,4,7,12\},\{-5,2,7,11\},\{-12,4,7,12\}) \in (\mathcal{P}(\mathbb{Z}))^3$$

where 0 *is the tonic.*

2. *Its corresponding positioned harmony in F is*

$$(\{\mathrm{C4},\mathrm{A4},\mathrm{C5},\mathrm{F5}\},\{\mathrm{C4},\mathrm{G4},\mathrm{C5},\mathrm{E5}\},\{\mathrm{F3},\mathrm{A4},\mathrm{C5},\mathrm{F5}\}) \in (\mathcal{P}(\mathcal{N}))^3\,.$$

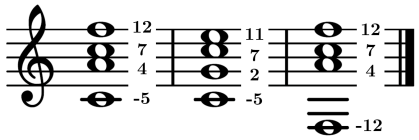
3. *The* $\mathbf{V^7}$-$\mathbf{I}$ *harmonic pattern is*

$$(\{\overline{7},\overline{11},\overline{2},\overline{5}\},\{\overline{0},\overline{4},\overline{7}\}) \in (\mathcal{P}(\mathbb{Z}_{12}))^2\,.$$

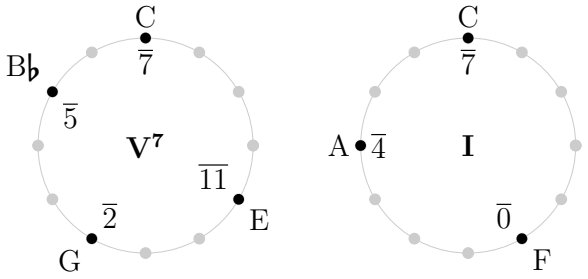
4. *Its corresponding chroma harmony in F is*

$$(\{\mathrm{C},\mathrm{E},\mathrm{G},\mathrm{B}\flat\},\{\mathrm{F},\mathrm{A},\mathrm{C}\}) \in (\mathcal{P}(\mathcal{N}_{12}))^2\,.$$

These examples are illustrated in Figure 4.2



(a) A cadential six-four positioned harmonic pattern and its positioned harmony in F



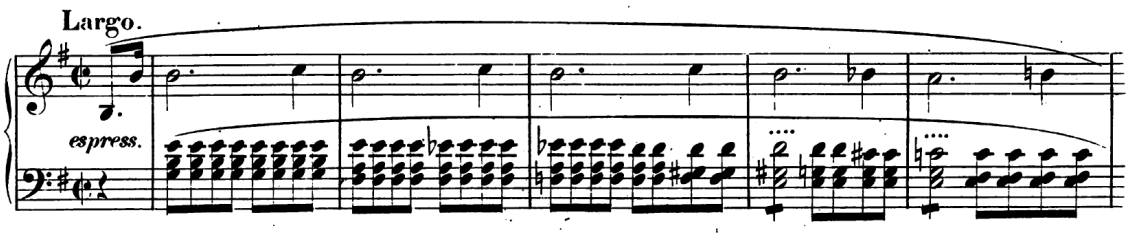
(b) The $\mathbf{V^7}$-$\mathbf{I}$ harmonic pattern and its corresponding chroma harmony in F

Figure 4.2: Illustration of the harmonies from Examples 4.13.

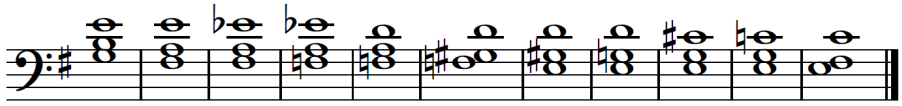
These examples highlight the relationship between chords and patterns. While chords correspond to points in the space $\mathcal{F}$, patterns are shifts within the group $G_{\mathcal{F}}$. In the upcoming discussion, we will simplify notation by using $\mathcal{F}$ interchangeably for either $\mathcal{F}$ or $G_{\mathcal{F}}$. This approach aims to enhance the ease of generalization when dealing with chords.

As in the case of textures, harmonies can be represented by a matrix.

Example 4.14. *Let us consider the harmony of the left hand of Chopin's Prelude n°4 in E minor (Figure 4.3a). If we consider the 11 chords that appear (Figure 4.3b) and we stack them in order of appearance, we have the harmony given in Figure 4.3c.*



(a) Score



(b) Harmony

C4	1	1									
			1	1							
					1	1	1	1			
									1		
										1	1
B3	1										
		1	1	1	1						
						1	1				
G3	1							1	1	1	
		1	1								1
				1	1	1					
E3							1	1	1	1	1

(c) Matrix representing harmony

Figure 4.3: Chopin's Prelude N°4, Op.28 mm.0-5.

4.1.3 Harmonic Texture

Now that we have defined texture and harmony, let us see how we can combine them to create time-frequency elements and create music.

We start by defining the combination of a rhythm and a chord. To do that, we draw upon two mathematical tools: first, we identify a chord with its characteristic

function, that is

$$\begin{array}{rccc} \mathbb{1}: & \mathcal{P}(\mathcal{F}) & \to & \mathcal{A}_2^{\mathcal{F}} \\ & C & \mapsto & \mathbb{1}_C \end{array} . \tag{4.6}$$

We omit the characteristic function and say that a chord C belongs to $\mathcal{A}_2^{\mathcal{F}}$. In this case, a harmony is an element of $\left(\mathcal{A}_2^{\mathcal{F}}\right)^I \simeq \mathcal{A}_2^{\mathcal{F}\times I}$.

Secondly, we use the residuated triplet induced by the lattice multiplication exposed in Section 2.1.3.2.

$$\begin{array}{rccc} \bullet: & \mathcal{A}_3 \times \mathcal{A}_2 & \to & \mathcal{A}_3 \\ & (a,b) & \mapsto & a \bullet b = \begin{cases} a & \text{if } b = 1 \\ \bot & \text{if } b = 0 \end{cases} \end{array} . \tag{4.7}$$

Now we are able to define a *rhythmed chord*.

Definition 4.15 (Rhythmed chord)**.** Let $R \in \mathcal{A}_3^{\mathcal{T}}$ be a rhythm and $C \in \mathcal{A}_2^{\mathcal{F}}$ be a chord. Then, the **rhythmed chord** $R \otimes C$ is defined as

$$\begin{array}{rccc} R \otimes C: & \mathcal{T} \times \mathcal{F} & \to & \mathcal{A}_3 \\ & (t,\xi) & \mapsto & R(t) \bullet C(\xi) \end{array} . \tag{4.8}$$

Let us illustrate a rhythmed chord in the following example.

Example 4.16. *Consider the left hand of the first four measures of the Mozart's Piano Sonata No. 8 in A minor, K. 310 / 300d (see Figure 4.7).*

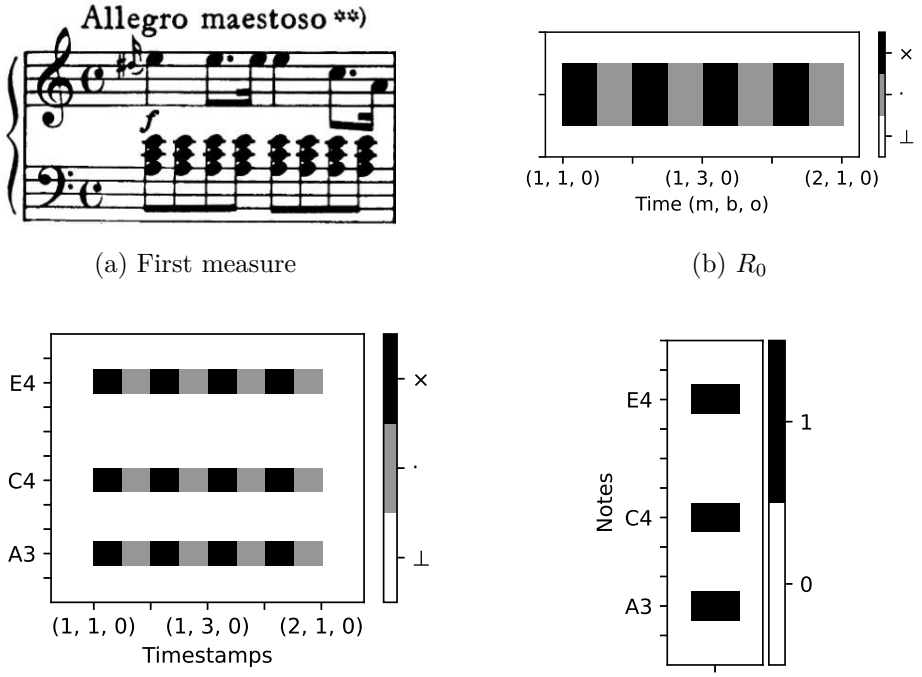
Then, if we consider the placed rhythm $R_0 = t_0 + R \in \mathcal{A}_3^{\mathcal{T}_\circ^{\mathbf{c}}}$ *given by*[8] $t_0 = (1,1,0) \in \mathcal{T}_\circ^{\mathbf{c}}$ *and* $R \equiv \{(0,\frac{1}{8}),(\frac{1}{8},\frac{1}{8}),(\frac{2}{8},\frac{1}{8}),(\frac{3}{8},\frac{1}{8})\}$*, and the positioned chord* $\mathrm{Am} = \{\mathrm{A3},\mathrm{C4},\mathrm{E4}\} \in \mathcal{A}_2^{\mathcal{N}}$*, we have that* $R_0 \otimes \mathrm{Am}$ *represents the two first beats of the left hand. Figure 4.4 illustrates this fact.*

We can now define our fundamental mathematical object.

Definition 4.17 (Harmonic texture)**.** Let I be a countable set of indices. Let $T = \{R_i\}_{i\in I} \in \mathcal{A}_3^{\mathcal{T}\times I}$ be a texture and let $H = \{C_i\}_{i\in I} \in \mathcal{A}_2^{\mathcal{F}\times I}$ be a harmony. We define the **harmonic texture** generated by T and H as

$$T \otimes H = \bigvee_{i\in I} R_i \otimes C_i \in \mathcal{A}_3^{\mathcal{T}\times\mathcal{F}} . \tag{4.9}$$

[8]We recall that, as exposed in Section 2.1.1.3, we denote a timestamp inside a score as a triplet referring to bar, beat and offset.



(a) First measure

(b) R_0

(c) $R_0 \otimes \mathrm{Am}$

(d) Am

Figure 4.4: Model of the left hand of the first measure of Mozart's Piano Sonata No. 8 in A minor, K. 310 / 300d through the rhythmed chord $R_0 \otimes \mathrm{Am}$.

Remark 4.18. Notice that we made an abuse of notation when writing $T \otimes H$; indeed, T and H are (strictly speaking) functions with I as domain, which means that their tensor product would have $I \times I$ as domain.

If we present texture and harmony with matrices, i.e., $\boldsymbol{T} \in \mathcal{A}_3^{N \times I}$ and $\boldsymbol{H} \in \mathcal{A}_2^{M \times I}$, with $N, M, I \in \mathbb{N}^*$, we find that the harmonic texture is a matrix multiplication, replacing the sum by the supremum.

Indeed, if we have a texture $T = (R_i)_{i \in I}$ and a harmony $H = (C_i)_{i \in I}$, both indexed by I, the harmonic texture $T \otimes H$ is written as

$$\begin{aligned}(T \otimes H)(t, \xi) &= \left(\bigvee_{i \in I} R_i \otimes C_i \right)(t, \xi) \\ &= \bigvee_{i \in I} R_i(t) \cdot C_i(\xi)\end{aligned}$$

that shall be compared with the matrix multiplication between two matrices $A = (a_{ti})$ and $B = (b_{i\xi})$ that is given by

$$\begin{aligned}(A \cdot B)_{t\xi} &= (a_{ti}) \cdot (b_{i\xi}) \\ &= \sum_{i \in I} a_{ti} \cdot b_{i\xi}\end{aligned}$$

with the same requirement of index compatibility.

Moreover, if we use the tensor notation for texture and harmony, i.e., $\boldsymbol{T}_n^i$, $\boldsymbol{H}_j^m$, the harmonic texture $\boldsymbol{HT}_n^m$ would be the (supremum) contraction on the index i, that is, using the Einstein summation convention,

$$\boldsymbol{HT}_n^m = \boldsymbol{T}_n^i \boldsymbol{H}_i^m := \bigvee_i \boldsymbol{T}_n^i \boldsymbol{H}_i^m \tag{4.10}$$

Example 4.19. *Let us consider the left hand of the measure 41 from the first movement of Beethoven's Piano Sonata No.17, Op.31 No.2.*

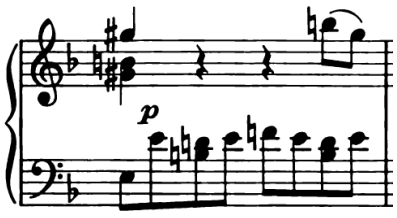


Figure 4.5 shows how to render this harmonic texture through matrix multiplication.

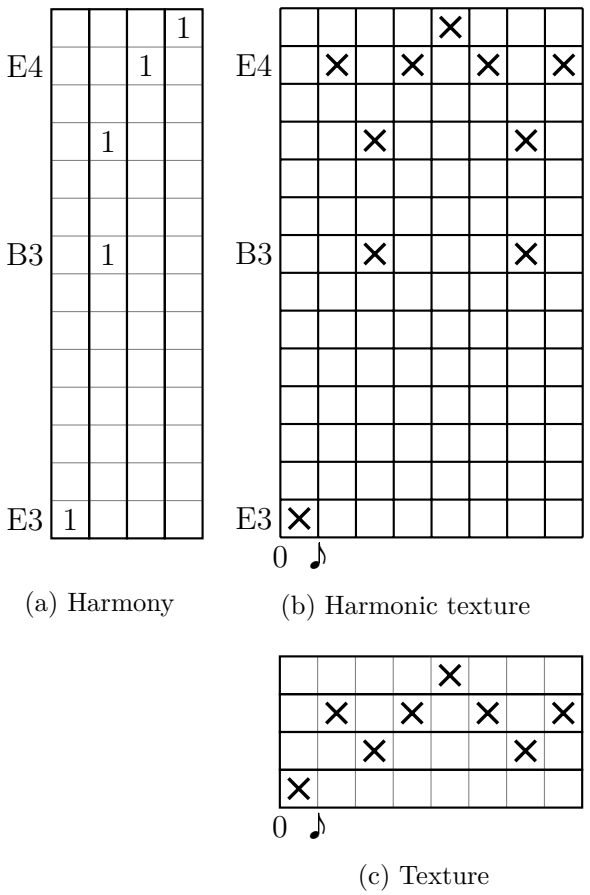
Let us explore some examples of harmonic textures.

Example 4.20. *We call the* Alberti bass *with tatum* ♪ *the texture*

$$T = (R_i)_{i \in I} = \begin{array}{|c|c|c|c|}\hline & \times & & \times \\ \hline & & \times & \\ \hline \times & & & \\ \hline \end{array}$$

0 ♪

with $I = \{0, 1, 2\}$.



(a) Harmony

(b) Harmonic texture

(c) Texture

Figure 4.5: Harmonic texture as a matrix multiplication representing the left hand of the measure 41 from the first movement of Beethoven's Piano Sonata No.17, Op.31 No.2.

It is used for instance in the left hand of Mozart's Piano Sonata No.16 in C major, K.545 mm.1-4 exposed in Figure 4.6. Indeed, if we choose the following harmonies:

$$H_0 = \begin{pmatrix} \{\text{E4}\} \\ \{\text{G4}\} \\ \{\text{C4}\} \end{pmatrix} \quad H_1 = \begin{pmatrix} \{\text{E4}\} \\ \{\text{G4}\} \\ \{\text{C4}\} \end{pmatrix} \quad H_2 = \begin{pmatrix} \{\text{F4}\} \\ \{\text{G4}\} \\ \{\text{D4}\} \end{pmatrix} \quad H_3 = \begin{pmatrix} \{\text{E4}\} \\ \{\text{G4}\} \\ \{\text{C4}\} \end{pmatrix}$$

$$H_4 = \begin{pmatrix} \{\text{F4}\} \\ \{\text{A4}\} \\ \{\text{C4}\} \end{pmatrix} \quad H_5 = \begin{pmatrix} \{\text{E4}\} \\ \{\text{G4}\} \\ \{\text{C4}\} \end{pmatrix} \quad H_6 = \begin{pmatrix} \{\text{D4}\} \\ \{\text{G4}\} \\ \{\text{B4}\} \end{pmatrix} \quad H_7 = \begin{pmatrix} \{\text{E4}\} \\ \{\text{G4}\} \\ \{\text{C4}\} \end{pmatrix}$$

we have that the first four measures are represented by supremum of the harmonic textures

$$\bigvee_{j=0}^{7}(t_j + T) \otimes H_j$$

where $t_j = (1, 1, 0) + \frac{j}{2} \in \mathcal{T}_{\mathbf{o}}^{\mathbf{C}}$ *and* $t + T = (t + R_i)_{i \in I} \in \mathcal{A}_3^{\mathcal{T}_{\mathbf{o}}^{\mathbf{C}} \times I}$.

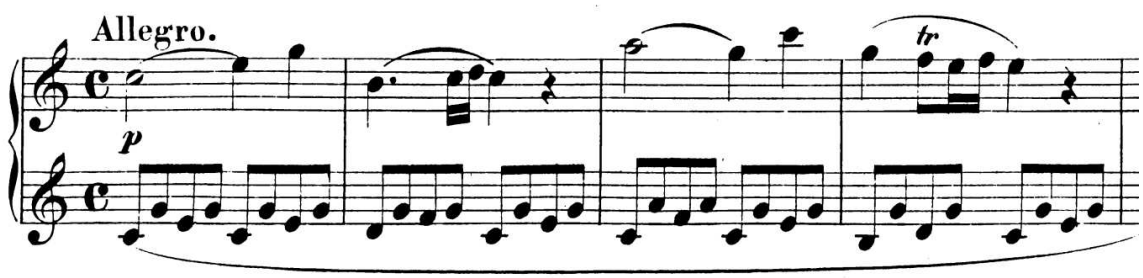


Figure 4.6: Mozart's Piano Sonata No.16 in C major, K.545 mm.1-4.

This construction is extremely flexible and allows us to understand through the same texture chords of different sizes. Let us show that by examining the left hand of the first four measures of the Mozart's Sonata No.16 in A minor, K. 310 / 300d (Figure 4.7).

Example 4.21. *Let R be the rhythm of the Example 4.16 and let C_0 and C_1 be the chords $C_0 = \{\text{A3}, \text{C4}, \text{E4}\}$ and $C_1 = \{\text{A3}, \text{B3}, \text{D4}, \text{E4}\}$. Then, the first four measures of the Mozart's Sonata No.16 in A minor, K. 310 / 300d are represented by*

$$(t_1 + R) \otimes C_0 \vee (t_2 + R) \otimes C_1 \vee (t_3 + R) \otimes C_0 \vee (t_4 + R) \otimes C_1$$

where $\forall n \in \{1, 2, 3, 4\}$, $t_n = (n, 1, 0) \in \mathcal{T}_{\mathbf{o}}^{\mathbf{C}}$.

An even more revealing example is the start of the *Lacrimosa* of Mozart's Requiem in D minor, K.626.

Example 4.22. *We consider together the parts of the Violins and Viola of the first two measures of the* Lacrimosa *of Mozart's Requiem in D minor, K.626 (Figure 4.8).*

Let $T \in \left(\mathcal{A}_3^{\mathbb{Q}}\right)^3$ *be the texture*

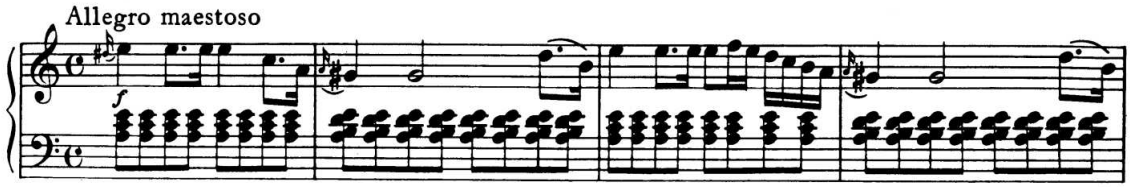
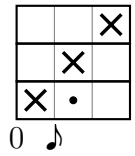


Figure 4.7: Mozart's Sonata No.16 in A minor, K. 310 / 300d, mm.1-4.



Let us consider the harmonies

$$H_0 = (\{\text{D4}, \text{F4}\}, \{\text{C}\sharp 5\}, \{\text{D5}\}) \qquad H_1 = (\{\text{F4}, \text{A4}\}, \{\text{A5}\}, \{\text{B}\flat 5\})$$
$$H_2 = (\{\text{E4}, \text{G4}\}, \{\text{D5}\}, \{\text{C}\sharp 5\}) \qquad H_3 = (\{\text{G4}, \text{C}\sharp 4\}, \{\text{C6}\}, \{\text{B}\flat 5\})$$
$$H_4 = (\{\text{F4}, \text{D5}\}, \{\text{A5}\}, \{\text{D6}\}) \qquad H_5 = (\{\text{G3}, \text{E4}\}, \{\text{B}\flat 5\}, \{\text{G5}\})$$
$$H_6 = (\{\text{A3}, \text{D4}\}, \{\text{E5}\}, \{\text{F5}\}) \qquad H_7 = (\{\text{A3}, \text{G4}\}, \{\text{A5}\}, \{\text{C}\sharp 5\})\,.$$

Then, the excerpt may be represented by the supremum

$$\bigvee_{i=0}^{7} (t_i + T) \otimes H_i$$

with $t_i = (1, 1, 0) + \frac{3i}{8} \in \mathcal{T}_{\mathbf{8}}^{\mathbf{12}}$ *and* $t + T = (t + R_i)_{i \in I} \in \mathcal{A}_3^{\mathcal{T}_{\mathbf{8}}^{\mathbf{12}} \times I}$.

A potential limitation of the notion of texture is that several different textures in the musical sense may appear as the same texture in our definition. For instance, the textures  and  are both represented by the texture $R = (R_i)_{i=0}^3$ with $R_i \equiv \{(\frac{i}{8}, \frac{1}{8})\}$ but they are fairly different musically. This flaw is solved by defining the concept of *chord texture*.

Definition 4.23 (Chord texture)**.** Let $T = (R_i)_{i \in \mathbb{N}}$ be a texture and let $C \in \mathcal{A}_2^{\mathcal{F}}$ be a positioned chord or a positioned pattern, i.e., $\mathcal{F} = \mathcal{N}$ or $\mathcal{F} = \mathbb{Z}$. We define the **chord texture** generated by T and C as

$$T \otimes C := T \otimes H_C \in \mathcal{A}_3^{\mathcal{T} \times \mathcal{F}} \tag{4.11}$$

with $H_C = (\{c_i\})_{i=1}^{|C|}$ and $\forall i, j \in \{1, 2, ..., |C|\}$, $i < j \Rightarrow c_i < c_j$, where the order in $\mathcal{N}$ is the one induced by the isomorphism with $\mathbb{Z}$.

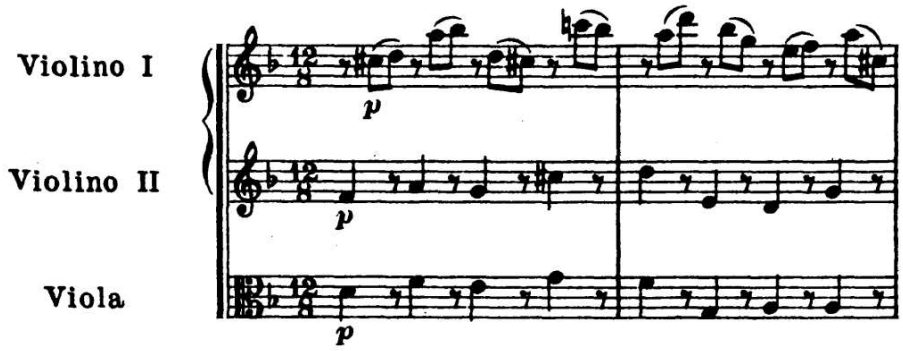


Figure 4.8: *Lacrimosa* of Mozart's Requiem in D minor, K.626, mm.1-4.

Example 4.24. *If we examine the left hand of Chopin's Nocturnes 1 and 2, Op. 9 (see Figure 4.10), we see that both share the same note values (eighth notes) and then similar textures; however, if we model them through chord textures, we have a different point of view.*

For Nocturne 1, we choose the texture exposed in Figure 4.9a.

For Nocturne 2, we choose the texture exposed in Figure 4.9b.

With these textures, we can then model the left hand through chord textures with the following chords:

1. *for Nocturne 1, mm. 1-2*

$$C_1 = (\mathrm{B}\flat 2, \mathrm{F}3, \mathrm{B}\flat 3, \mathrm{D}\flat 4, \mathrm{F}4) \qquad C_2 = (\mathrm{B}\flat 2, \mathrm{F}3, \mathrm{A}3, \mathrm{E}\flat 4, \mathrm{F}4)$$
$$C_3 = (\mathrm{B}\flat 2, \mathrm{F}3, \mathrm{B}\flat 3, \mathrm{D}\flat 4, \mathrm{F}4) \qquad C_4 = (\mathrm{B}\flat 2, \mathrm{F}3, \mathrm{B}\flat 3, \mathrm{D}\flat 4, \mathrm{F}4)$$

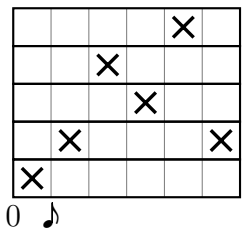
2. *for Nocturne 2, m. 1*

$$C_1 = (\mathrm{E}\flat 2, \mathrm{G}3, \mathrm{B}\flat 3, \mathrm{E}\flat 4, \mathrm{G}4) \qquad C_2 = (\mathrm{E}\flat 3, \mathrm{A}\flat 3, \mathrm{C}\flat 4, \mathrm{D}4, \mathrm{A}\flat 4)$$
$$C_3 = (\mathrm{E}\flat 2, \mathrm{G}3, \mathrm{B}\flat 3, \mathrm{E}\flat 4, \mathrm{G}4) \qquad C_4 = (\mathrm{D}2, \mathrm{G}3, \mathrm{B}\flat 3, \mathrm{E}\flat 4, \mathrm{G}4)\,.$$

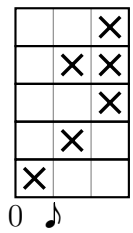
Another interesting point of chord textures is that we may model melodies and motives with them[9]. Let us see an example.

Example 4.25. *Let us consider the right hand of the 3rd movement of Mozart's Sonata Nº11 (see Figure 4.11). We consider the texture* $T = \{R_i\}_{i=0}^{3}$ *with*

[9]It is not exclusive to chord textures; we can do that also with harmonic textures, but usually it is important to keep the notion of order in the notes when dealing with melodies (and with motives, to a less extent).

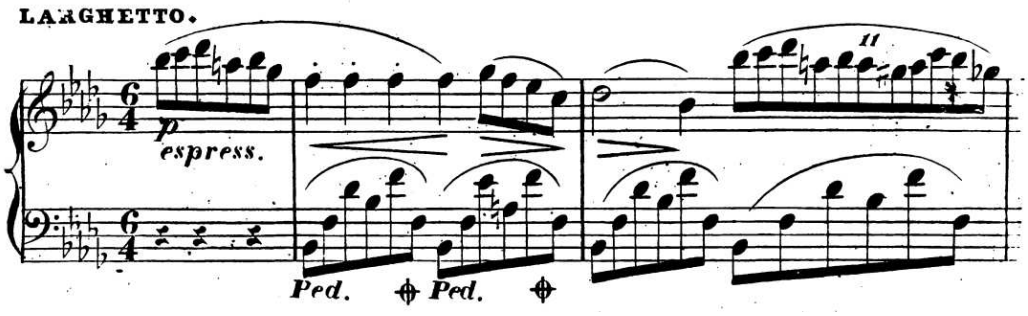


(a) Texture for Nocturne 1

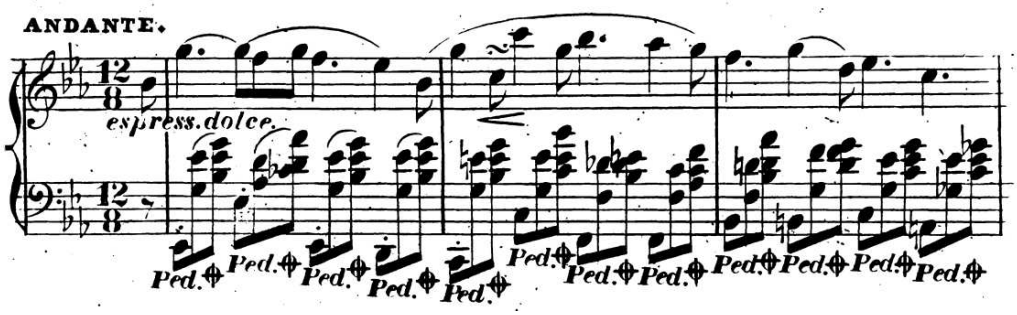


(b) Texture for Nocturne 2

Figure 4.9: Textures for chord textures represented by arrays.

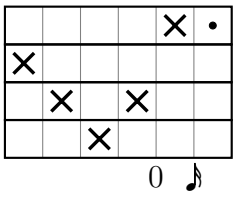


(a) Nocturne 1, mm. 0-2



(b) Nocturne 2, mm. 1-4

Figure 4.10: Incipit of Chopin's Nocturnes 1 and 2, Op. 9.



Then, the 10 first notes of the right hand may be represented by the texture T applied to the chords $C_1 = (\mathrm{G\sharp 4}, \mathrm{A4}, \mathrm{B4}, \mathrm{C5})$ and $C_2 = (\mathrm{B4}, \mathrm{C5}, \mathrm{D5}, \mathrm{E5})$. Furthermore, the second half of the measure 3 and the first eighth of the measure 4 may be represented by the chord $C_3 = (\mathrm{G\sharp 5}, \mathrm{A5}, \mathrm{B5}, \mathrm{C6})$.

Actually, if we had chosen as texture the sub-texture $\{R_i\}_{i=0}^{2}$ of texture T, we may even justify the second half of measure 2 and the first half of measure 3.

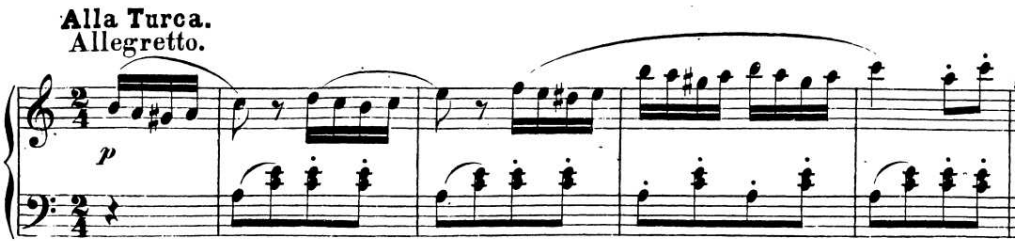


Figure 4.11: Mozart's Sonata N°11, 3rd movement, *Alla Turca* - Allegretto, mm. 0-4

4.2 Generating Piano Rolls with Harmonic Textures and Mathematical Morphology

We have now a potentially powerful tool for generating piano rolls and explaining the way scores are created through the use of time-frequency entities (rhythms/chords and textures/harmonies). However, to generate complex tonal[10] pieces, we need to be able to combine these harmonic textures into bigger structures.

We do that by using morphological dilation and a tree structure. In Section 4.2.1, we present how to combine several harmonic textures by using the morphological greyscale dilation. Then, in Section 4.2.2, we organize several dilations into a tree, which enables us to create complex pieces through the use of hierarchical structures. Finally, in Section 4.2.3, we expose how to implement this computationally, in particular by the creation of an XML schema that defines a ScoreXML file that can be compiled into a `PianoRoll` data structure.

[10]In this work, the examples are picked-up from Western tonal music, but the formalism can handle any type of music based on rational divisions of the time and integer divisions of the octave.

4.2.1 Combining Harmonic Textures with Dilation

In previous examples, we have seen how to produce music excerpts by using harmonic textures. However, there was always a moment in which we needed to combine different harmonic textures to render an excerpt. In these examples, the combination was made by using the supremum operator on shifted versions of the texture, with expressions like $\bigvee t + T$, which emulated concatenation.

In this section, we will generalize both concatenation and superposition by using morphological dilation. Let us illustrate that with an example. We consider measures 0 to 8 of Beethoven's Piano Sonata No.17, Op.31 No.2 mm.0-8 (Figure 4.12). We are showing how to build them by using harmonic textures and morphological dilation.

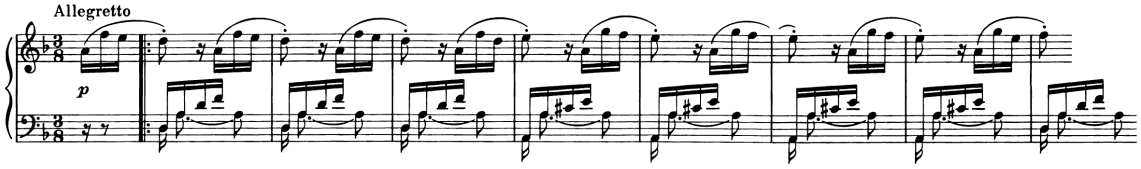
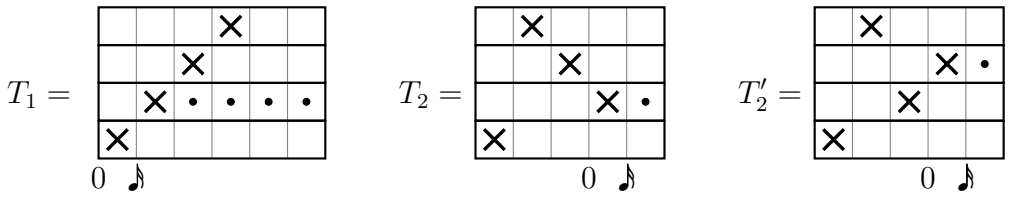


Figure 4.12: Beethoven's Piano Sonata No.17, Op.31 No.2 mm.0-8.

Let us define the textures



belonging to $\mathcal{A}_3^{\mathbb{Q}\times I}$, with $|I| = 4$, and the chords[11]

$$\begin{aligned} \text{Dmin} &= \{\text{A4}, \text{D5}, \text{E5}, \text{F5}\} & \text{A7} &= \{\text{A4}, \text{E5}, \text{F5}, \text{G5}\} \\ \text{Dm} &= \{\text{D3}, \text{A3}, \text{D4}, \text{F4}\} & \text{AM} &= \{\text{A2}, \text{A3}, \text{C}\sharp\text{4}, \text{E4}\} \end{aligned}$$

belonging to $\mathcal{A}_2^{\mathcal{N}}$, and with cardinal 4 (thus generating harmonies of size 4 for the chord texture).

We now form the following chord textures:

$$\begin{aligned} \text{Dmin}_2 &= T_2 \otimes \text{Dmin} & \text{A7}_2 &= T_2 \otimes \text{A7} & \text{Dm}_1 &= T_1 \otimes \text{Dm} \\ \text{Dmin}_2' &= T_2' \otimes \text{Dmin} & \text{A7}_2' &= T_2' \otimes \text{A7} & \text{AM}_1 &= T_1 \otimes \text{AM}\,. \end{aligned}$$

[11] The names of the chords are selected based on the quality of the chords (M → major, m → minor, 7 → dominant seventh and min → minor scale) for avoiding long notations, but we shall recall that they are positioned chords.

We want to combine them into a single piano roll P. For doing that, we use the morphological dilation

$$P = A \oplus B := \bigvee_{j \in J} A_j \oplus B_j$$

where $A_j \in \mathcal{A}_2^{\mathcal{T}^{\mathbb{3}} \times \mathbb{Z}}$ and $B_j \in \mathcal{A}_3^{\mathbb{Q} \times \mathcal{N}}$ are the following:

$$\begin{aligned}
B_1 &= \mathrm{Dmin}_2 & A_1 &= \{\big((1,1,0),0\big), \big((2,1,0),0\big), \big((3,1,0),0\big)\} \\
B_2 &= \mathrm{Dmin}_2' & A_2 &= \{\big((4,1,0),0\big)\} \\
B_3 &= \mathrm{A7}_2 & A_3 &= \{\big((5,1,0),0\big), \big((6,1,0),0\big), \big((7,1,0),0\big)\} \\
B_4 &= \mathrm{A7}_2' & A_4 &= \{\big((8,1,0),0\big)\} \\
B_5 &= \mathrm{Dm}_1 & A_5 &= \{\big((1,1,0),0\big), \big((2,1,0),0\big), \big((3,1,0),0\big), \big((8,1,0),0\big)\} \\
B_6 &= \mathrm{AM}_1 & A_6 &= \{\big((4,1,0),0\big), \big((5,1,0),0\big), \big((6,1,0),0\big), \big((7,1,0),0\big)\}\,.
\end{aligned}$$

The result is show in Figure 4.13.

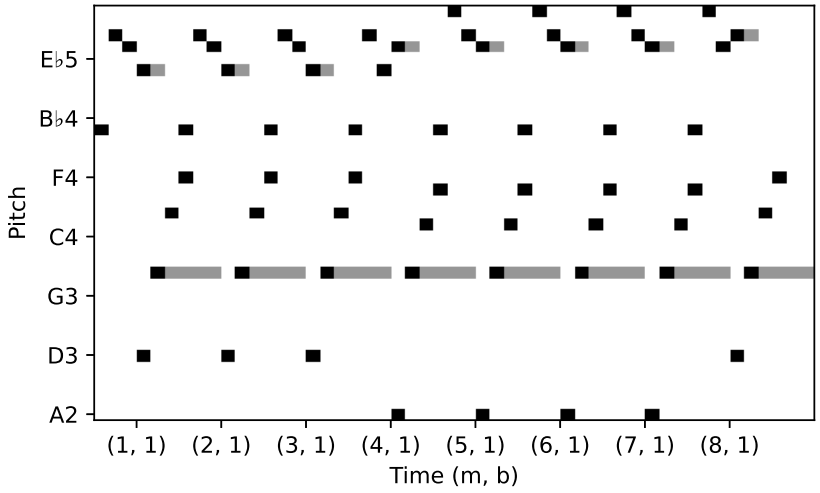


Figure 4.13: Piano roll generated by a dilation representing the measures 0 to 8 of the 3rd movement of Beethoven's of Piano Sonata No.17, Op.31 No.2.

This procedure is a way of generating infinite possibilities of piano rolls, using three elements:

1. a collection of activations $\mathscr{A} = (A_j)_{j \in J} \in \mathcal{A}_2^{\mathcal{T} \times \mathbb{Z} \times J}$,

2. a collection of harmonic textures $\mathscr{B} = (B_j)_{j\in J} \in \mathcal{A}_3^{\mathbb{Q}\times\mathcal{F}\times J}$ generated by
   (a) a collection of textures $\mathscr{T} = (T_j)_{j\in J} = (R_i^j)_{i\in I_j}^{j\in J} \in \mathcal{A}_3^{\mathbb{Q}\times I_j\times J}$,
   (b) a collection of harmonies $\mathscr{H} = (H_j)_{j\in J} = (C_i^j)_{i\in I_j}^{j\in J} \in \mathcal{A}_2^{\mathcal{F}\times I_j\times J}$.

The generation of a piano roll $P \in \mathcal{A}_3^{\mathcal{T}\times\mathcal{F}}$ is summarized by the formula[12]

$$\begin{aligned} P &= \mathscr{A} \oplus \mathscr{B} \\ &= \mathscr{A} \oplus \mathscr{T} \otimes \mathscr{H} \qquad (4.12) \\ &= \bigvee_{j\in J} A_j \oplus T_j \otimes H_j \\ &= \bigvee_{j\in J} A_j \oplus \left(\bigvee_{i\in I_j} R_i^j \otimes C_i^j \right) \\ &= \bigvee_{j\in J} \bigvee_{i\in I_j} A_j \oplus R_i^j \otimes C_i^j \,. \qquad (4.13) \end{aligned}$$

Some interesting nuances merit to be outlined; the textures have $\mathbb{Q} \times I_j \times J$ as domain, and the harmonies $\mathcal{F} \times I_j \times J$, which make the harmonic textures having as domain $\mathbb{Q} \times \mathcal{F} \times J$. This is compatible with the fact that the activations have $\mathcal{T} \times \mathbb{Z} \times J$ as domain, since there is a sum action defined between $\mathcal{T}$ and $\mathbb{Q}$, and between $\mathcal{F}$ and $\mathbb{Z}$. However, another possibility, like using $\mathcal{T}$ for textures and $\mathbb{Q}$ for activations, can work. In fact, every combination that allows to have a sum between members of the different spaces (which is required by the definition of the dilation) is acceptable.

This leads us to explain the meaning of the 0 that is the frequency used in every point of A_j; this frequency means "no shift". However, if we consider the measures 150 to 158 of the same piece (Figure 4.14), we see that we might have represented this excerpt by using the exact same harmonic textures but changing the activations, making a translation in time of $150 \cdot \frac{3}{8}$ and using -4 instead of 0 in the frequencies.

These considerations lead us to the definition of compatibility.

Definition 4.26 (Compatibility)**.** Let $\mathcal{T}_A$ and $\mathcal{T}_B$ be either a time space either a time group. Let $\mathcal{F}_A$ and $\mathcal{F}_B$ be either a frequency space either a frequency group. We say that $\mathcal{T}_A$ (resp. $\mathcal{F}_A$) is compatible with $\mathcal{T}_B$ (resp. $\mathcal{F}_B$) with output $\mathcal{T}_C$ (resp. $\mathcal{F}_C$) if there is a sum operator $+ : \mathcal{T}_A \times \mathcal{T}_B \to \mathcal{T}_C$ (resp. $+ : \mathcal{F}_A \times \mathcal{F}_B \to \mathcal{F}_C$).

[12] We assume that $\otimes$ has priority over $\oplus$ as in the case of $\times$ and $+$.

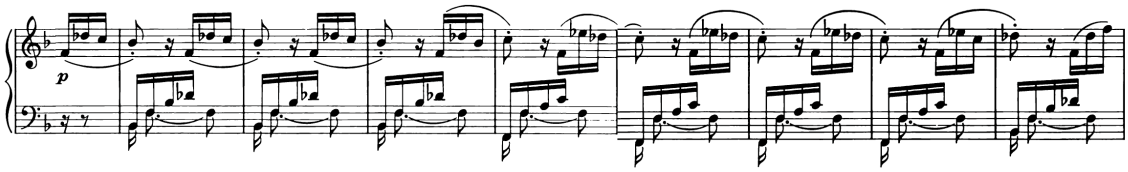


Figure 4.14: Beethoven's Piano Sonata No.17, Op.31 No.2 mm.150-158.

The compatibility arises in the context of spaces with groups acting on them; when we have a space E and a group $(G, +)$ acting on it, there is three cases of compatibility:

1. the group with itself, the output being the group, given by the sum $+ : G \times G \to G$,

2. the space with the group, the output being the space, given by the sum $+ : E \times G \to E$,

3. the group with the space, the output being the space, given by the sum $+ : G \times E \to E$, $(x, p) \mapsto p + x$.

The only case of incompatibility is between the space and the space, where there is no meaningful[13] interpretation of a sum; what would mean C4 + C4?

The notion of compatibility might also be defined for the lattices; if we have the lattices $\mathcal{A}_A$, $\mathcal{A}_B$ and $\mathcal{A}_C$, the compatibility would mean that there is a residuated triplet defined by the lattice multiplication $\bullet : \mathcal{A}_A \times \mathcal{A}_B \to \mathcal{A}_C$. The paradigmatic case of that is the lattice multiplication presented in Section 2.1.3.2.

What we think might be the default choice is to use the space-like elements for activations and the group-like elements for harmonic textures. For instance, we might have chosen the patterns

$$\begin{aligned} \text{min} &= \{-5, 0, 2, 3\} & 7 &= \{-5, 2, 3, 5\} \\ \text{m} &= \{0, 7, 12, 15\} & \text{M} &= \{-5, 7, 11, 14\} \end{aligned}$$

which belong to $\mathcal{A}_2^{\mathbb{Z}}$, and thus having harmonic textures belonging to $\mathcal{A}_3^{\mathbb{Q}\times\mathbb{Z}}$, and the

[13]Of course we may artificially define a sum, but we only focus on sums that make musical sense.

activations

$$
\begin{aligned}
A_1 &= \{\big((1,1,0),\mathrm{D5}\big),\big((2,1,0),\mathrm{D5}\big),\big((3,1,0),\mathrm{D5}\big)\}\\
A_2 &= \{\big((4,1,0),\mathrm{D5}\big)\}\\
A_3 &= \{\big((5,1,0),\mathrm{D5}\big),\big((6,1,0),\mathrm{D5}\big),\big((7,1,0),\mathrm{D5}\big)\}\\
A_4 &= \{\big((8,1,0),\mathrm{D5}\big)\}\\
A_5 &= \{\big((1,1,0),\mathrm{D3}\big),\big((2,1,0),\mathrm{D3}\big),\big((3,1,0),\mathrm{D3}\big),\big((8,1,0),\mathrm{D3}\big)\}\\
A_6 &= \{\big((4,1,0),\mathrm{D3}\big),\big((5,1,0),\mathrm{D3}\big),\big((6,1,0),\mathrm{D3}\big),\big((7,1,0),\mathrm{D3}\big)\}
\end{aligned}
$$

which belong to $\mathcal{A}_2^{\mathcal{T}_{\mathbf{3}}^{\mathbf{3}}\times\mathcal{N}}$, and we would produce the same piano roll.

We finish this section mentioning that this formalism can handle also the use of dynamics: if we use functions $A_j \in \mathcal{D}^{\mathcal{T}\times\mathcal{F}}$ and the residuated triplet exposed in Section 2.1.3.4, we would have

$$
\mathscr{A} \in \mathcal{D}^{\mathcal{T}\times\mathcal{F}\times J}, \qquad \mathscr{T} \in \mathcal{A}_3^{\mathbb{Q}\times I_j\times J}, \qquad \mathscr{H} \in \mathcal{A}_2^{\mathbb{Z}\times I_j\times J},
$$

and

$$
P = \mathscr{A} \oplus \mathscr{T} \otimes \mathscr{H} \in \mathcal{A}_{\mathcal{D}}^{\mathcal{T}\times\mathcal{F}}\,. \tag{4.14}
$$

4.2.2 Structuring a Piano Roll as a Tree

We have now a way of creating piano rolls that relies on applying the morphological dilation on activations and harmonic textures that share a set of indexes (that we called J). However, this is not very practical for creating bigger structures like complete pieces of music. A mere set of indexes J dos not tell that much about the structure of a piece and how it is organized.

This is why we develop in this section a more refined strategy, that is organizing the piano roll as a tree.

Hierarchical structures are omnipresent in music analysis and composition; it is the core of the generative grammar (Chomsky, 1965) that impulsed the Generative Theory of Tonal Music (Lerdahl, 1983). In particular, hierarchy is used to explain structure in music (Lerdahl & Jackendoff, 1983), and its representation in form of a tree is a very used one (Marsden et al., 2013; Orio & Roda, 2009; Koelsch et al., 2013; Carnovalini et al., 2021).

In this section, we are going to organize the structure of a musical piece in a tree formed by activations and piano rolls. This organization will permit us to understand better the fragments and their relative role. We call this tree a *score tree*.

Definition 4.27 (Score tree)**.** We call S a **score tree** of domain $\mathcal{T}_S \times \mathcal{F}_S$ if

$$S = \big((A_1, B_1), (A_2, B_2), ..., (A_N, B_N)\big) = \big((A_n, B_n)\big)_{n=1}^{N} \tag{4.15}$$

with $A_n \in \mathcal{A}_2^{\mathcal{T}_A \times \mathcal{F}_A}$ and B_n being either a harmonic texture $B_n \in \mathcal{A}_3^{\mathcal{T}_B \times \mathcal{F}_B}$, either a score tree of domain $\mathcal{T}_B \times \mathcal{F}_B$, such that $\mathcal{T}_A \times \mathcal{F}_A$ and $\mathcal{T}_B \times \mathcal{F}_B$ are compatible with output $\mathcal{T}_S \times \mathcal{F}_S$.

The way of transforming a score tree into a piano roll is by using recursively the morphological dilation. Indeed, the piano roll P_S associated to the score tree $S = \big((A_n, B_n)\big)_{n=1}^{N}$ is given by the formula

$$P_S = \bigvee_{n=1}^{N} A_n \oplus P_{B_n} \tag{4.16}$$

where P_{B_n} is either B_n if it is a harmonic texture or the piano roll associated to B_n if it is a score tree.

In the following, we make the abuse of notation

$$S = A_1 \oplus B_1 \vee A_2 \oplus B_2 \vee ... \vee A_N \oplus B_N$$

that consists on identifying a score tree with its resulting piano roll.

Let us show an example of such a decomposition of a score into a score tree.

Example 4.28. *We call S the score tree representing first movement of Mozart's Piano Sonata No.16 in C major, K.545. We divide its structure in two parts corresponding to the sections enclosed by repeat signs, i.e., Part 1 consists on mm. 1-28 and Part 2 in mm. 29-73. The corresponding score tree description is*

$$S = \Big(\{(0,0),(28,0)\} \oplus \text{Part 1} \vee \{(56,0),(100,0)\} \oplus \text{Part 2}\Big) \tag{4.17}$$

$$\oplus \big((1,1,0), \text{C4}\big) \tag{4.18}$$

where we factor out the point-like elements $(1,1,0)$ *and* C4*.*

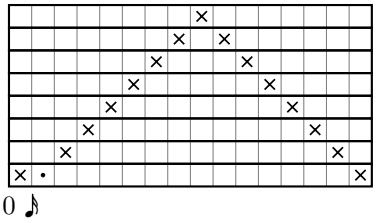
We can divide even further the score in the following way

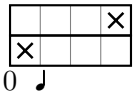
$$\begin{aligned} S = \Big(&\{(0,0),(28,0)\} \oplus (\{(0,0)\} \oplus \text{Exp.}) \vee \\ &\{(56,0),(100,0)\} \oplus (\{(0,0)\} \oplus \text{Dev.} \vee \{(13,0)\} \oplus \text{Rec.})\Big) \\ \oplus &\big((1,1,0), \text{C4}\big) \end{aligned}$$

where Exp.[14] consists on mm. 1-28, Dev.[14] on mm. 29-41 and Rec.[14] on mm. 42-73.

The split can go even further, but writing it in mathematical form is cumbersome. We choose rather a diagram shown in Figure 4.15. While the leaves in the tree are not precisely harmonic textures, they exhibit a certain textural similarity that allows the possibility of generating them with few textures and harmonies.

For instance, let us show how to render A_2 as a score tree. We can indeed decompose it as in Figure 4.16 with

$T_1 =$ 

$T_2 =$ 

and

$$
\begin{aligned}
C_1 &= \mathrm{Maj}_6^{13}, & H_1 &= (\{-4\}, \{5, 12\}),\\
C_2 &= \mathrm{Maj}_5^{12}, & H_2 &= (4, 12\}, \{4, 12\}),\\
C_3 &= \mathrm{Maj}_4^{11}, & H_3 &= (2, 12\}, \{2, 11\}),\\
C_4 &= \mathrm{Maj}_3^{10}, & H_4 &= (0, 12\}, \{0, 4\}).
\end{aligned}
$$

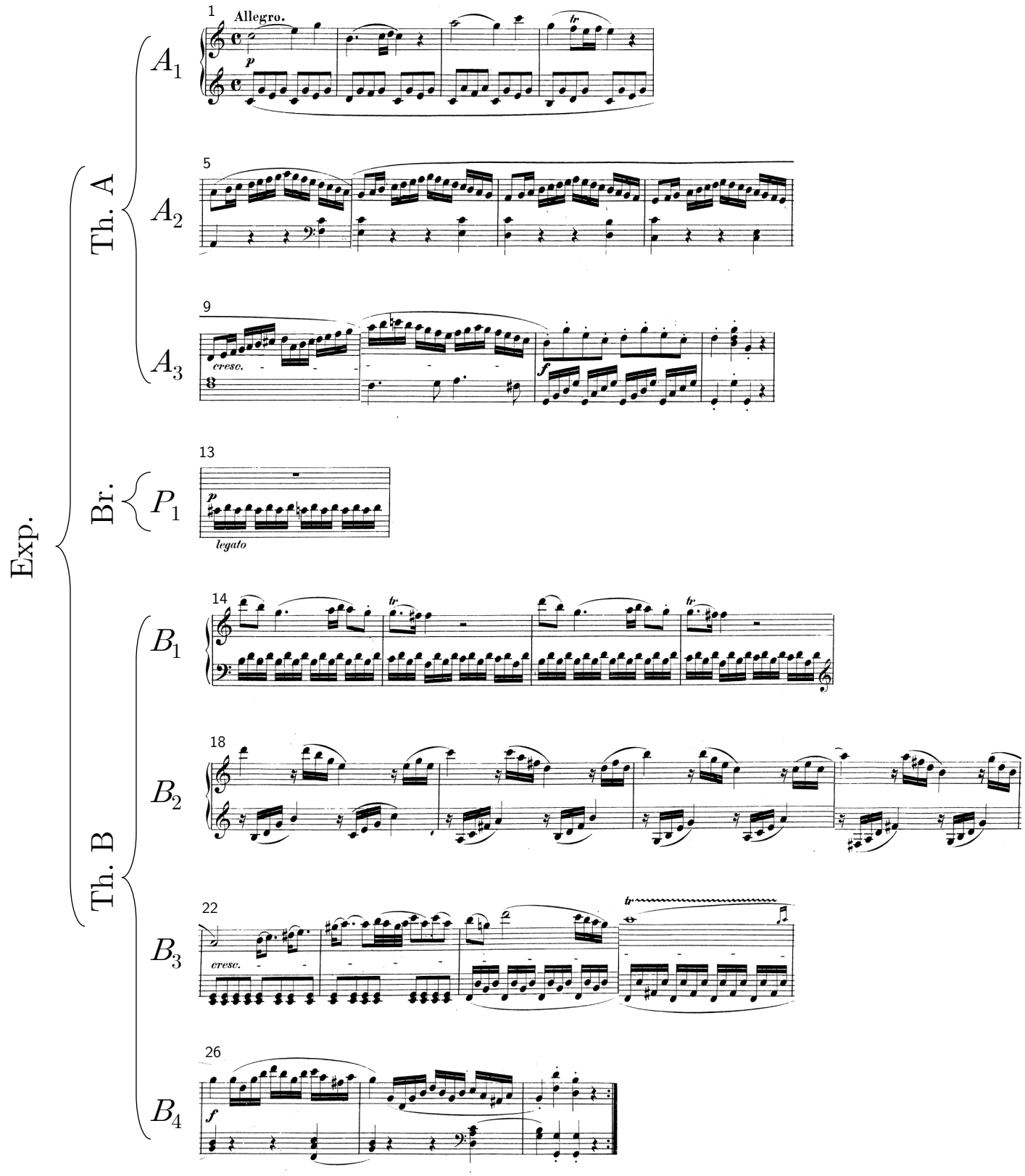
with Maj_a^b *being the chord formed by taking the elements* a^{th} *to* b^{th} *from the list*

$$\mathrm{Maj} = (0, 2, 4, 5, 7, 9, 11, 12, 14, 16, 17, 19, 21)\,.$$

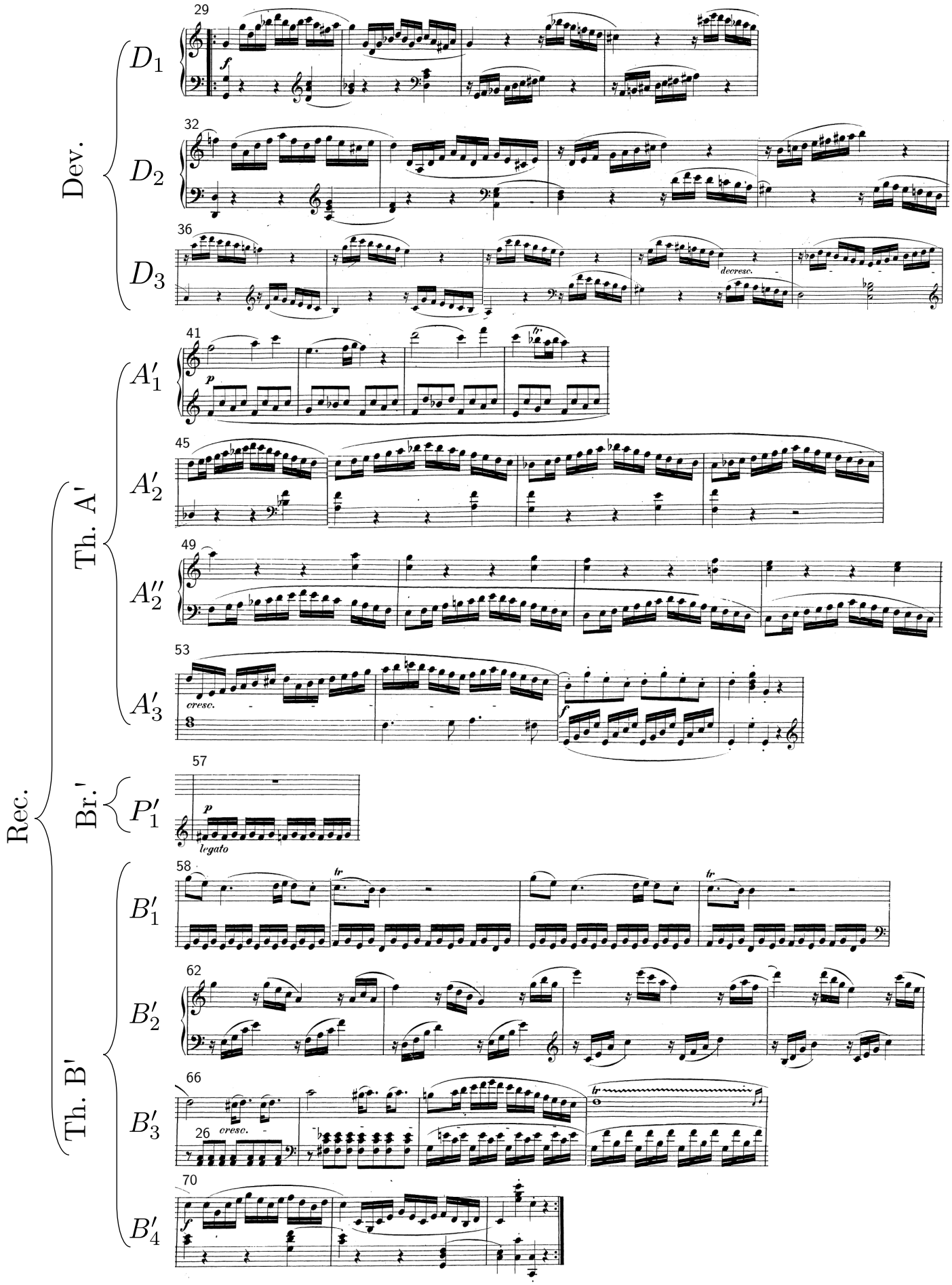
[14]The labels Exp., Dev. and Rec. stand for the sections of the sonata form, namely Exposition, Development and Recapitulation respectively.

Sonata No.16 in C major, K.545

Part 1



Part 2


Rec.
Dev.
D_1
D_2
D_3
Th. A'
A'_1
A'_2
A''_2
A'_3
Br.'
P'_1
Th. B'
B'_1
B'_2
B'_3
B'_4
29
32
36
41
45
49
53
57
58
62
66
70
f
p
tr
decresc.
cresc.
legato
26

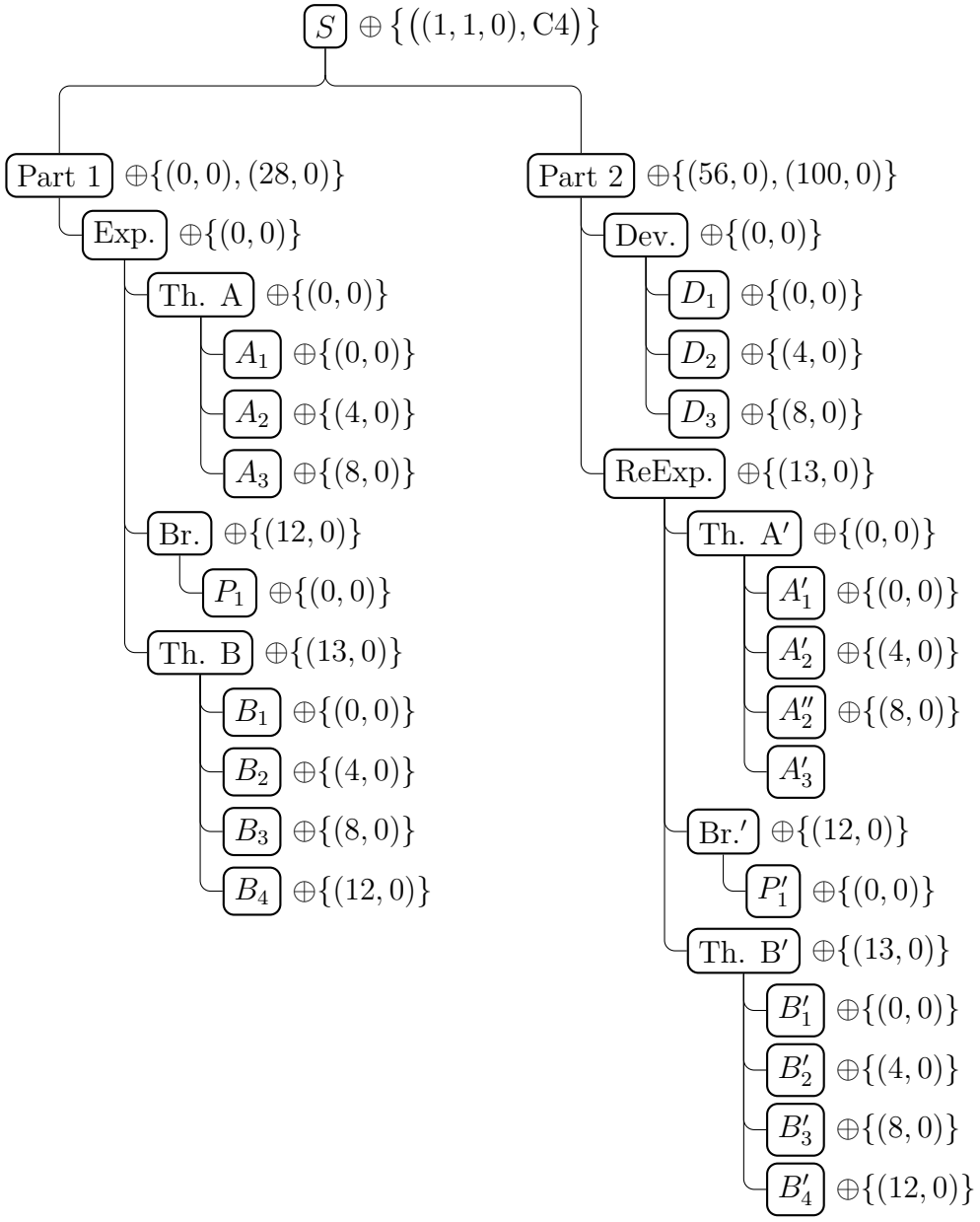


Figure 4.15: Formal structure of Mozart's Piano Sonata No.16 in C major, K.545.

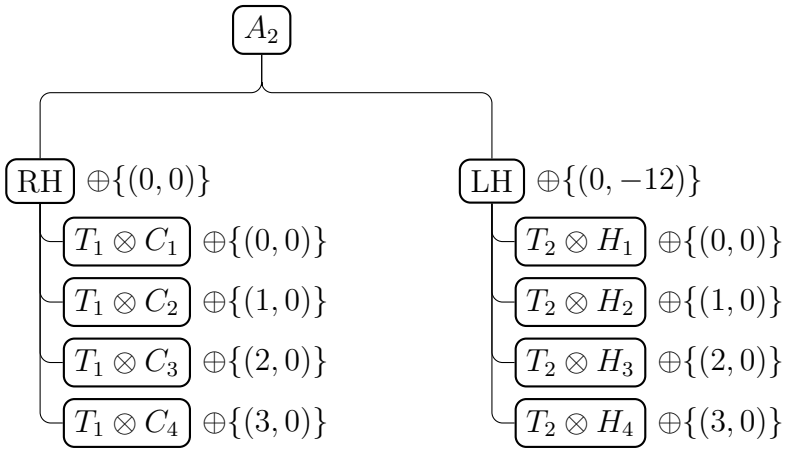


Figure 4.16: Score tree of A_2

4.2.3 Computational Implementation

This framework allows us to generate piano rolls by using harmonic textures organized in score trees. In this section, we show how to implement this model computationally in an object-oriented language. In particular, we use two languages that will interact: XML and Python.

The idea is to describe the score tree as an XML file (that we call ScoreXML) with a XML Schema Definition (XSD) associated that validates it. Once the score tree is written, it can be compiled by a Python script to produce a piano roll (using morphological dilation). This piano roll, which is a Python object, can then be plotted[15], transformed into a MIDI file or, in future extensions, written as a MusicXML file.

The full code is available in the repository that accompanies this thesis[16], so we limit ourselves to expose a reduced description of the objects (that serve both for the XML and for the Python objects). We recall that in Python the methods `__neg__`, `__add__`, `__mul__` override the operators `-`, `+` and `*`, respectively. We also precise that `frac` stands for fractional numbers and `int` for integers.

We expose first the most basic elements of our model: time and frequency.

```
Time: {
    value: frac
}
```

[15] All the figures that represent piano rolls in this thesis are done with this method.

[16] https://github.com/Manza12/MMM

```
Frequency: {
    value: int
}
TimeShift(Time): {
    __neg__(self) -> TimeShift
    __add__(self, time: TimeShift) -> TimeShift
}
FrequencyShift(Frequency): {
    __neg__(self) -> FrequencyShift
    __add__(self, frequency: FrequencyShift) -> FrequencyShift
}
TimePoint(Time): {
    __add__(self, time: TimeShift) -> TimePoint
}
FrequencyPoint(Frequency): {
    __add__(self, frequency: FrequencyShift) -> FrequencyPoint
}
TimeFrequency: {
    time: Time
    frequency: Frequency
}
```

With these objects we can already define a piano roll; indeed, we can implement the lattice multiplication $\bullet : \mathcal{A}_3 \times \mathcal{A}_2 \to \mathcal{A}_3$ by using integer numbers with the product if we set $\mathcal{A}_2 = \{0, 1\}$ and $\mathcal{A}_3 = \{0, 1, 2\}$.

```
PianoRoll: {
    array: int[M][N]
    origin: TimeFrequency
    tatum: TimeShift
    __add__(self, piano_roll: PianoRoll) -> PianoRoll
}
```

Then, we can create the objects for creating harmonic textures.

```
Hit: {
    start: Time
    duration: TimeShift
}
Rhythm: {
```

```
    hits: List[Hit]
}
Texture: {
    rhythms: List[Rhythm]
    __mul__(self, harmony: Harmony) -> HarmonicTexture
    __mul__(self, chord: Chord) -> ChordTexture
}
Chord: {
    notes: List[Frequency]
}
Harmony: {
    chords: List[Chord]
}
HarmonicTexture(PianoRoll): {
    texture: Texture
    harmony: Harmony
}
ChordTexture(HarmonicTexture): {
    texture: Texture
    chord: Chord
}
```

Finally, we can define the score tree. We need for that to define also the activations.

```
Activations: {
    values: List[TimeFrequency]
    __add__(self, piano_roll: PianoRoll) -> PianoRoll
}
ScoreTree: {
    components: List[
        Tuple[Activations, Union[HarmonicTexture, ScoreTree]]
    ]
    to_piano_roll(self) -> PianoRoll
}
```

Chapter 5

Mathematical Morphology Applied to Analyze Piano Rolls

The analysis of piano rolls using mathematical morphology was the first application of MM to music (Karvonen & Lemström, 2008; Karvonen, 2008; Karvonen et al., 2011; Lascabettes, 2019; Lascabettes et al., 2020; Lascabettes et al., 2022). The primary focus of these works has been on motif detection. In this thesis, we extend the analysis by incorporating considerations related to texture and harmony. An attempt to analyze harmony using MM has already been made in a previous study (Romero-García, Bloch, et al., 2022).

In the previous chapter, we proposed a description of a piano roll $P \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ as a combination of activations of harmonic textures by the formula[1]

$$P = \mathscr{A} \oplus \mathscr{T} \otimes \mathscr{H} \tag{5.1}$$

where $\mathscr{A} \in \mathcal{A}_A^{\mathcal{T}_A\times\mathcal{F}_A\times J}$, $\mathscr{T} \in \mathcal{A}_T^{\mathcal{T}_T\times I_j\times J}$, $\mathscr{H} \in \mathcal{A}_H^{\mathcal{F}_H\times I_j\times J}$ are such that $(\mathcal{A}_A, \mathcal{A}_T, \mathcal{A}_H)$ form a residuated triplet, $(\mathcal{T}_A, \mathcal{T}_T)$ and $(\mathcal{F}_A, \mathcal{F}_H)$ are compatible (see Definition 4.26), and J and I_j, $\forall j \in J$, are sets of indices.

This framework has enabled us to establish a method for constructing music using elementary components known as harmonic textures. These textures are positioned within the piano roll by means of morphological dilation between them and the activations. This approach highlights two fundamental parameters of music: texture and harmony.

In this chapter, our objective is to perform the reverse operation: can we extract the underlying parameters from a given piano roll? In mathematical terms, this can

[1]We recall that $\otimes$ has priority over $\oplus$.

be reformulated as follows: given a piano roll denoted as P, can we determine the values of $\mathscr{A}$, $\mathscr{T}$, and $\mathscr{H}$ in such a way that Equation (5.1) is satisfied?

Indeed, this task is highly challenging. It is not even well-posed due to the existence of a trivial solution where a single texture, harmony, and activation could be assigned to each note. The more pertinent question is: among the countless possible choices, which values of $\mathscr{A}$, $\mathscr{T}$, and $\mathscr{H}$ would constitute the optimal solution?

Providing a definitive answer to this question is not within our expectations, as we believe that a unique solution might not exist. Instead, in the upcoming sections, we delve into the difficulties, theoretical considerations, and practical challenges we encounter in this endeavor.

The chapter is structured into three main sections. In Section 5.1, we employ morphological erosion to analyze piano rolls, specifically choosing harmonic textures as structuring elements. In Section 5.2, we continue to employ erosion, this time with textures as structuring elements. This analytical approach to piano rolls presents several challenges that we address by using graphs. The payoff of this approach is significant, leading to the compression of piano rolls into vertical chords. With these chord sequences as input, we delve into Section 5.3, where we once again employ erosion, this time using harmonies as structuring elements. The interpretation of the erosion proves to be highly productive, enabling us to tackle the Roman numeral analysis problem using this technique and once again making use of graphs.

5.1 Analyzing Piano Rolls with Harmonic Textures

We begin by examining what insights the theory of mathematical morphology can offer regarding a piano roll generated with Equation (5.1). To enhance clarity, we will accompany each step with illustrative examples.

Let $P \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ be a piano roll. In this section, we do not delve into the particular attributes of $\mathcal{A}$, $\mathcal{T}$, and $\mathcal{F}$. Instead, we assume that these aspects have been appropriately determined. We recall that $(\mathcal{A}^{\mathcal{T}\times\mathcal{F}}, \preceq)$ is a complete lattice. Furthermore, we assume that $(\mathcal{A}, \mathcal{A}, \mathcal{A})$ is a residuated triplet and that $\mathcal{T} \times \mathcal{F}$ is compatible with itself.

In this section, our objective is to find $\mathscr{A} \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$ and $\mathscr{B} \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$ such that $P = \mathscr{A} \oplus \mathscr{B}$. Similar research is currently be done by Lascabettes, 2023.

To approach this task, we can begin by employing the adjoint operator of dilation $\oplus$, which is the erosion $\ominus$. For instance, if we let $B_1 \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$, applying the erosion

to P results in:

$$P \ominus B_1 := A_1 \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}. \tag{5.2}$$

Since $\oplus$ and $\ominus$ are adjoint operators forming a dilation/erosion pair, we know that $\bullet := \oplus \circ \ominus$ is a closing and $\circ := \ominus \circ \oplus$ is an opening[2]. Consequently, owing to the anti-extensivity property of the opening, we can assert that

$$A_1 \oplus B_1 = (P \ominus B_1) \oplus B_1 = P \circ B_1 \preceq P\,. \tag{5.3}$$

We can proceed by selecting another $B_2 \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ and repeating the process to obtain

$$A_2 \oplus B_2 = (P \ominus B_2) \oplus B_2 = P \circ B_2 \preceq P\,. \tag{5.4}$$

with $A_2 := P \ominus B_2$.

At this point, we can stack B_1 and B_2 into $\mathscr{B}$; to achieve this, we define $\mathscr{B} = (B_1, B_2) \in \left(\mathcal{A}^{\mathcal{T}\times\mathcal{F}}\right)^J \simeq \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$, with $J = \{1, 2\}$. Similarly, we can stack A_1 and A_2 into $\mathscr{A} = (A_1, A_2) \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$.

However, attempting to directly define an erosion $P \ominus \mathscr{B}$ faces compatibility issues due to the differing domains $\mathcal{T} \times \mathcal{F}$ and $\mathcal{T} \times \mathcal{F} \times J$. To address this issue, we introduce the trivial group $(\mathbf{0}, +)$ with $\mathbf{0} = \{0\}$, which acts on any set X through the action $+ : X \times \mathbf{0} \to X$, $(x, 0) \mapsto x$. In this context, we apply this action to the set J. Additionally, we use the canonical bijection

$$\begin{array}{rccl} \iota : & \mathcal{A}^{\mathcal{T}\times\mathcal{F}} & \to & \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times\mathbf{0}} \\ & P & \mapsto & P := \iota(P) : \begin{array}{ccc} \mathcal{T}\times\mathcal{F}\times\mathbf{0} & \to & \mathcal{A} \\ (t,\xi,0) & \mapsto & P(t,\xi) \end{array} \end{array}. \tag{5.5}$$

Since the domains $\mathcal{T} \times \mathcal{F} \times \mathbf{0}$ and $\mathcal{T} \times \mathcal{F} \times J$ are now compatible, the erosion $P \ominus \mathscr{B} \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$ becomes well defined. Given the isomorphism, we can simplify the statement by noting that the domain of the erosion is simply $\mathcal{A}^{\mathcal{T}\times\mathcal{F}}$. Recalling that $\mathscr{B} = (B_j)_{j\in J}$, the explicit definition of this erosion is given by

$$\begin{array}{rccl} \varepsilon_{\mathscr{B}} : & \mathcal{A}^{\mathcal{T}\times\mathcal{F}} & \to & \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J} \\ & P & \mapsto & (P \ominus B_j)_{j\in J} \end{array}. \tag{5.6}$$

Before delving into the applications of these erosions (which will be used extensively throughout this chapter), let us first define the adjoint dilation of this erosion.

[2]The symbol for opening and composition is the same ($\circ$) but will be discernible from the context.

Proposition 5.1. *Let $\varepsilon_{\mathscr{B}} : \mathcal{A}^{\mathcal{T}\times\mathcal{F}} \to \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$ be the erosion defined in Equation (5.6), with $\mathscr{B} = (B_j)_{j\in J} \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$. We define $\mathscr{A} = (A_j)_{j\in J} \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$. Then, the adjoint operator of $\varepsilon_{\mathscr{B}}$ is the dilation*

$$\begin{array}{rccc} \delta_{\mathscr{B}} : & \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J} & \to & \mathcal{A}^{\mathcal{T}\times\mathcal{F}} \\ & \mathscr{A} & \mapsto & \bigvee_{j\in J}(A_j \oplus B_j) \end{array} . \tag{5.7}$$

Proof. We know by Theorem 1.10 that since $\varepsilon_{\mathscr{B}} : \mathcal{A}^{\mathcal{T}\times\mathcal{F}} \to \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$ is an erosion there is an adjoint dilation given by the formula

$$\begin{array}{rccc} \delta : & \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J} & \to & \mathcal{A}^{\mathcal{T}\times\mathcal{F}} \\ & \mathscr{A} & \mapsto & \bigwedge\{P \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}} : \mathscr{A} \preceq \varepsilon_{\mathscr{B}}[P]\} \end{array} . \tag{5.8}$$

We shall prove that $\bigvee_{j\in J}(A_j \oplus B_j) = \bigwedge\{P \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}} : \mathscr{A} \preceq \varepsilon_{\mathscr{B}}[P]\}$. By replacing $\varepsilon_{\mathscr{B}}[P]$ by $P \ominus \mathscr{B}$, let us call $\mathscr{P} = \{P \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}} : \mathscr{A} \preceq P \ominus \mathscr{B}\}$. We are proving that $\bigvee_{j\in J}(A_j \oplus B_j) \in \mathscr{P}$ and that $\forall P \in \mathscr{P}$, $\bigvee_{j\in J}(A_j \oplus B_j) \preceq P$, which means that $\bigvee_{j\in J}(A_j \oplus B_j) = \bigwedge \mathscr{P}$ and finishes the proof.

$\boxed{\bigvee_{j\in J}(A_j \oplus B_j) \in \mathscr{P}}$

Indeed, $\forall j_0 \in J$,

$$\begin{aligned} A_{j_0} &\preceq A_{j_0} \bullet B_{j_0} \\ &= (A_{j_0} \oplus B_{j_0}) \ominus B_{j_0} \\ &\preceq \left(\bigvee_{j\in J}(A_j \oplus B_j)\right) \ominus B_{j_0} \end{aligned}$$

which means that $\mathscr{A} \preceq \left(\bigvee_{j\in J}(A_j \oplus B_j)\right) \ominus \mathscr{B}$ and thus $\bigvee_{j\in J}(A_j \oplus B_j) \in \mathscr{P}$.

$\boxed{\forall P \in \mathscr{P},\ \bigvee_{j\in J}(A_j \oplus B_j) \preceq P}$

$\forall P \in \mathscr{P}$,

$$\begin{aligned} \mathscr{A} \preceq P \ominus \mathscr{B} &\Rightarrow \forall j \in J,\ A_j \preceq P \ominus B_j \\ &\Rightarrow \forall j \in J,\ A_j \oplus B_j \preceq (P \ominus B_j) \oplus B_j \\ &\Rightarrow \forall j \in J,\ A_j \oplus B_j \preceq P \circ B_j \preceq P \\ &\Rightarrow \bigvee_{j\in J}(A_j \oplus B_j) \preceq P\,. \end{aligned}$$

□

Now we have erosion and dilation defined, and thus opening and closing. By letting $P \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ being a piano roll, $\mathscr{B} \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$ a stack of structuring elements, and $\mathscr{A} \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$ a stack of activations, we can write

$$\begin{aligned} P \ominus \mathscr{B} &\in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J} \quad & \mathscr{A} \oplus \mathscr{B} &\in \mathcal{A}^{\mathcal{T}\times\mathcal{F}} \\ P \circ \mathscr{B} &\in \mathcal{A}^{\mathcal{T}\times\mathcal{F}} & \mathscr{A} \bullet \mathscr{B} &\in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}\,. \end{aligned} \tag{5.9}$$

Let us return to the process of adding more B_j to $\mathscr{B}$ with the property $\forall j \in J$, $P \circ B_j \preceq P$ which can be restated using Equation (5.9) as $P \circ \mathscr{B} \preceq P$. At some point, it *might*[3] happen that $P \circ \mathscr{B} = P$, allowing us to potentially conclude the search.

Let us illustrate this process with an example. For the examples, we make the following choices for time, frequency and amplitude: our space is $\mathcal{T}_{\mathbf{c}}^{\mathbf{c}} \times \mathcal{N}$ with its corresponding group $(\mathbb{Q} \times \mathbb{Z}, +)$, and the residuated triplet is given by the lattice multiplication $\bullet : \mathcal{A}_3 \times \mathcal{A}_2 \to \mathcal{A}_3$. This means that $P \in \mathcal{A}_3^{\mathcal{T}_{\mathbf{c}}^{\mathbf{c}}\times\mathcal{N}}$, $\mathscr{B} \in \mathcal{A}_3^{\mathbb{Q}\times\mathbb{Z}\times J}$, leading to $\mathscr{A} \in \mathcal{A}_2^{\mathcal{T}_{\mathbf{c}}^{\mathbf{c}}\times\mathcal{N}\times J}$.

Now, let us consider the piano roll P in Figure 5.1, which represents the first two measures of the third movement of Beethoven's Piano Sonata No.14, Op.27 No.2.

We want to evaluate the different choices for $\mathscr{B} = (B_j)_{j\in J}$ such that $P = P \circ \mathscr{B}$. To measure the quality of these choices, we introduce the concept of *redundancy* of a decomposition. For this purpose, we first define a measure for a piano roll with amplitude ranges $\mathcal{A}_2$ or $\mathcal{A}_3$.

Definition 5.2 (Measure of a piano roll)**.** Let $P \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ be a piano roll. For all $a \in \mathcal{A}$, let $\mu_a : \Sigma_{\mathcal{T}\times\mathcal{F}} \to \mathbb{R}^+$ be a measure, where $\Sigma_{\mathcal{T}\times\mathcal{F}}$ is a σ-algebra over $\mathcal{T} \times \mathcal{F}$. Then, we define the **measure** of P relative to $\{\mu_a\}_{a\in\mathcal{A}}$ as

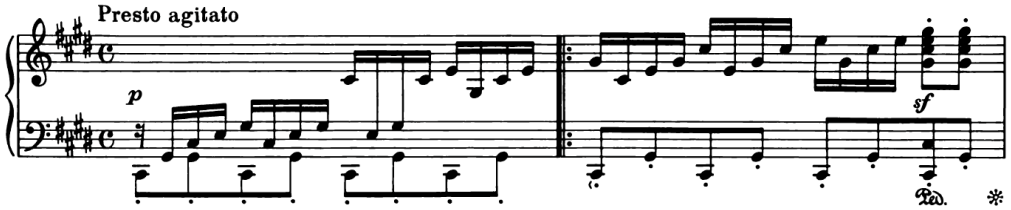
$$|P| = \sum_{a\in\mathcal{A}} \mu_a(\{(t,\xi) \in \mathcal{T} \times \mathcal{F} : P(t,\xi) = a\}) \in \mathbb{R}^+\,. \tag{5.10}$$

It is evident by the definition that the measure of a piano roll is an increasing function from $(\mathcal{A}^{\mathcal{T}\times\mathcal{F}}, \preceq)$ to $(\mathbb{R}^+, \leq)$, i.e., $P_0 \preceq P_1 \Rightarrow |P_0| \leq |P_1|$.

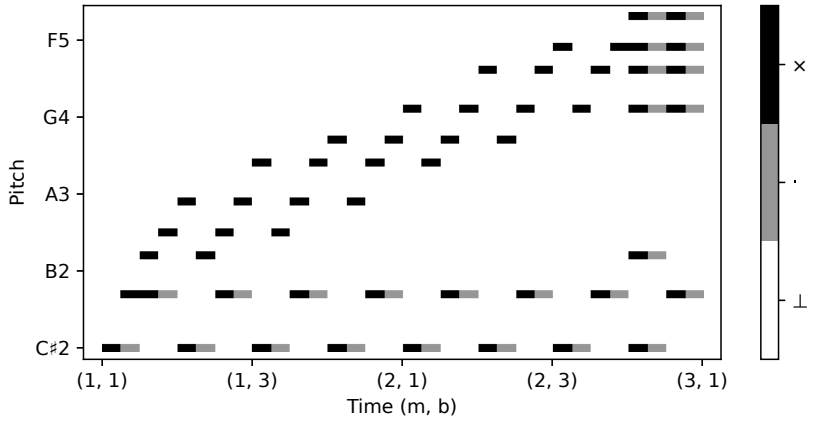
The choice of $\{\mu_a\}_{a\in\mathcal{A}}$ depends on the choices of $\mathcal{T}$, $\mathcal{F}$ and $\mathcal{A}$. When dealing with activations piano rolls, i.e., $A \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{F}}$, we use

$$|A| = \mu_\delta(\mathrm{supp}(A)) \tag{5.11}$$

[3]The emphasized term *might* here indicates the possibility that the search might not reach a conclusion if the selection of the B_j is not appropriate. Furthermore, even when the search does conclude, there is no guarantee that we obtain the desired elements.



(a) Score



(b) Piano roll

Figure 5.1: Third movement of Beethoven's Piano Sonata No.14, Op.27 No.2 mm. 1-2.

where μ_δ is the discrete measure defined by $\mu_\delta(X) = |X|$, $\forall X \subseteq \mathcal{T} \times \mathcal{F}$.

However, when using a piano roll with amplitude represented as the rhythmic range $\mathcal{A}_3$, we use a different measure. Let $P \in \mathcal{A}_3^{\mathcal{T}_\circ^{\mathbf{c}} \times \mathcal{N}}$. Since the time-frequencies $(t, \xi) \in \mathcal{T}_\circ^{\mathbf{c}} \times \mathcal{N}$ where $P(t, \xi)$ is the attack ($\times$) form a discrete set, we use a discrete measure for them. However, for the values where $P(t, \xi)$ is the sustain ($\cdot$) we use a different measure: the product measure between the Lebesgue measure induced by the isomorphism between $\mathbb{Q}$ and $\mathcal{T}_\circ^{\mathbf{c}}$, and μ_δ, the discrete measure presented before. We call it μ. This results in the following measure definition:

$$|P| = c_\times \cdot \mu_\delta(\{(t, \xi) \in \mathcal{T} \times \mathcal{F} : P(t, \xi) = \times\}) + c_\cdot \cdot \mu(\text{supp}(P)) \tag{5.12}$$

where $c_\times, c_\cdot \in \mathbb{R}^+$ are the coefficients that control the relative weight given to the measure of attacks and the measure of sustains, respectively.

Now let us compute the measure of the piano roll shown in Figure 5.1. As there

are 25 ♪ and 27 ♪, the measure of the attacks is $52c_\times$, and the measure of the sustains is $(25 \times \frac{1}{8} + 27 \times \frac{1}{16})c.$, leading to a total measure of P:

$$|P| = c_\times \cdot 52 + c. \cdot \left(\frac{25}{8} + \frac{27}{16}\right) = 52c_\times + \frac{77}{16}c. \,.$$

With this framework in place, we can introduce the concept of redundancy.

Definition 5.3 (Redundancy)**.** Let $P \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}}$ and $\mathscr{B} \in \mathcal{A}^{\mathcal{T}\times\mathcal{F}\times J}$. We define the **redundancy** of $\mathscr{B}$ over P denoted by $\rho(\mathscr{B}, P)$ as

$$\rho(\mathscr{B}, P) = \frac{\left(\sum_{j\in J} |P \ominus B_j| \cdot |B_j|\right) - |P|}{|P|} \in \mathbb{R}\,, \tag{5.13}$$

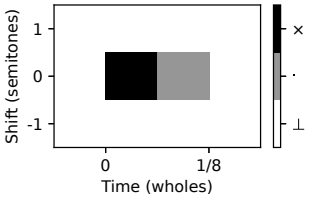
where $|\cdot|$ is defined as in Equation (5.12).

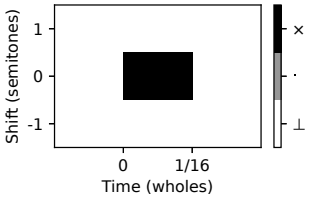
The redundancy may be negative. Indeed, if we have $P \circ \mathscr{B} \preceq P$ it may happen that $\sum_{j\in J} |P \ominus B_j| \cdot |B_j| < |P|$. When $P \circ \mathscr{B} = P$, the redundancy is positive and we give it in percentage. Moreover, the redundancy is not limited to 100 %.

While the redundancy relies on the choice of $(c_\times, c.) \in (\mathbb{R}^+)^2$, to gain an overview, we can use two reference sets of coefficients: $(c_\times, c.) = (1, 0)$ and $(c_\times, c.) = (0, 1)$. We denote the resulting redundancies as $\rho_\times$ and $\rho.$, respectively.

Now that we have established a measure to assess the efficiency of different choices for $\mathscr{B}$, we can proceed to evaluate a couple of scenarios. Specifically, we present two examples: one involving a trivial choice and another involving a more effective one.

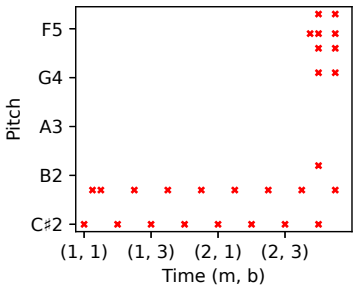
The trivial choice is to select $\mathscr{B} = (B_j)_{j=1}^2$, where we choose the note values that are already present in the score. That is,

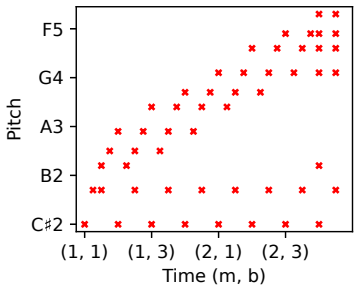
$B_1 =$ 

$B_2 =$ 

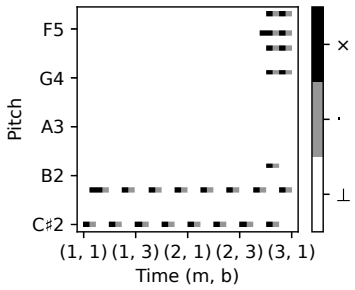
.

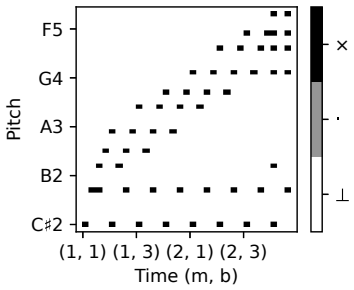
The resulting $A_j = P \ominus B_j$ are

$A_1 =$ 

$A_2 =$ 

and the resulting $A_j \oplus B_j$ are

$A_1 \oplus B_1 =$ 

$A_2 \oplus B_2 =$ 

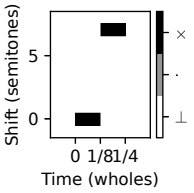
where we have $P \circ \mathscr{B} = (A_1 \oplus B_1) \vee (A_2 \oplus B_1) = P$.

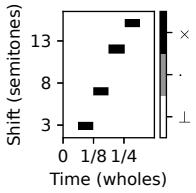
We can now compute the corresponding redundancies:

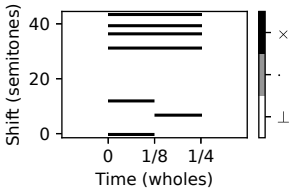
$$\begin{aligned}
\rho_\times &= \frac{\sum_{j=1}^{2} |A_j| \cdot \mu_\times(B_j) - \mu_\times(P)}{\mu_\times(P)} \\
&= \frac{27 \cdot 1 + 52 \cdot 1 - 52}{52} \\
&= \frac{27}{52} \\
&\approx 51.9\,\%
\end{aligned}
\qquad
\begin{aligned}
\rho_\cdot &= \frac{\sum_{j=1}^{2} |A_j| \cdot \mu_\cdot(B_j) - \mu_\cdot(P)}{\mu_\cdot(P)} \\
&= \frac{27 \cdot \frac{1}{16} + 52 \cdot \frac{1}{16} - \frac{77}{16}}{\frac{77}{16}} \\
&= \frac{29}{77} \\
&\approx 37.7\,\%
\end{aligned}$$

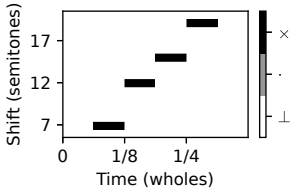
where $\mu_\times$ and $\mu_\cdot$ are the measures obtained by setting $(c_\times, c_\cdot)$ to $(1, 0)$ and $(0, 1)$, respectively.

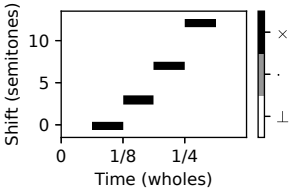
Now, let us choose $\mathscr{B} = (B_j)_{j=1}^5$ with

$B_1 =$ 

$B_2 =$ 

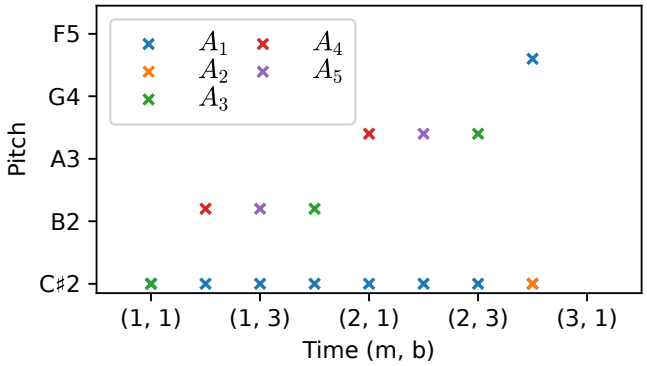
$B_3 =$ 

$B_4 =$ 

$B_5 =$ 

.

We have that $\mathscr{A} = (A_j)_{j \in J} = P \ominus \mathscr{B}$ is

$\mathscr{A} =$ 

The redundancies are

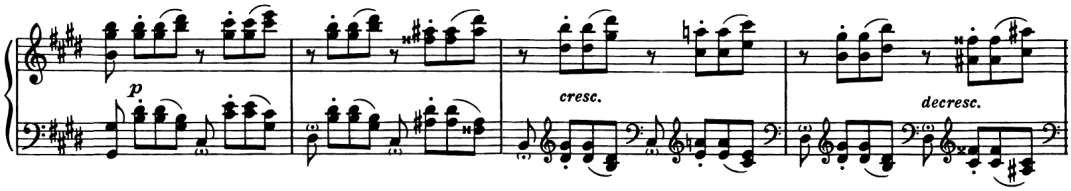
$$\rho_\times = \frac{\sum_{j=1}^5 |A_j| \cdot \mu_\times(B_j) - \mu_\times(P)}{\mu_\times(P)} = \frac{9 \cdot 2 + 1 \cdot 11 + 3 \cdot 4 + 2 \cdot 4 + 2 \cdot 4 - 52}{52}$$
$$= \frac{5}{52} \approx 9.6\,\%$$
$$\rho_\cdot = \frac{\sum_{j=1}^5 |A_j| \cdot \mu_\cdot(B_j) - \mu_\cdot(P)}{\mu_\cdot(P)} = \frac{9 \cdot \frac{1}{4} + 1 \cdot \frac{11}{8} + 3 \cdot \frac{1}{4} + 2 \cdot \frac{1}{4} + 2 \cdot \frac{1}{4} - \frac{77}{16}}{\frac{77}{16}}$$
$$= \frac{9}{77} \approx 11.7\,\%\,.$$

These redundancies are fairly small. To reduce them to the minimum, we are using an alternative approach that involves modifying $\mathscr{A}$ to contain fewer activations. This strategy will be explored in the following section.

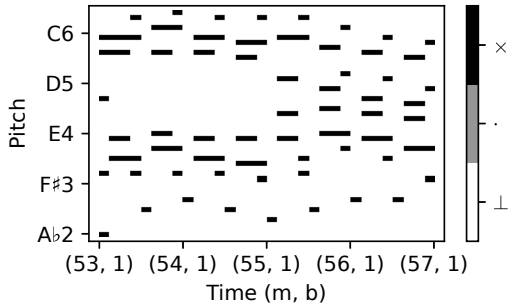
5.2 Analyzing Piano Rolls with Textures

In the previous section, we showed how to use erosion and a measure to analyze a piano roll P using a stack of harmonic textures $\mathscr{B}$. This involved searching for $\mathscr{A}$ and $\mathscr{B}$ such that $P = \mathscr{A} \oplus \mathscr{B}$. However, our original intention was to take this a step further: to find $\mathscr{A}$, $\mathscr{T}$ and $\mathscr{H}$ such that $P = \mathscr{A} \oplus \mathscr{T} \otimes \mathscr{H}$.

In this section, we attempt to find these elements, with a specific focus on the texture component. Let us consider an example involving measures from later in the same musical piece, as depicted in Figure 5.2.



(a) Score

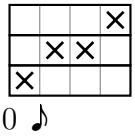


(b) Piano roll

Figure 5.2: Third movement of Beethoven's Piano Sonata No.14, Op.27 No.2 mm. 53-56.

This excerpt can be built by using a single texture applied to several harmonies.

If we choose the texture

$T =$ 

and the harmonies

$$\begin{aligned}
H_1 &= (\{-12, 0, 15, 24, 27\}, \{3, 7, 24, 27\}, \{0, 3, 27, 31\}) \\
H_2 &= (\{-7\}, \{5, 8, 24, 29\}, \{0, 5, 29, 32\}) \\
H_3 &= (\{-5\}, \{3, 7, 24, 27\}, \{0, 3, 27, 31\}) \\
H_4 &= (\{-7\}, \{2, 7, 23, 26\}, \{-1, 2, 26, 31\}) \\
H_5 &= (\{-9\}, \{7, 12, 19, 27\}, \{0, 3, 27, 31\}) \\
H_6 &= (\{-7\}, \{8, 13, 17, 25\}, \{5, 8, 20, 29\}) \\
H_7 &= (\{-5\}, \{7, 12, 15, 24\}, \{3, 7, 19, 27\}) \\
H_8 &= (\{-5\}, \{5, 11, 14, 23\}, \{-1, 5, 17, 26\}) .
\end{aligned}$$

we have that

$$P = \mathscr{A} \oplus \mathscr{T} \otimes \mathscr{H} \in \mathcal{A}_3^{\mathcal{T}_{\mathbf{c}} \times \mathcal{N}} \tag{5.14}$$

where $J = \{1, 2, ..., 8\}$, $I = \{1, 2, 3\}$ and

$$\begin{aligned}
\mathscr{A} &= (A_j)_{j \in J} = \left(\left((53, 1, 0) + \frac{j-1}{2}, \mathrm{G}\sharp 3 \right) \right)_{j=1}^{8} \in \mathcal{A}_2^{\mathcal{T}_{\mathbf{c}} \times \mathcal{N} \times J} \\
\mathscr{T} &= (T)_{j=1}^{8} \in \mathcal{A}_3^{\mathbb{Q} \times I \times J} \\
\mathscr{H} &= (H_j)_{j=1}^{8} \in \mathcal{A}_2^{\mathcal{N} \times I \times J} .
\end{aligned}$$

The redundancies are

$$\begin{aligned}
\rho_{\times} &= \frac{\sum_{j=1}^{8} |A_j| \cdot \mu_{\times}(T \otimes H_j) - \mu_{\times}(P)}{\mu_{\times}(P)} = \frac{17 + 7 \cdot 13 - 108}{108} = 0\,\% \\
\rho_{\cdot} &= \frac{\sum_{j=1}^{8} |A_j| \cdot \mu_{\cdot}(T \otimes H_j) - \mu_{\cdot}(P)}{\mu_{\cdot}(P)} = \frac{\frac{17+7\cdot 13}{8} - \frac{108}{8}}{\frac{108}{8}} = 0\,\% .
\end{aligned}$$

The challenge lies in determining how to get $\mathscr{T}$, $\mathscr{H}$, and $\mathscr{A}$ from the given piano roll P. We believe that this task is extremely complex, so we will proceed by assuming we have the texture T and demonstrate how to obtain $\mathscr{H}$ and $\mathscr{A}$.

To achieve this, we once again employ the erosion. However, this time we use the erosion of a piano roll by a texture. It is important to note that since $P \in \mathcal{A}_3^{\mathcal{T}_{\mathbf{c}} \times \mathcal{N}}$ and $T \in \mathcal{A}_3^{\mathbb{Q} \times I}$, there is no direct definition for such an erosion.

For the sake of generality, let us abstract away the specific choices for time and frequency and consider $P \in \mathcal{A}_3^{\mathcal{T} \times \mathcal{F}}$ and $T \in \mathcal{A}_3^{\mathbb{Q} \times I}$. We will employ a similar approach as in the previous section and use the bijections:

$$\iota_3 : \mathcal{A}_3^{\mathcal{T} \times \mathcal{F}} \to \mathcal{A}_3^{\mathcal{T} \times \mathcal{F} \times \mathbf{0}} \qquad \iota_2 : \mathcal{A}_3^{\mathbb{Q} \times I} \to \mathcal{A}_3^{\mathbb{Q} \times \mathbf{0} \times I} \tag{5.15}$$

where we recall that $\mathbf{0} = \{0\}$.

We have now two compatible domains $\mathcal{T} \times \mathcal{F} \times \mathbf{0}$ and $\mathbb{Q} \times \mathbf{0} \times I$. Moreover, we can use the inclusion:

$$\begin{array}{rccc} \iota_{\mathbb{Z}} : & \mathcal{A}_3^{\mathbb{Q} \times I} & \to & \mathcal{A}_3^{\mathbb{Q} \times \mathbb{Z} \times I} \\ & T = (R_i)_{i \in I} & \mapsto & T \otimes \mathbf{0} := (R_i \otimes \mathbf{0})_{i \in I} \end{array} \tag{5.16}$$

and treat any texture as a stack of piano rolls.

The erosion is now well defined and has the explicit expression

$$\begin{array}{rccc} \varepsilon_T : & \mathcal{A}_3^{\mathcal{T} \times \mathcal{F}} & \to & \mathcal{A}_2^{\mathcal{T} \times \mathcal{F} \times I} \\ & P & \mapsto & (P \ominus R_i \otimes \mathbf{0})_{i \in I} \end{array}$$

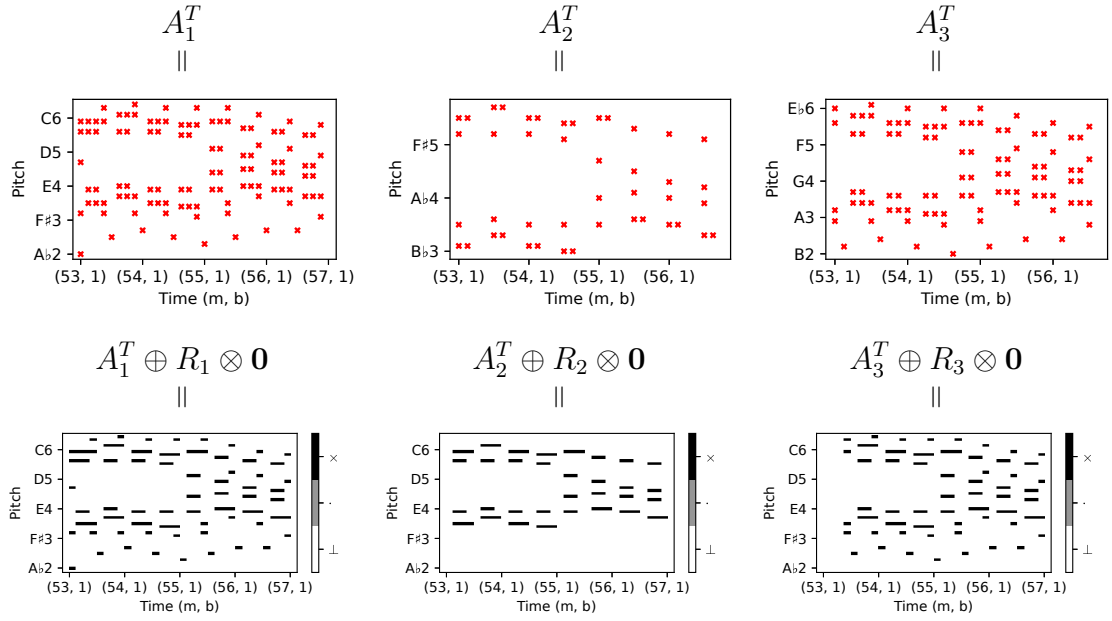
with corresponding adjoint dilation

$$\begin{array}{rccc} \delta_T : & \mathcal{A}_2^{\mathcal{T} \times \mathcal{F} \times I} & \to & \mathcal{A}_3^{\mathcal{T} \times \mathcal{F}} \\ & \mathscr{A} & \mapsto & \bigvee_{i \in I} A_i \oplus R_i \otimes \mathbf{0} \end{array}$$

where $T = (R_i)_{i \in I} \in \mathcal{A}_3^{\mathbb{Q} \times I}$ and $\mathscr{A} = (A_i)_{i \in I}$ with $A_i \in \mathcal{A}_2^{\mathcal{T} \times \mathcal{F}}$, $\forall i \in I$.

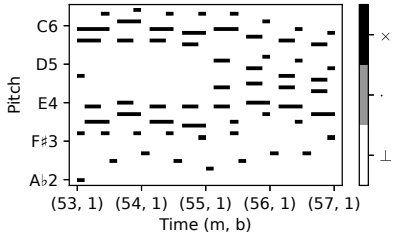
Let us illustrate this technique with our example. By calling $\mathscr{A}^T = (A_i^T)_{i \in I} =$

$P \ominus T$, we have



and

$$\mathscr{A}^T \oplus T \otimes \mathbf{0} = \bigvee_{i \in I} A_i \oplus R_i \otimes \mathbf{0} = \quad = P\,. \qquad (5.17)$$



In this case, we have a huge redundancy[4]. Indeed, using the reference redundan-

[4]The redundancy of T over P is defined as the redundancy of $T \otimes \mathbf{0}$ over P.

cies, we obtain

$$\begin{aligned}\rho_{\times} &= \frac{\sum_{i=1}^{3} |A_i| \cdot \mu_{\times}(R_i \otimes \mathbf{0}) - \mu_{\times}(P)}{\mu_{\times}(P)} = \frac{108 \cdot 1 + 44 \cdot 2 + 95 \cdot 1 - 108}{108} \\ &= \frac{183}{108} \approx 169.4\,\% \\ \rho_{\cdot} &= \frac{\sum_{i=1}^{3} |A_i| \cdot \mu_{\cdot}(R_i \otimes \mathbf{0}) - \mu_{\cdot}(P)}{\mu_{\cdot}(P)} = \frac{108 \cdot \frac{1}{8} + 44 \cdot \frac{1}{8} + 95 \cdot \frac{1}{8} - \frac{108}{8}}{\frac{108}{8}} \\ &= \frac{183}{108} \approx 169.4\,\%\,.\end{aligned}$$

The challenge now is to remove the redundant activations of $\mathscr{A}^T$ such that the redundancy is $0\,\%$.

5.2.1 Extracting a Minimal Set of Activations

We recall that we have a piano roll $P \in \mathcal{A}_3^{\mathcal{T}\times\mathcal{F}}$ and a texture $T \in \mathcal{A}_3^{\mathbb{Q}\times I}$. Let us consider $\mathscr{A} \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{F}\times I}$ such that $P = \mathscr{A} \oplus T \otimes \mathbf{0}$. Then, by using the properties of mathematical morphology, we have that

$$\begin{aligned}\mathscr{A} &\preceq \mathscr{A} \bullet T \otimes \mathbf{0} \\ &= (\mathscr{A} \oplus T \otimes \mathbf{0}) \ominus T \otimes \mathbf{0} \\ &= P \ominus T \otimes \mathbf{0} \\ &:= \mathscr{A}_T^P\end{aligned} \tag{5.18}$$

which means that by taking the erosion $\mathscr{A}_T^P = P \ominus T \otimes \mathbf{0}$ we will always have extra activations. We are focusing now on extracting $\mathscr{A}_{\min} \preceq \mathscr{A}_T^P$ such that it is minimal, i.e.,

$$\forall \mathscr{A} \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{F}\times I} : P = \mathscr{A} \oplus T \otimes \mathbf{0},\ \mathscr{A}_{\min} \preceq \mathscr{A}_T^P\,. \tag{5.19}$$

Since the amplitude range of $\mathscr{A}_T^P$ is the Boolean lattice $\mathcal{A}_2$, we can use the isomorphism given by the support function[5]

$$\begin{array}{rccc}\text{supp}: & \mathcal{A}_2^{\mathcal{T}\times\mathcal{F}\times I} & \to & \mathcal{P}(\mathcal{T}\times\mathcal{F}\times I) \\ & \mathscr{A} & \mapsto & \text{supp}(\mathscr{A})\end{array}$$

and identify $\mathscr{A}_T^P$ with its support. From now, we make the abuse of notation $\mathscr{A}_T^P \subseteq \mathcal{T}\times\mathcal{F}\times I$. We use then $\subseteq$ instead of $\preceq$ for the order.

[5]The inverse of the characteristic function.

Let us now transform our problem into a problem on sets: we call

$$\mathfrak{A}_T^P = \{\mathscr{A} \subseteq \mathcal{T} \times \mathcal{F} \times I : P = \mathscr{A} \oplus T \otimes \mathbf{0}\} \tag{5.20}$$

and we want to find the $\mathscr{A}_{\min} \in \mathfrak{A}_T^P$ that are minimal in the sense of $\subseteq$.

Let us give a brief overview of how such a task can be achieved. Let $X = \{x_1, x_2, ..., x_n\}$ be a set with $|X| = n \in \mathbb{N}^*$. We call $X_{i_1,i_2,...,i_k} = X \setminus \{x_{i_1}, x_{i_2}, ..., x_{i_k}\}$. The lattice $(\mathcal{P}(X), \subseteq)$ of subsets of X with $|X| = 3$ is shown in Figure 5.3 with the arrows meaning inclusion.

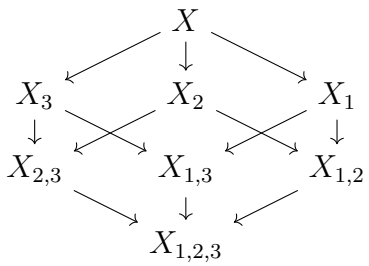


Figure 5.3: Graph of inclusion for subsets of X with $|X| = 3$.

If we set $X = \mathscr{A}_T^P$ and remove the elements that do not belong to $\mathfrak{A}_T^P$, we can identify a minimal element as one that has no outgoing edges. However, to exhaustively check all the edges and explore the entire graph[6], we would need to perform $\left|\mathscr{A}_T^P\right|!$ checks. This quickly becomes infeasible; for instance, in the excerpt of Figure 5.2 of 4 measures, we have $\left|\mathscr{A}_T^P\right| = 247$, $247! \approx 10^{484}$, which is astronomically large.

Nonetheless, there is a theoretical insight that can aid us. Since we are interested in exploring the elements of $\mathfrak{A}_T^P$, which are those $\mathscr{A}$ satisfying $\mathscr{A} \oplus T = P$, the following property holds: if $\mathscr{A}_1 \subseteq \mathscr{A}_2$, then $\mathscr{A}_1 \oplus T \subseteq \mathscr{A}_2 \oplus T$. Therefore, if we find a $\mathscr{A}$ such that $\mathscr{A} \oplus T \not\subseteq P$, there is no need to explore any subset $\mathscr{A}' \subseteq \mathscr{A}$.

This insight helps to reduce the number of elements to check. Nevertheless, even with this optimization, the worst-case complexity remains $\mathcal{O}(n!)$ with $n = \left|\mathscr{A}_T^P\right|$.

However, we know that there are $2^{\left|\mathscr{A}_T^P\right|}$ subsets of $\mathscr{A}_T^P$ and then we can check if they belong to $\mathfrak{A}_T^P$ one by one. This approach leads to the graph depicted in Figure 5.4 and a time complexity of $\mathcal{O}(2^n)$. In order to exploit the property mentioned before, we avoid the exploration of deeper nodes once it is determined that they do not belong to $\mathfrak{A}_T^P$. This pruning strategy significantly reduces the number of paths explored.

[6]All the graphs used in this work are in fact digraphs (directed graphs).

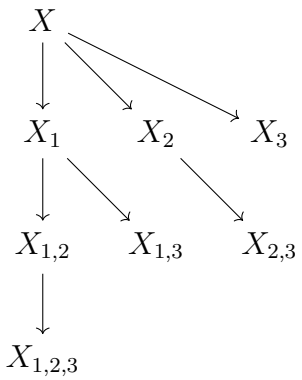


Figure 5.4: Graph of the subsets of X with $2^{|X|} - 1$ arrows, where $|X| = 3$.

While this approach is better in terms of time complexity, it has a drawback: we no longer know if a node is minimal with this graph since there are missing edges in terms of inclusion; for instance, if we attain $X_{1,2}$ as a minimal node and later we see that X_2 works but not $X_{2,3}$, we may be tempted to say that $X_{2,3}$ is minimal (since there is no outgoing edges) but this is not true since X_2 is already an option.

To determine if a node without outgoing edges in this graph is truly minimal, it is necessary to compare it against all other minimal subsets found earlier. While this is possible, we still have a time complexity of $\mathcal{O}(2^n)$, which is intractable: in our example with $\left|\mathscr{A}_T^P\right| = 247$, the number of subsets to consider is $2^{247} \approx 10^{74}$, which is far beyond the capabilities of current computing resources.

5.2.2 Linear Approach

In this section, we present an alternative approach to tackling this problem. This approach leverages the inherent topology of our space and converts the problem into a shortest path problem within a directed acyclic graph (DAG), ultimately transforming it into a linear problem.

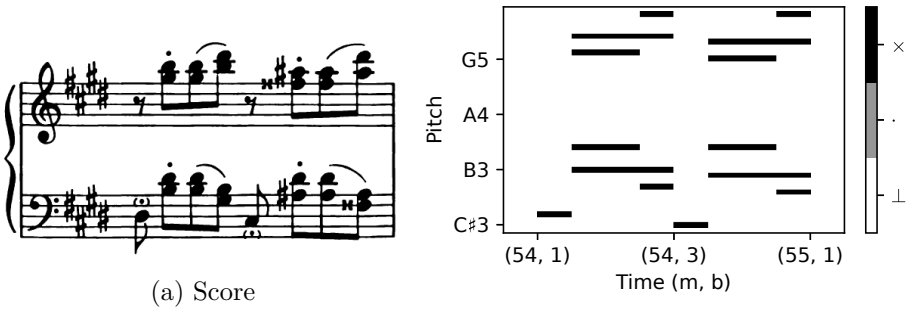
As previously discussed, the challenge involves exploring all possible combinations of elements in $\mathscr{A}_T^P$ to identify the minimal subsets. While this is initially an exponential problem, we can capitalize on the structure of the set $\mathscr{A}_T^P$ to reformulate it into a linear problem in the size of the graph.

The central insight guiding our strategy is as follows: since each rhythm R_i has a support contained inside an interval $[s_i, e_i] \subseteq \mathbb{Q}$, any element in the piano roll denoted as $(t, \xi, P(t, \xi)) \in \mathcal{T} \times \mathcal{F} \times \mathcal{A}$ can be attributed to an activation within $[t - e_i, t - s_i] \times \{\xi\} \times \{i\} \subseteq \mathcal{T} \times \mathcal{F} \times I$. In essence, covering an element $(t, \xi, P(t, \xi))$ necessitates

selecting an element from a restricted set $\mathscr{A}_{(t,\xi)} \subset \mathscr{A}_T^P$ capable of producing the value $P(t,\xi)$.

We use this principle to construct a graph wherein each element of the piano roll has the activations capable of covering it. Subsequently, we traverse paths within this graph, ensuring the inclusion of at least one activation for each element of the piano roll.

Let us illustrate the process of creating the graph using a fragment from our example, as depicted in Figure 5.5. We only consider the measure 54 of Beethoven's Piano Sonata No. 14, Op. 27 No.2.

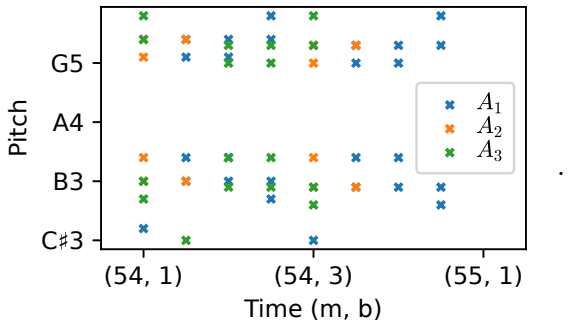


(a) Score

(b) Piano roll

Figure 5.5: Third movement of Beethoven's Piano Sonata No.14, Op.27 No.2 measure 54.

The resulting activations $\mathscr{A}_T^P = (A_i)_{i\in I}$, $I = \{1,2,3\}$ are



.

To construct the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that represents the potential activations covering each element of P, we begin by determining the vertices $\mathcal{V}$.

Each vertex will be an element of $\mathcal{T} \times \mathcal{T} \times \mathcal{F} \times I$ denoted by $(t_P, t_{\mathscr{A}}, \xi, i)$ where

- $t_P \in \mathcal{T}$ is the time coordinate of the element of P that is covered,
- $t_{\mathscr{A}} \in \mathcal{T}$ is the time coordinate of the activation that covers it,
- $\xi \in \mathcal{F}$ is the frequency coordinate of both the element of P and the activation[7],
- $i \in I$ is the index of the activation.

The resulting vertices of the graph are depicted in Figure 5.6. The coordinates (t_P, ξ) are used to create the axes that map the elements of the piano roll. Each element belonging to a cell is of the form $(t_{\mathscr{A}}, i)$, indicating the timestamp of the activation and the index. Please note that in the diagram, pitch notations are used based on the default system, rather than the ones present in the actual score (for instance, we use E♭ instead of D♯ and G instead of F𝄪).

Having defined the vertices of the graph, our next task is to determine the edges. As a reminder, our goal is to find a path that passes through at least one element of each cell of the grid defined by (t_P, ξ), skipping those where there are no elements. This ensures that each element of the score is covered by at least one activation. To model this, we need to select edges that connect elements from each cell of the grid. Furthermore, we will arrange these edges in a topological order, creating a directed acyclic graph that allows us to find a linear solution to the shortest path problem.

Various edge choices are possible, but we opt for the following approach:

1. We arrange the non-empty different cells in the lexicographical order going first through the frequency axis and then through the time axis.
2. For two consecutive cells, we connect every node in the first cell with a node in the second cell.

The result of this process leads to a directed acyclic graph. Moreover, the graph is a *chain of bipartite graphs*. This concept will be exposed in Section 5.2.4. The resulting graph is depicted in Figure 5.7. Since the number of edges is big and makes the graph difficult to understand, we zoom to the vertices that are on the cells $\big((3,0), \text{C}\sharp 3\big)$ and $\big((3, \frac{1}{8}), \text{B}\flat 3\big)$ in Figure 5.7b.

Now, each path in this graph that goes from a node at the bottom left to a node at the top right represents a sequence of nodes that corresponds to the activations selected from $\mathscr{A}$ to cover P.

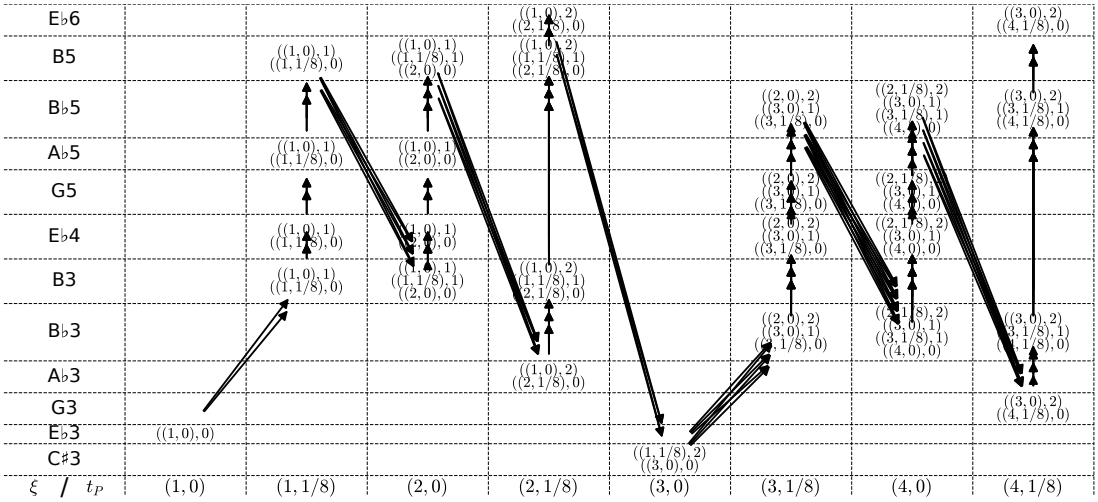
The challenge at this point is to devise a method for determining the optimal path. The objective is to minimize the number of distinct activations used, and this

[7]The frequency coordinate is shared by the covered element and the activation because of the nature of the dilation by a texture.

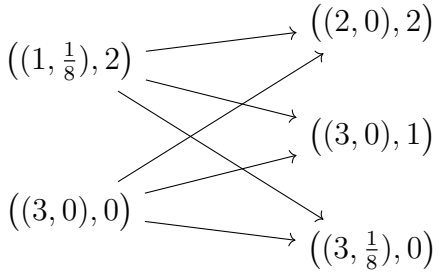
E♭6				$((1,0),2)$ $((2,1/8),0)$				$((3,0),2)$ $((4,1/8),0)$
B5		$((1,0),1)$ $((1,1/8),0)$	$((1,0),1)$ $((1,1/8),1)$ $((2,0),0)$	$((1,0),2)$ $((1,1/8),1)$ $((2,1/8),0)$				
B♭5						$((2,0),2)$ $((3,0),1)$ $((3,1/8),0)$	$((2,1/8),2)$ $((3,0),1)$ $((3,1/8),1)$ $((4,0),0)$	$((3,0),2)$ $((3,1/8),1)$ $((4,1/8),0)$
A♭5		$((1,0),1)$ $((1,1/8),0)$	$((1,0),1)$ $((2,0),0)$					
G5						$((2,0),2)$ $((3,0),1)$ $((3,1/8),0)$	$((2,1/8),2)$ $((3,0),1)$ $((4,0),0)$	
E♭4		$((1,0),1)$ $((1,1/8),0)$	$((1,0),1)$ $((2,0),0)$			$((2,0),2)$ $((3,0),1)$ $((3,1/8),0)$	$((2,1/8),2)$ $((3,0),1)$ $((4,0),0)$	
B3		$((1,0),1)$ $((1,1/8),0)$	$((1,0),1)$ $((1,1/8),1)$ $((2,0),0)$	$((1,0),2)$ $((1,1/8),1)$ $((2,1/8),0)$				
B♭3						$((2,0),2)$ $((3,0),1)$ $((3,1/8),0)$	$((2,1/8),2)$ $((3,0),1)$ $((3,1/8),1)$ $((4,0),0)$	$((3,0),2)$ $((3,1/8),1)$ $((4,1/8),0)$
A♭3				$((1,0),2)$ $((2,1/8),0)$				
G3								$((3,0),2)$ $((4,1/8),0)$
E♭3	$((1,0),0)$							
C♯3					$((1,1/8),2)$ $((3,0),0)$			
ξ / t_P	$(1,0)$	$(1,1/8)$	$(2,0)$	$(2,1/8)$	$(3,0)$	$(3,1/8)$	$(4,0)$	$(4,1/8)$

Figure 5.6: Vertices of the graph of activations for the piano roll given in Figure 5.5.

consideration relies not only on the edges between two individual nodes, but also on the nodes selected along the path. Addressing this issue has led to the use of the derived graph.



(a) Full graph.



(b) Zoom on the vertices between the cells $\big((3,0), \mathrm{C\sharp 3}\big)$ and $\big((3, \frac{1}{8}), \mathrm{B\flat 3}\big)$

Figure 5.7: Graph of activations for the piano roll given in Figure 5.5.

5.2.3 The Derived Graph of a Graph

The notion of derived graph of a graph was introduced by Harary and Norman, 1960 under the name of *line graph*, and it is known by various other names such as *interchange graph*, *adjoint*, or *edge-to-vertex dual* (Beineke, 1970). The idea behind this construction appeared earlier in (Whitney, 1932; Krausz, 1943), and it has been extensively studied by researchers like Beineke, 1968, 1970; Beineke and Zamfirescu, 1982; Beineke and Bagga, 2021, from who we borrow the term *derived graph* and the notation $\partial\mathcal{G}$, used for instance by Beineke, 1968, 1970.

The definition of the derived graph is as follows.

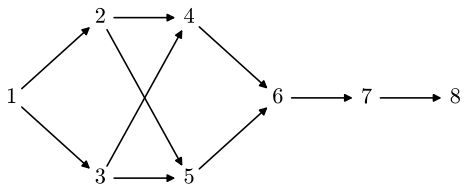
Definition 5.4 (Derived graph)**.** Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph where $\mathcal{E} \subseteq \mathcal{V}^2$. Then, the **derived graph** of $\mathcal{G}$ is the graph $\partial\mathcal{G} = (\mathcal{E}, \mathcal{E}')$ where $e' \in \mathcal{E}' \subseteq \mathcal{E}^2 \Leftrightarrow e' = \big((v_1, v_2), (v_2, v_3)\big)$ such that $(v_1, v_2) \in \mathcal{E}$ and $(v_2, v_3) \in \mathcal{E}$.

Since the vertex v_2 is shared, we will rather write $e' \in \mathcal{E}' \subseteq \mathcal{V}^3$, $e' = (v_1, v_2, v_3)$.

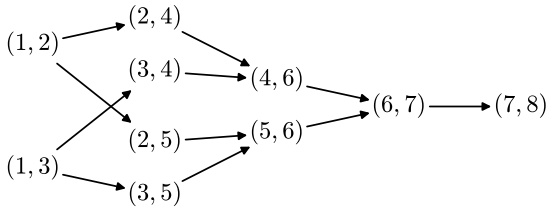
Notice that we can derive a graph as much as we want, and we notate $\partial^k\mathcal{G}$ the k^{th} derived graph of $\mathcal{G}$. The k^{th} derived graph of $\mathcal{G}$ can be seen as the graph whose vertices are the paths of length $k+1$ of $\mathcal{G}$ and the edges represent that two paths share common elements and thus can form a bigger path.

To illustrate this concept, let us consider the following example.

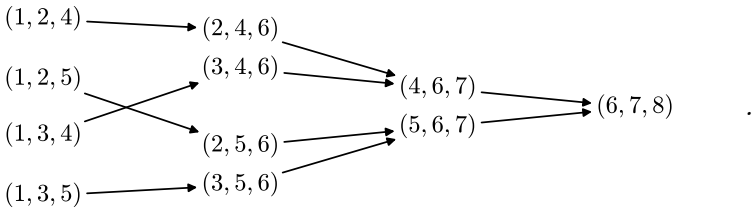
Example 5.5. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the graph represented by*



Then, the derived graph of $\mathcal{G}$, $\partial\mathcal{G}$ is represented by



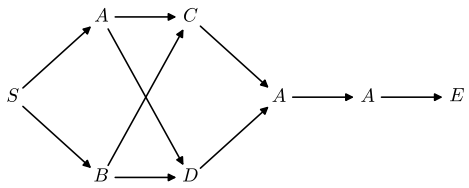
The 2^{nd} derived graph of $\mathcal{G}$, $\partial^2\mathcal{G}$ is represented by



The derived graph becomes useful when we want to consider second-order information in a graph. To further explain this concept, let us consider a simple problem

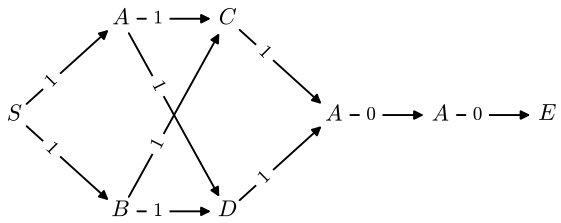
as a toy example, which will provide insights into solving the larger problem of finding minimal activations.

Imagine assigning a letter to each node in the graph presented in the previous example. The graph is illustrated as follows:



.

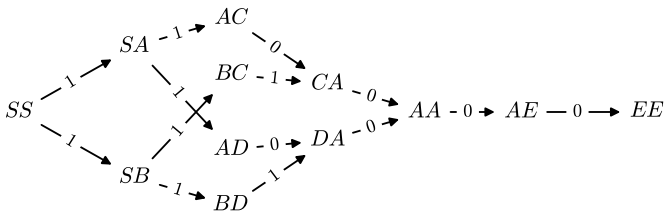
Now, the goal is to find a path from node S (start) to node E (end) that uses the minimum number of distinct letters. In other words, we aim to minimize the number of different letters encountered along the path (excluding the start and end nodes). We can assign a weight of 1 to an edge if the letters on the connected nodes are different, and a weight of 0 if they are the same. This gives us the weighted graph:



.

If we choose the path $S - B - D - A - A - E$ and calculate its weight, we get 3, which represents the number of different letters (A, B, and D) encountered along the path. However, if we choose the path $S - A - C - A - A - E$, we still get a weight of 3, even though there are only 2 distinct letters (A and C) in this case.

This issue can be addressed using the concept of the derived graph. By differentiating the graph once, we obtain the following modified graph (with artificial start and end nodes denoted as SS and EE):



.

In this new graph, if we follow the path $SS - SA - AC - CA - AA - AE - EE$ (which can be simplified to $S - A - C - A - A - E$), we obtain a weight of 2, which accurately represents the number of distinct letters.

By using this concept, we can iteratively differentiate the graph to consider common letters that are spaced more than two edges apart. The number of times we need to differentiate is equal to the distance (in terms of edges) between the common letters minus one.

5.2.4 Chain of Bipartite Graphs

In this chapter, we use a particular type of graph: *chain of bipartite graphs*. This concept appears in (Sathiamoorthy, 2020) and can be understood as a chain in the sense of (Concas et al., 2021) of bipartite graphs. We present the definition next.

Definition 5.6 (Chain of bipartite graphs)**.** A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a **chain of bipartite graphs** of size $L \in \mathbb{N}^*$ if the set of vertices $\mathcal{V}$ can be subdivided into L disjoint non-empty subsets

$$\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2 \cup ... \cup \mathcal{V}_L \tag{5.21}$$

such that $\forall l \in \{1, 2, ..., L-1\}$, all the vertices of $\mathcal{V}_l$ are connected with all the vertices in $\mathcal{V}_{l+1}$, and there are no other edges.s

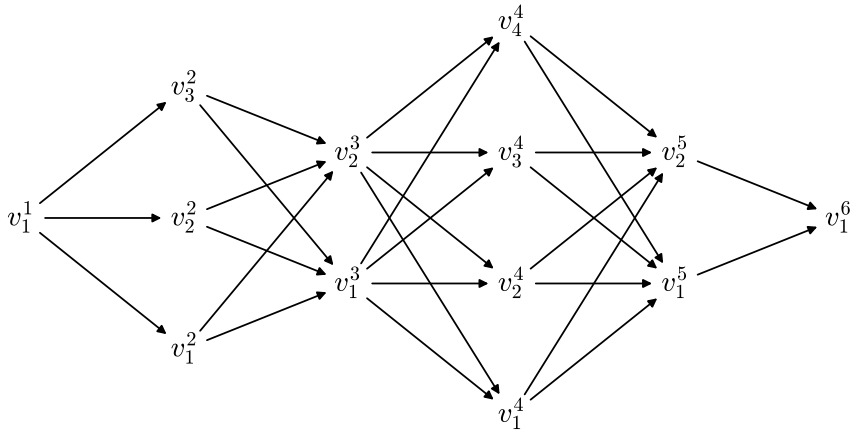
Formally, for digraphs,

$$\begin{aligned}
&\mathcal{V} = \bigcup_{l=1}^{L} \mathcal{V}_l, \quad l \neq l' \Rightarrow \mathcal{V}_l \cap \mathcal{V}_{l'} = \emptyset \text{ and } \forall l \in \{1, 2, ..., L\},\ \mathcal{V}_l \neq \emptyset \\
&\mathcal{E} \subseteq \{(u, v) \in \mathcal{V}^2 : \exists l \in \{1, 2, ..., L-1\}, u \in \mathcal{V}_l, v \in \mathcal{V}_{l+1}\}\,.
\end{aligned}$$

If $\mathcal{E} = \{(u, v) \in \mathcal{V}^2 : \exists l \in \{1, 2, ..., L-1\}, u \in \mathcal{V}_l, v \in \mathcal{V}_{l+1}\}$, we say that the digraph is a **chain of complete bipartite graphs**.

We use the notation $G_{n_1 \to n_2 \to ... \to n_L}$ for a chain of bipartite graphs and $\mathcal{K}_{n_1 \to n_2 \to ... \to n_L}$ for a chain of complete bipartite graphs, where $n_l = |\mathcal{V}_l|$, $l \in \{1, 2, ..., K\}$.

Example 5.7. *The chain of complete bipartite graphs $\mathcal{K}_{1 \to 3 \to 2 \to 4 \to 2 \to 1}$ is represented by*



We denote the number of vertices of a graph $\mathcal{G}$ by $|\mathcal{G}|_{\mathcal{V}}$ and the number of edges by $|\mathcal{G}|_{\mathcal{E}}$. The k^{th} derived graph of chains of complete bipartite graphs have the following properties.

Proposition 5.8. *Let $\mathcal{K}_{n_1 \to n_2 \to ... \to n_L}$ be a chain of complete bipartite graphs. Then, $\forall k \in \{1, 2, ..., L-1\}$,*

1. *$\partial^k \mathcal{K}_{n_1 \to n_2 \to ... \to n_L}$ is a chain of bipartite graphs of size $L - k$,*
2. *$\left|\partial^k \mathcal{K}_{n_1 \to n_2 \to ... \to n_L}\right|_{\mathcal{V}} = \sum_{l=k+1}^{L} \prod_{m=0}^{k} n_{l-m}$,*
3. *$\left|\partial^k \mathcal{K}_{n_1 \to n_2 \to ... \to n_L}\right|_{\mathcal{E}} = \sum_{l=k+2}^{L} \prod_{m=0}^{k+1} n_{l-m}$.*

Proof. Let us prove the first statement by induction over k.

$\partial^0 \mathcal{K}_{n_1 \to n_2 \to ... \to n_L} = \mathcal{K}$ is indeed a chain of bipartite graphs of length $L - 0 = L$. Now, if $\partial^k \mathcal{K}_{n_1 \to n_2 \to ... \to n_L}$ is a chain of bipartite graphs of size $L - k$, then we have $\mathcal{V}_{k+1}^k, \mathcal{V}_{k+2}^k, ..., \mathcal{V}_L^k$ disjoint non-empty sets such that the vertices in $\mathcal{V}_l$ are connected only to those on $\mathcal{V}_{l+1}$. Thus, the edges can be split into $\mathcal{E}_{k+1 \to k+2}, \mathcal{E}_{k+2 \to k+3}, ..., \mathcal{E}_{L-1 \to L}$, where $\mathcal{E}_{l \to l+1}$ is the set of edges that connect elements of $\mathcal{V}_l$ with elements of $\mathcal{V}_{l+1}$. They are then disjoint and an element of $\mathcal{E}_{l \to l+1}$ is connected to one in $\mathcal{E}_{l' \to l'+1}$ in $\partial^{k+1} \mathcal{K}_{n_1 \to n_2 \to ... \to n_L}$ only if $l + 1 = l'$.

It remains to be seen that the $\mathcal{E}_{l \to l+1}$ are non empty. Their cardinal is equal to the cardinal of $\mathcal{V}_l^k$ times n_{l+1}, so let us compute the cardinal of $\mathcal{V}_l^k$. We know that the elements of $\mathcal{V}_l^k$ are paths of length $k + 1$. Since the graph is complete, we know that there are $\prod_{m=0}^{k} n_{l-m}$ paths, which implies that $|\mathcal{V}_l^k| = \prod_{m=0}^{k} n_{l-m}$ and $|\mathcal{E}_{l \to l+1}| = \left(\prod_{m=0}^{k} n_{l-m}\right) n_{l+1} = \prod_{m=0}^{k+1} n_{l+1-m} \neq 0$.

Finally, let us prove the formulas 2 and 3:

$$\left|\partial^k \mathcal{K}_{n_1\to n_2\to\ldots\to n_L}\right|_{\mathcal{V}} = \sum_{l=k+1}^{L} |\mathcal{V}_l^k| = \sum_{l=k+1}^{L}\prod_{m=0}^{k} n_{l-m}$$

$$\begin{aligned}\left|\partial^k \mathcal{K}_{n_1\to n_2\to\ldots\to n_L}\right|_{\mathcal{E}} &= \sum_{l=k+1}^{L-1} |\mathcal{V}_l^k| n_{l+1} = \sum_{l=k+1}^{L-1}\left(\prod_{m=0}^{k} n_{l-m}\right) n_{l+1} \\ &= \sum_{l=k+1}^{L-1}\prod_{m=0}^{k+1} n_{l+1-m} = \sum_{l=k+2}^{L}\prod_{m=0}^{k+1} n_{l-m}\,.\end{aligned}$$

□

5.2.5 Modeling the Problem as a Shortest Path Problem

We have currently all the tools that we need to solve our initial problem (to find a minimal set of activations). Indeed, the graph presented in Figure 5.7 is a chain of complete bipartite graphs of length 26: each cell is a set of vertices $\mathcal{V}_l$ and they are ordered in the lexicographical order starting by the frequency. We have then a graph isomorphic to

$$\mathcal{K}_{1\to 2\to 2\to 2\to 2\to 3\to 2\to 2\to 3\to 2\to 3\to 3\to 2\to 2\to 3\to 3\to 3\to 3\to 4\to 3\to 3\to 4\to 2\to 3\to 3\to 2}\,.$$

We want now to find a path that starts from one of the first nodes to one of the end nodes. To model that as a shortest path problem, we add an artificial node at the beginning that is connected to all the first nodes and one that is connected to all of the end nodes (thus having a complete bipartite graph of length 28 with $n_1 = n_{28} = 1$).

We want now to assign weights to this graph. We saw in Section 5.2.3 that we can know information from $k \in \mathbb{N}^*$ vertices before if we use the $k-1^{\text{th}}$ derived graph. In our case, we want to know if an activation $(t_{\mathscr{A}}, \xi, i)$ is present already in the path, thus assigning 0 to the weight of the edge. To do that, we need to compute the maximum number of cells that can separate two activations. We call it

$$K_T = \max\{\frac{e_i - s_i}{\Delta_P} : i \in I\} \tag{5.22}$$

where $T = (R_i)_{i\in I}$ and $\text{supp}(R_i) \subseteq [s_i, e_i] \subseteq \mathbb{Q}$ and Δ_P is the time resolution of the piano roll to perform the computations[8].

[8]Note that K_T depends on computational parameters.

In our case, if we choose as Δ_P the tatum, i.e., ♪, we have

$$\begin{aligned} K_T &= \max\{\frac{\frac{1}{8}-0}{\frac{1}{8}}, \frac{\frac{3}{8}-\frac{1}{8}}{\frac{1}{8}}, \frac{\frac{4}{8}-\frac{3}{8}}{\frac{1}{8}}\} \\ &= \max\{1,2,1\} = 2 \end{aligned}$$

Instead of the lexicographical order that starts by frequency we can use the one that starts by time and solve the problem with a derived graph of first order. The problem is that this approach, where we only take into account the number of different activations, is that it gives a huge quantity of shortest paths. We would like to refine this approach to find a better path.

5.2.6 The Sparsity of Time Activations

In order to improve the quality of our results we introduce the notion of *sparsity of time activations*. The idea behind this is that we want to have the textures activated in the minimum possible times. Moreover, we require a property for the chosen activations: when an activation is chosen and has timestamp $t_{\mathscr{A}}$ and index i, we need to choose activations with the same timestamp and with the remaining indices in $I \setminus \{i\}$. This is related to the concept that a texture should be complete once it is activated.

This allows us to eliminate several activations that cannot be chosen. Indeed, if we consider the contraction

$$\lambda(t) = \bigwedge_{i\in I} \bigvee_{\xi\in\mathcal{F}} \mathscr{A}(t,\xi,i) = \bigwedge_{i\in I} \bigvee_{\xi\in\mathcal{F}} A_i(t,\xi)$$

we have the timestamps where this property can hold.

Then, we multiply[9] $\mathscr{A}$ by λ and we get the filtered activations that might have this property.

However, even after filtering $\mathscr{A}$ with λ, we need to ensure that the property aforementioned holds. To do that, we need to go through every element in frequency. This is why we presented the graph in the lexicographical order of frequency first. We have then that there are at most

$$K_{\mathcal{F}} = \max\{|\operatorname{supp}(P|_t)| : t \in \mathcal{T}\} \tag{5.23}$$

[9]The multiplication is performed between an element of $\mathcal{A}_2^{\mathcal{T}\times\mathcal{F}\times I}$ and an element of $\mathcal{A}_2^{\mathcal{T}}$, which means that it should be understood as $\mathscr{A}\cdot\lambda : \mathcal{T}\times\mathcal{F}\times I \to \mathcal{A}_2$, $(t,\xi,i)\mapsto \mathscr{A}(t,\xi,i)\cdot\lambda(t)$.

elements at the same time, which gives us the order of derivation

$$K = K_T \cdot K_{\mathcal{F}} - 1\,. \tag{5.24}$$

In our example, where $k_{\mathcal{F}} = 4$, we have $k = 4 \cdot 4 - 1 = 15$, which is a huge order of derivation. Indeed, if we call

$$\mathcal{G} = \mathcal{K}_{1\to2\to2\to2\to2\to3\to2\to2\to3\to2\to3\to3\to2\to2\to3\to3\to3\to3\to4\to3\to3\to4\to2\to3\to3\to2}$$

our graph and if we use the formulas from Proposition 5.8, we have

$$|\mathcal{G}|_{\mathcal{V}} = \sum_{l=1}^{26} n_l = 67 \qquad |\partial^{15}\mathcal{G}|_{\mathcal{V}} = \sum_{l=16}^{26}\prod_{m=0}^{15} n_{l-m} = 84\,633\,984$$
$$|\mathcal{G}|_{\mathcal{E}} = \sum_{l=2}^{26} n_l n_{l-1} = 174 \qquad |\partial^{15}\mathcal{G}|_{\mathcal{E}} = \sum_{l=17}^{26}\prod_{m=0}^{16} n_{l-m} = 192\,595\,968\,.$$

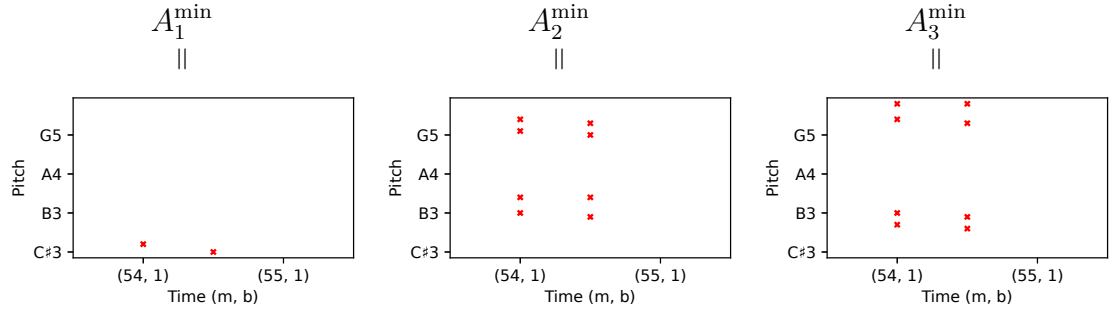
These sizes are too big to perform computations (specially because the result is only a single measure). When we filter the activations with the contraction we obtain the graph

$$\mathcal{G} = \mathcal{K}_{1\to2\to2\to2\to2\to2\to1\to1\to2\to1\to2\to2\to1\to2\to1\to1\to1\to1\to1\to1\to1\to1\to1\to1\to1\to1}\,.$$

Then, the sizes of the graph and its 15^{th} derivative are

$$|\mathcal{G}|_{\mathcal{V}} = 35 \qquad |\partial^{15}\mathcal{G}|_{\mathcal{V}} = \sum_{l=16}^{26}\prod_{m=0}^{15} n_{l-m} = 1568$$
$$|\mathcal{G}|_{\mathcal{E}} = 48 \qquad |\partial^{15}\mathcal{G}|_{\mathcal{E}} = \sum_{l=17}^{26}\prod_{m=0}^{16} n_{l-m} = 1560\,.$$

which is now tractable. Indeed, we obtain as minimal activations



We finish by giving some bounds of our problem: while the problem is now linear in N (the number of elements of P), it has a polynomial component due to the derivation. Indeed, if we call our graph $\mathcal{G}$ and we set $n_{\max} = \max\{n_l : l \in \{1, 2, ..., L\}$, we have the bounds

$$|\mathcal{G}|_{\mathcal{V}} \leq (N+1)n_{\max}, \qquad |\mathcal{G}|_{\mathcal{E}} \leq (N+1)n_{\max}^2,$$
$$|\partial^K \mathcal{G}|_{\mathcal{V}} \leq (N+1-K)n_{\max}^{K+1}, \qquad |\partial^K \mathcal{G}|_{\mathcal{E}} \leq (N-K)n_{\max}^{K+2}.$$

5.2.7 Conclusions

While the utilization of the derived graph proves to be a powerful theoretical tool for transforming the problem into a linear one, it comes at the price of a polynomial bound in the size of texture and number of simultaneous elements in the score. This limitation presents a significant drawback that often renders practical computations unfeasible.

We consider the challenge of finding the minimal activations as a highly intricate one, closely tied to the inherent complexity of music and its extensive combinatorial nature.

Nonetheless, we believe that this approach offers a profound method for music analysis, as it delves into the essence of each note. Future research might attempt to address this problem using machine learning techniques. While the abstract nature of the problem presents significant difficulties, there could exist various heuristics (such as identifying repeated patterns) that facilitate pruning possibilities in rhythmically complex scenarios.

In particular, an approach centered around genetic algorithms, with evolving agents attempting to cover the score, could enhance efficiency and explanatory power. This methodology could mirror the creative process in musical composition, where motifs are often modified and used as foundational material. Exploring such an approach will be a key direction for our future research in this field, although it extends beyond the scope of this thesis.

5.3 Analyzing Piano Rolls with Harmonies

In the preceding sections, we analyzed piano rolls using both harmonic textures and textures. In this section, we extend our analysis of piano rolls incorporating a frequency object: harmonies. The principal innovation of this section is the development of what we refer to as the *tonal graph*, a construct that enables us to reframe the harmonic analysis challenge as a shortest path problem.

To initiate our exploration, let us delve deeper into the harmonic analysis problem itself.

5.3.1 Harmonic Analysis

The harmonic analysis problem involves providing a harmonic interpretation of a musical score, or a fragment of it, by using harmonic features such as chords, scales, or Roman numerals. Our focus lies in the task of *Roman numeral analysis*, which entails assigning a label expressed as a Roman numeral, along with potential additional indications, to a each fragment of the score. These indications, coupled with the knowledge of the tonic, allow us to deduce details about the chord, its inversion, and any sevenths or added notes (Tymoczko et al., 2019).

There exist diverse approaches to conduct Roman numeral analysis. In this work, we confine our focus to identifying the tonic, the Roman numeral label, and any added notes. We abstain from specifying the inversion; however, this aspect can be addressed using a straightforward dictionary-based lookup approach. Figure 5.8 provides an illustration of a Roman numeral analysis of the previously provided excerpt.

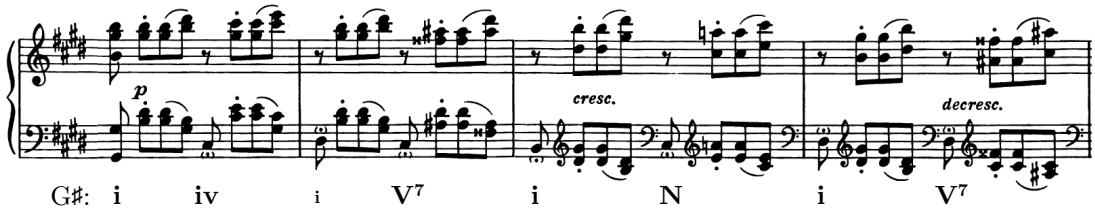


Figure 5.8: Roman numeral analysis of the third movement of Beethoven's Piano Sonata No.14, Op.27 No.2 mm. 53-56.

Roman numeral analysis was initially introduced by Weber, 1832, building upon the groundwork laid by Rameau, 1722. While other techniques, such as Riemann's functional analysis (Riemann, 1893), have garnered the interest of music analysts (Agmon, 1995; Illescas et al., 2007; De Haas et al., 2013), they maintain a close relationship with Roman numeral analysis. Nevertheless, as highlighted by Tymoczko et al., 2019, the translation between Roman numerals and tonal functions is not symmetrical, with the former generally encapsulating more information than the latter. Hence, we opt for Roman numeral analysis due to its richness in explanatory capabilities.

Automatic Roman numeral analysis begins with the contributions of Winograd, 1968 and Maxwell, 1992, who proposed rule-based algorithms. Subsequently, Tem-

perley introduced another rule-based algorithm (Temperley, 1997). Further progress in this direction encompasses works such as (Pardo & Birmingham, 2002; Temperley, 2002, 2004; Illescas et al., 2007; Temperley, 2009; De Haas et al., 2013).

Additionally, a significant advancement within the harmonic analysis domain is the construction of datasets. Although a substantial dataset emerged in 2002 (Goto et al., 2002), it was not until 2015 that we witnessed a proliferation of Roman numeral analysis datasets for classical music (Devaney et al., 2015; López, 2017; Chen & Su, 2018; Neuwirth et al., 2018; Tymoczko et al., 2019), with Micchi et al., 2020 consolidating many of these datasets into a meta-corpus. This influx of datasets has paved the way for the development of machine learning techniques; some of these datasets include neural network architectures for automatic analysis (Chen & Su, 2018; Micchi et al., 2020; López et al., 2021).

5.3.2 The Tonal Graph

In this work, we introduce an alternative approach distinct from rule-based and machine learning methods: a novel model for Roman numeral analysis, framed as a shortest path problem within a chain of bipartite graphs.

To construct this graph, we leverage the outcomes of the preceding section, comprising a collection of activations denoted as $\mathscr{A} \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{N}\times I}$. However, we contract these activations across the dimension I to yield $A = \bigvee_{i\in I} \mathscr{A} = \bigvee_{i\in I} A_i \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{N}}$. This aggregated representation is depicted in Figure 5.9.

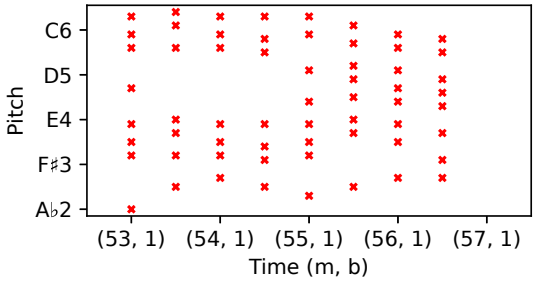


Figure 5.9: Input activations $A \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{N}}$ representing the excerpt from Figure 5.2.

Furthermore, the harmonic attributes we aim to identify are independent of the pitch octave. Thus, we use activations up to the octave, considering them as a chroma roll. This leads to the representation depicted in Figure 5.10.

In order to build the tonal graph, we use as input the activations and a harmony.

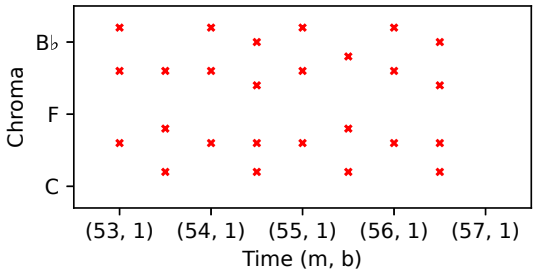


Figure 5.10: Input activations $A \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{N}_{12}}$ up to the octave representing the excerpt from Figure 5.2.

The harmony represents the different Roman numerals we allow in our analysis. For instance, a common harmony that we can use is the one presented in Table 5.1.

RN	Chord	RN	Chord	RN	Chord
I	$\{\overline{0}, \overline{4}, \overline{7}\}$	**IV**	$\{\overline{0}, \overline{5}, \overline{9}\}$	**vi**	$\{\overline{0}, \overline{4}, \overline{9}\}$
i	$\{\overline{0}, \overline{3}, \overline{7}\}$	**iv**	$\{\overline{0}, \overline{5}, \overline{8}\}$	**VI**	$\{\overline{0}, \overline{3}, \overline{8}\}$
ii	$\{\overline{2}, \overline{5}, \overline{9}\}$	**V**	$\{\overline{2}, \overline{7}, \overline{11}\}$	**vii°**	$\{\overline{2}, \overline{5}, \overline{11}\}$
ii°	$\{\overline{2}, \overline{5}, \overline{8}\}$	$\mathbf{V^{45}}$	$\{\overline{0}, \overline{2}, \overline{7}\}$	**N**	$\{\overline{1}, \overline{5}, \overline{8}\}$

Table 5.1: Harmony that we choose for analyzing common tonal music.

The final aspect we need to address before detailing how we construct the tonal graph is the concept of erosion applied to a piano roll by a harmony. Our input consists of $A \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{N}_{12}}$ along with a harmony represented by an element $H \in \mathcal{A}_2^{\mathbb{Z}_{12}\times I}$. To achieve this, we apply a similar approach to the one discussed in Section 5.2 concerning the erosion of a texture. In this case, we treat A as an element of $\mathcal{A}_2^{\mathcal{T}\times\mathcal{N}_{12}\times\mathbf{0}}$, and H as an element of $\mathcal{A}_2^{\mathbf{0}\times\mathbb{Z}_{12}\times I}$. The erosion operation is then defined as follows:

$$\begin{array}{rccc} A \ominus H : & \mathcal{T}\times\mathcal{N}_{12}\times I & \to & \mathcal{A}_2 \\ & (t,\xi,i) & \mapsto & (A \ominus C_i)(t,\xi) \end{array} \tag{5.25}$$

where $H = (C_i)_{i\in I}$.

We now expose the creation of the tonal graph.

Definition 5.9 (Tonal graph)**.** Let $A \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{N}_{12}}$ be a chroma roll. Let $H = \mathcal{A}_2^{\mathbb{Z}_{12}\times I}$ be a chroma harmony. We consider the erosion of A by H, $A \ominus H \in \mathcal{A}_2^{\mathcal{T}\times\mathcal{N}_{12}\times I}$.

Then, the **tonal graph** of A associated with H is the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where

$$\mathcal{V} = \operatorname{supp}(A \ominus H) \subseteq \mathcal{T} \times \mathcal{N}_{12} \times I, \text{ and}$$
$$\mathcal{E} = \left\{ \big((t_1, \xi_1, i_1), (t_2, \xi_2, i_2) \big) \in \mathcal{V}^2 : \ A|_{]t_1, t_2[} = \emptyset \right\}.$$

Let us discuss in detail the interest of this construction. Each vertex takes the form of a triplet $(t, \xi, i) \in \mathcal{T} \times \mathcal{N}_{12} \times I$ representing a potential interpretation of the chord as a Roman numeral. Here is how the components of this triplet shall be interpreted:

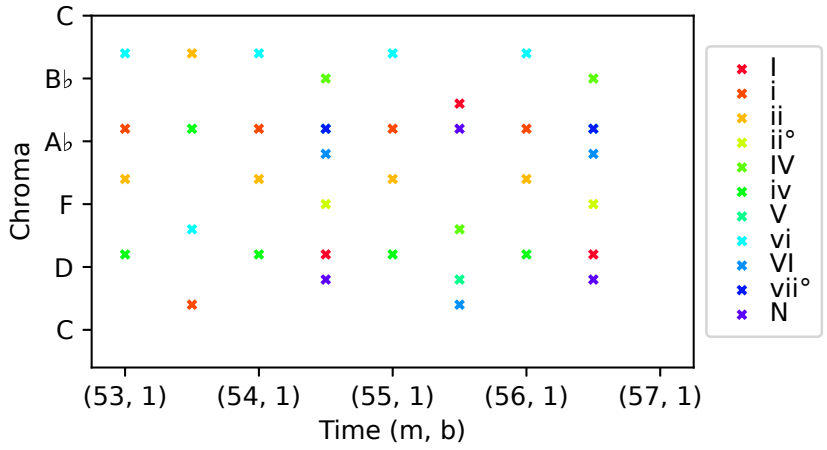
- the time value t designates the moment when the chord is present within the musical piece,
- the chroma ξ represents the tonic, that is the key to which the Roman numeral is subordinated,
- the index i corresponds to the Roman numeral, characterizing the harmonic label.

For any two successive time points t_1 and t_2, i.e., $A|_{]t_1, t_2[} = \emptyset$, all vertices existing at time t_1 are linked to all vertices at time t_2. This makes the graph a chain of complete bipartite graphs. Moreover, traversing a path within the graph signifies an interpretation of each chord present in the input as a Roman numeral in a key.

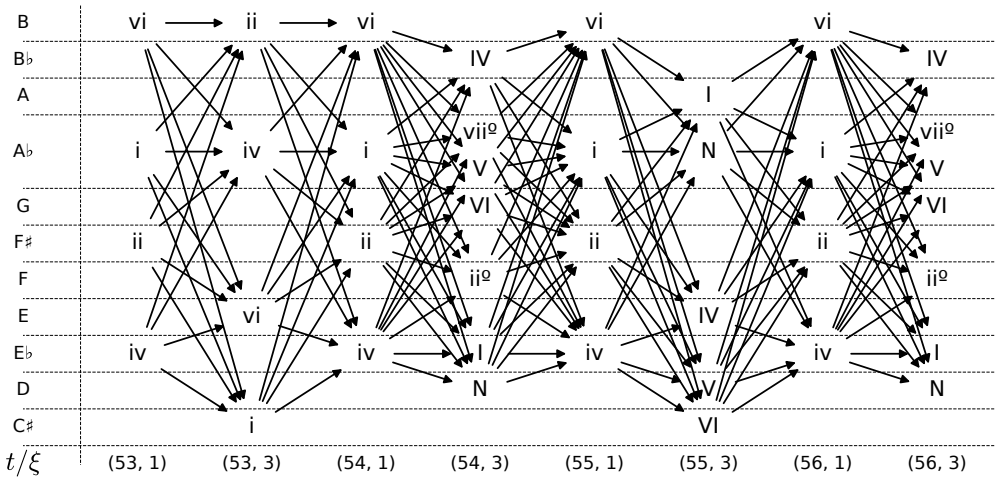
Let us illustrate that with an example.

Example 5.10. *We consider the activations piano roll* $A \in \mathcal{A}_2^{\mathcal{T} \times \mathcal{N}_{12}}$ *from Figure 5.10.* $H = \mathcal{A}_2^{\mathbb{Z}_{12} \times I}$ *is the harmony exposed in Table 5.1.*

Then, the erosion $A \ominus H$ *is*



We can then build the tonal graph that may be represented as



We now aim to reformulate the task of identifying the correct Roman numeral analysis[10] into a shortest path problem.

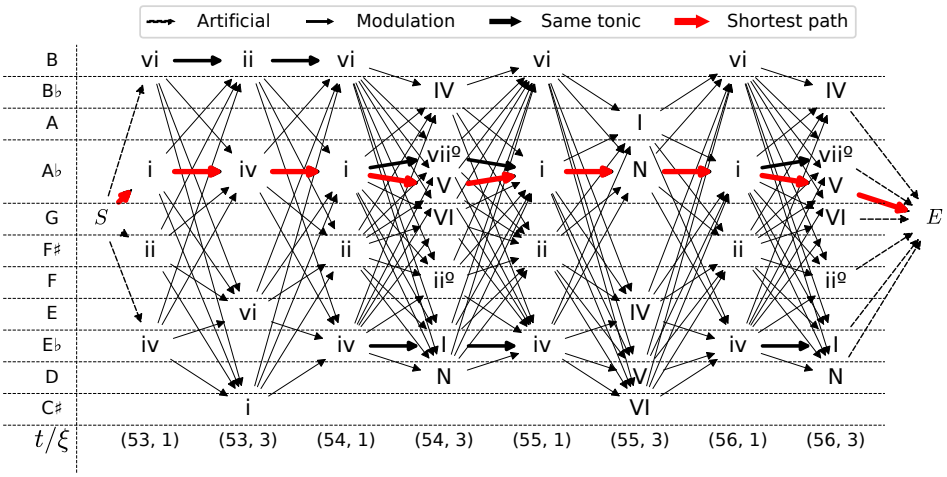
To achieve this, we must assign weights to each edge, such that we have a shortest path problem. One of the simpler yet effective methods involves assigning a weight denoted as w using the formula:

$$w\big((t_1, \xi_1, i_1), (t_2, \xi_2, i_2)\big) = \begin{cases} 1 & \text{if } \xi_1 \neq \xi_2 \\ 0 & \text{if } \xi_1 = \xi_2 \end{cases} \tag{5.26}$$

which means that we count the number of modulations. This way, our shortest path problem is a minimization of modulations (which makes a lot of musical sense).

We add a start node S connected to the first nodes and an end node E connected to the final nodes and we have then the graph

[10]The existence of a (single) correct interpretation may be discussed by musicologists, but there are a lot of cases where there is no doubt, as in this example.



and the shortest path

$$(\mathrm{G}\sharp : \mathbf{i}, \mathrm{G}\sharp : \mathbf{iv}, \mathrm{G}\sharp : \mathbf{i}, \mathrm{G}\sharp : \mathbf{V}, \mathrm{G}\sharp : \mathbf{i}, \mathrm{G}\sharp : \mathbf{N}, \mathrm{G}\sharp : \mathbf{i}, \mathrm{G}\sharp : \mathbf{V}),$$

which gives us the expected solution.

It is important to highlight that the algorithm favored **V** over **vii°** when the chord was $\mathbf{V}^7$ (and thus the erosion detected both of them) solely due to the sequence order in the harmony, while both options shared an identical path length. The issue of the sevenths will be discussed later.

5.3.3 Application to Other Pieces

Let us show its application to more complex pieces to see its effectiveness. We start by Bach's Prelude 1 in C major from the first book of the Well Tempered Clavier. Following our established approach, we take the output from the preceding section, contract it in I, and generate a chroma roll of activations to which we apply the erosion and generate the tonal graph.

The expected result that we target is exposed in Figure 5.11. The input chroma roll[11] is presented in Figure 5.12.

We choose as previously the harmony of Table 5.1. The resulting graph[12] is presented in Figure 5.13.

[11]Technically, we shall represent the chroma roll with crosses since it represents activations, but we have decided to represent it with black and white for improving clarity.

[12]We omitted the edges (except the shortest path ones) for clarity but they are obvious because it is a chain of complete bipartite graphs.

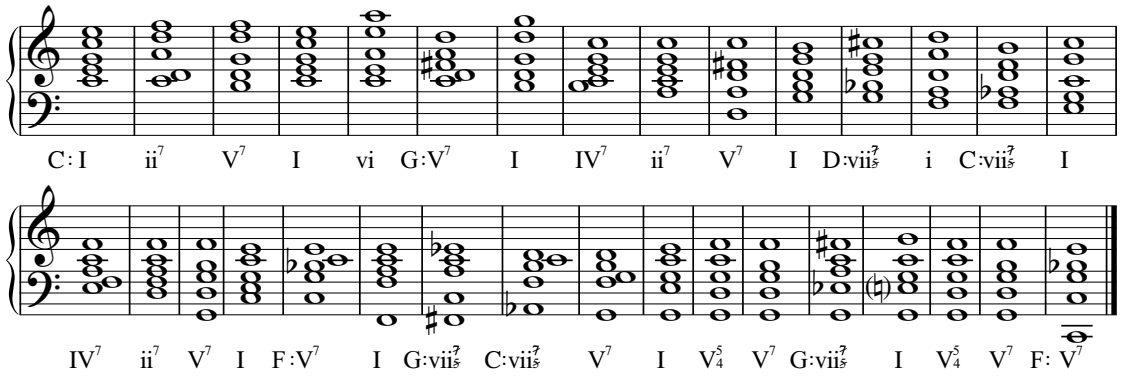


Figure 5.11: Roman numeral analysis of Bach's Prelude in C major, BWV 846.

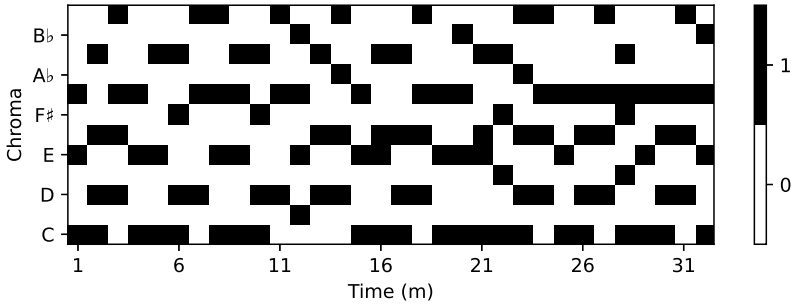


Figure 5.12: Chroma roll of Bach's Prelude in C major, BWV 846 with chords played at once.

Table 5.2 presents a comparative analysis between the expected analysis (the one of Figure 5.11) and the outcome derived from our shortest path algorithm. Focusing solely on the agreement between the tonic and the Roman numeral (excluding sevenths and assuming that $\mathbf{7^o}$ corresponds to **vii**) we observe 22 out of 32 matching instances, resulting in an accuracy rate of 68.75 %.

In the harmony table provided in Table 5.1, our consideration was limited to triads. However, for a thorough analysis of tonal music, the inclusion of seventh chords is necessary. Yet, the challenge arises when seventh chords can be perceived as a supremum of two triads; for instance, $\mathbf{ii} \vee \mathbf{IV} = \{\overline{2}, \overline{5}, \overline{9}\} \vee \{\overline{5}, \overline{9}, \overline{10}\} = \{\overline{2}, \overline{5}, \overline{9}, \overline{10}\} = \mathbf{ii^7}$. Consequently, the erosion detects both **ii** and **IV** along with $\mathbf{ii^7}$.

To address this scenario and prioritize seventh chords, a strategic approach in-

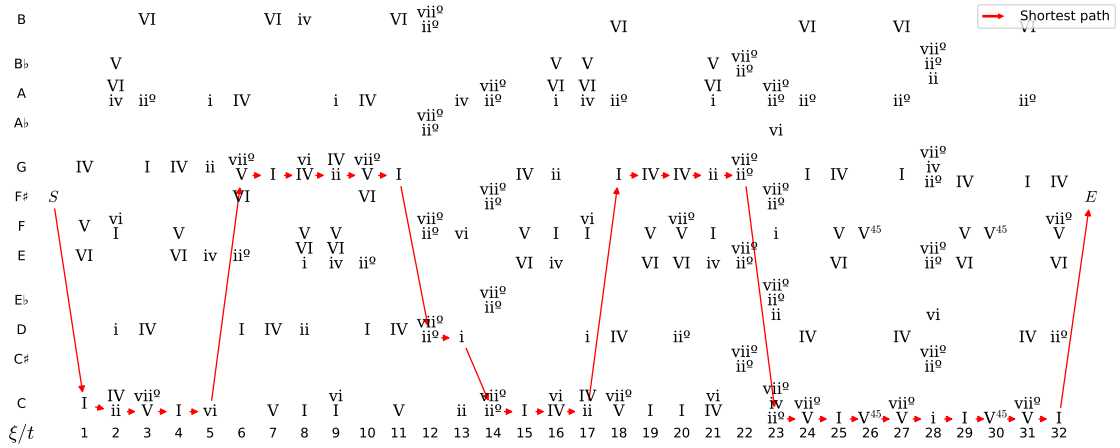


Figure 5.13: Tonal graph of Bach's Prelude in C major, BWV 846.

Measure	1	2	3	4	5	6	7	8	9	10	11
Actual	C: **I**	**ii**	**V**	**I**	**vi**	G: **V**	**I**	**IV**	**ii**	**V**	**I**
Expected	C: **I**	**ii^7**	**V^7**	**I**	**vi**	G: **V^7**	**I**	**IV^7**	**ii^7**	**V^7**	**I**
Measure	12	13	14	15	16	17	18	19	20	21	22
Actual	D: **iiº**	**i**	C: **iiº**	**I**	**IV**	**ii**	G: **I**	**IV**	**IV**	**ii**	**iiº**
Expected	D: **7º**	**i**	C: **7º**	**I**	**IV^7**	**ii^7**	**V^7**	**I**	F: **V^7**	**I**	G: **7º**
Measure	23	24	25	26	27	28	29	30	31	32	
Actual	C: **iiº**	**V**	**I**	**V^{45}**	**V**	**i**	**I**	**V^{45}**	**V**	**I**	
Expected	C: **7º**	**V^7**	**I**	**V^{457}**	**V^7**	G: **7º**	C: **I**	**V^{457}**	**V^7**	F: **V^7**	

Table 5.2: Result of the analysis of Bach's Prelude in C major, BWV 846 by means of the tonal graph.

volves assigning a reduced weight to edges leading to seventh chords[13]. For instance, we can deduct an arbitrary small amount[14] from the weight of each arrow directed towards a seventh chord.

The new chords that we add to the harmony presented in Table 5.1 are shown in Table 5.3.

While we omit the presentation of the new graph due to its similarity to the previous one, the corresponding results are captured in Table 5.4. In this updated analysis, we have achieved 24 accurate chord identifications out of 32, resulting in a

[13]Actually, we may also play with the order of exploration of the graph to check later the seventh chords, but is not a robust approach in our opinion.

[14]We chose 0.1 for practical computations.

RN	Chord	RN	Chord	RN	Chord
$\mathbf{I^7}$	$\{\overline{0}, \overline{4}, \overline{7}, \overline{11}\}$	$\mathbf{IV^7}$	$\{\overline{0}, \overline{5}, \overline{9}, \overline{4}\}$	**vi**	$\{\overline{0}, \overline{4}, \overline{9}, \overline{7}\}$
$\mathbf{i^7}$	$\{\overline{0}, \overline{3}, \overline{7}, \overline{10}\}$	$\mathbf{iv^7}$	$\{\overline{0}, \overline{5}, \overline{8}, \overline{3}\}$	$\mathbf{VI^7}$	$\{\overline{0}, \overline{3}, \overline{8}, \overline{7}\}$
$\mathbf{ii^7}$	$\{\overline{2}, \overline{5}, \overline{9}, \overline{0}\}$	$\mathbf{V^7}$	$\{\overline{2}, \overline{7}, \overline{11}, \overline{5}\}$	$\mathbf{vii^{o7}}$	$\{\overline{2}, \overline{5}, \overline{11}, \overline{9}\}$
$\mathbf{ii^{o7}}$	$\{\overline{2}, \overline{5}, \overline{8}, \overline{0}\}$	$\mathbf{V^{457}}$	$\{\overline{0}, \overline{2}, \overline{7}, \overline{5}\}$	$\mathbf{7^o}$	$\{\overline{0}, \overline{2}, \overline{7}, \overline{8}\}$

Table 5.3: Seventh chords extending the harmony presented in Table 5.1.

75 %accuracy rate.

Notably, further enhancements can be made to address the remaining errors. Two notable avenues for improvement are evident: first, certain options should be ruled out as possibilities; for example, the last chord cannot be a **I** in C because there is a B♭; random modulations in various places can be rectified – such as the modulation occurring one measure earlier in measure 12, where we want to enforce the $\mathbf{V^7}$ to resolve into the **I**.

We initiate the refinement process by implementing the first technique, employing the hit-or-miss transform. Subsequently, we proceed to apply the second technique involving the utilization of weights that are depend on the Roman numerals of the vertices.

Measure	1	2	3	4	5	6	7	8	9	10	11
Actual	C: **I**	$\mathbf{ii^7}$	$\mathbf{V^7}$	**I**	**vi**	G: $\mathbf{V^7}$	**I**	$\mathbf{IV^7}$	$\mathbf{ii^7}$	$\mathbf{V^7}$	D: **IV**
Expected	C: **I**	$\mathbf{ii^7}$	$\mathbf{V^7}$	**I**	**vi**	G: $\mathbf{V^7}$	**I**	$\mathbf{IV^7}$	$\mathbf{ii^7}$	$\mathbf{V^7}$	**I**
Measure	12	13	14	15	16	17	18	19	20	21	22
Actual	$\mathbf{7^o}$	C: **ii**	$\mathbf{7^o}$	**I**	$\mathbf{IV^7}$	$\mathbf{ii^7}$	$\mathbf{V^7}$	**I**	**I**	$\mathbf{IV^7}$	C♯: $\mathbf{7^o}$
Expected	D: $\mathbf{7^o}$	**i**	C: $\mathbf{7^o}$	**I**	$\mathbf{IV^7}$	$\mathbf{ii^7}$	$\mathbf{V^7}$	**I**	F: $\mathbf{V^7}$	**I**	G: $\mathbf{7^o}$
Measure	23	24	25	26	27	28	29	30	31	32	
Actual	C: $\mathbf{ii^{o7}}$	$\mathbf{V^7}$	**I**	$\mathbf{V^{457}}$	$\mathbf{V^7}$	**i**	**I**	$\mathbf{V^{457}}$	$\mathbf{V^7}$	**I**	
Expected	C: $\mathbf{7^o}$	$\mathbf{V^7}$	**I**	$\mathbf{V^{457}}$	$\mathbf{V^7}$	G: $\mathbf{7^o}$	C: **I**	$\mathbf{V^{457}}$	$\mathbf{V^7}$	F: $\mathbf{V^7}$	

Table 5.4: Result of the analysis using seventh chords.

By adopting a hit or miss transform instead of an erosion, we gain the ability to not only specify which chromas we desire but also those we wish to exclude. To achieve this, we introduce two harmonies: one for the "contains" condition (equivalent to erosion) and another for the "contained" condition (complemented erosion). For the second harmony, we use scales like the major scale Maj $= \{\overline{0}, \overline{2}, \overline{4}, \overline{5}, \overline{7}, \overline{9}, \overline{11}\}$ or a combination of minor scales (natural, harmonic, and melodic). This refined approach yields an accuracy of 27 out of 32 (84.375 %), with the remaining errors attributed to timing discrepancies in modulations.

To address this timing issue, we can weight differently the chains in function of the corresponding Roman numeral. For instance, we can assign lower weights to $\mathbf{V}^7 - \mathbf{I}$ and $\mathbf{7^o} - \mathbf{I}$ transitions. By reducing the weight by 0.1 to edges linking these Roman numerals (in major and minor modes), we achieve a success rate of 29 out 32 (90.625 %). The remaining errors can be summarized as follows:

1. In measure 22, we get C♯: $\mathbf{7^o}$ instead of G: $\mathbf{7^o}$; this is due to the equivalence between these two chords and the fact that there is no G chord before or after.

2. In measure 23, we get $\mathbf{ii^{o7}}$ instead of C: $\mathbf{7^o}$; this is interesting because whereas we have affirmed the the correct chord is C: $\mathbf{7^o}$, there may be an argument to defend the actual output[15].

3. In measure 28, we get E: $\mathbf{7^o}$ instead of G: $\mathbf{7^o}$; this is the same problem than the first item.

5.3.4 Conclusion

In conclusion, the construction of tonal graphs proves to be remarkably beneficial for the analysis of chroma rolls comprised of chords. The examples we have presented are just a glimpse of the potential; by adjusting harmonies, weights, edges, or even incorporating higher-order weights derived from the graph, we can significantly enhance their accuracy. Notably, future research may involve using machine learning to learn edge weights, offering an avenue for refinement and optimization in the analysis process.

[15]This argument is that it is followed by a $\mathbf{V}$ and the $\mathbf{ii^o} - \mathbf{V}$ movement is very present in tonal music.

Conclusions and Perspectives

Throughout this thesis, we have demonstrated the utility of mathematical morphology in the analysis and generation of time-frequency representations of music. Our focus has centered on two primary representations: spectrograms and piano rolls. In the following, we will provide a summary of the main findings, their respective successes, and the contributions and future prospects of our research.

Overall, MM proves effective in both time-frequency representations. Nevertheless, the approach we employed for each of them differs significantly. While we adopted a more traditional approach for spectrograms, MM reveals itself as more meaningful and adaptable when applied to piano rolls. Let us delve deeper into each application.

Spectrograms are very useful representations of sound, but they lack an important element: phase information. While this is not problematic in most cases, it does impose constraints on certain applications. For instance, the sum of two spectrograms is not the spectrogram of the sum, which makes linear analysis inappropriate. Furthermore, the spectrogram of the combination of two sounds is not the supremum of the spectrograms, which violates one of the basic assumptions for using MM. This fact challenges the applicability of the MM framework.

In particular, the presence of interferences cannot be effectively addressed by the STN model. When our processing pipeline (see Figure 3.3) is applied to spectrograms with interferences, the output proves inadequate. This raises questions about the suitability of these methods for critical cases. Future research may explore the application of MM to a different class of spectrograms: those derived from the time-frequency-scale transformation. This transformation could mitigate time-frequency uncertainty and resolve interference issues.

Despite these challenges, our current pipeline performs well at detecting both horizontal and vertical lines. Lines in the spectrogram are a crucial component of many musical instruments, and we perform correctly at detecting them in such instruments. Furthermore, the pipeline effectively identifies noise, which is particu-

larly important for other types of instruments, especially percussion ones. While the outcomes of this procedure are promising, there is significant room for refinement; the results do not yet capture the purity of the original sounds. This issue could be attributed either to the synthesis model, the detection model, or most likely, a combination of both. Subsequent research may illuminate the way to integrate these two components, resulting in a more coherent and robust model.

Our current pipeline can replicate the sound of a musical instrument but does not identify the specific instrument or the number of instruments present. This task, called polyphonic multi-pitch estimation, exceeds our current capabilities and was explored at the outset of the doctoral research without success.

However, we believe that the geometry of spectrograms remains largely an uncharted territory. The presence of holes in noisy spectrograms, complementary to lines, may be well adapted to MM operators. Given our success in detecting holes and lines, a model that generates sounds while adhering to these constraints might prove more effective. We propose that future research could explore an approach based on what we might call *ridges and sinks* model, that would be more adapted to MM.

Transitioning to the realm of piano rolls, MM emerges as an elegant and seamless approach. Furthermore, it catalyzed the development of a new formalism of greyscale MM based on residuated triplets.

The introduction of residuated triplets aligns effectively with the demands of MM. It underscores a critical distinction between the interpretation of inputs in erosion/dilation operations and their corresponding outputs. In standard MM, both are interpreted as images, but with residuated triplets, the lattices hold distinct meanings, influencing the interpretation. Our model typically interprets the input of a dilation (or the output of an erosion) as an *activation*, while the input of an erosion (or the output of a dilation) represents the object itself, in our case, piano rolls.

Also important is the relationship between the space and the group that acts on it, especially in the context of music. An input piano roll represents a musical piece, whereas a structuring element represents a motif. The group elements encompass rhythms in the time dimension and Roman numerals in the frequency dimension, which are combined to generate motives. This framework enables us to understand a musical piece as a composition of building blocks (the motives) activated through the dilation operator.

While mathematical morphology has been at the forefront of our discussion, we

must acknowledge the utilization of other tools in our research, notably tensor products and graphs. Let us delve into their roles and significance.

Tensor product, a versatile mathematical construct, finds application across numerous domains. In our case, it served as a means to encapsulate the concept of a chord distributed rhythmically over time. Through the amalgamation of diverse rhythms and chords, i.e., texture and harmony, we crafted intricate musical motifs and elucidated entire compositions.

We posit that the use of texture and harmony as compositional tools bears significant potential, particularly in the realm of education. The amalgamation of these parameters can serve as a fertile playground for aspiring musicians, offering a novel and meaningful approach to music creation.

While we proposed a XML approach to create music, it is not intended to be a way of writing music. Instead, we envision the development of a user-friendly interface that streamlines the process and abstracts the XML from the user. Such an interface could potentially serve as an alternative to traditional score editors and sequencers, eliminating the need for users to know staff notation. Furthermore, it could incorporate suggestions to enhance the intuitiveness of music creation. An implementation of this interface in OpenMusic (Agón, 1998) is ongoing.

Expanding upon the concept of texture and harmony, their integration could transcend mere composition; it may extend into the realm of Artificial Intelligence (AI) applied to music generation and analysis. An AI agent, equipped to comprehend musical pieces or segments using these factors, may find alignment with common neural networks, given the inherent array-like nature of these musical objects. Additionally, genetic algorithms, capable of manipulating parameters individually to produce incremental changes akin to musical evolution, could play a role. We ardently believe that this approach represents a significant stride in advancing the understanding of music by machines.

Another notable facet of our research is the application of graph theory, particularly the extensive use of chains of bipartite graphs. These graphs represent the concept of sequential decision-making, mirroring the process of musical piece analysis in a meaningful manner.

They perform particularly well for Roman numeral analysis. However, there are still many unexplored possibilities. A promising avenue for future research lies in determining the correct weights for the different edges using machine learning techniques with annotated corpora. Additionally, the utilization of derived graphs may prove beneficial in transitioning from sequences of two chords to more complex chord sequences.

The use of such graphs for the extraction of a minimal set of activations has shown

theoretically promising but practically limited. This challenge remains for the future, possibly necessitating the development of novel tools for resolution. Furthermore, it is worth noting that our assumptions regarding prior knowledge of the texture are not warranted *a priori*.

The quest for a unified method to simultaneously uncover texture and harmony is, in our opinion, one of the most exhilarating challenges this thesis presents. We view it as an *all-in* analysis that can provide a comprehensive understanding of a musical piece at various levels: from the hierarchical perspective at a broader level, through the identification of the motives that constitute a piece at a medium level, to the recognition of individual notes as harmonic elements related to a tonic at a lower level. Such an approach could foster a compelling human-machine interaction, with machines proposing possibilities and humans making decisions.

List of publications

Romero-García, G. (2022). *nnMorpho, a PyTorch library for mathematical morphology operators* (Poster at IAPR Second International Conference on Discrete Geometry and Mathematical Morphology).

Romero-García, G., Agón, C., & Bloch, I. (2022). Estimation de paramètres de resynthèse de sons d'instruments de musique avec des outils de morphologie mathématique. *Proceedings of the 19th Sound and Music Computing Conference*, 653–662.

Romero-García, G., Bloch, I., & Agón, C. (2022). Mathematical morphology operators for harmonic analysis. *Proceedings of the 8th international conference of mathematics and computation in music* (pp. 255–266).

Romero-García, G., Guichaoua, C., & Chew, E. (2022). A model of rhythm transcription as path selection through approximate common divisor graphs. *Proceedings of the 7th International Conference on Technologies for Music Notation and Representation*.

Romero-García, G., Lascabettes, P., & Chew, E. (2022). Automated musical rhythm transcription of ECG RR interval time series as a tool for representing rhythm variations and annotation anomalies in arrhythmia heartbeat classifications. *498*, 1–4.

Bibliography

Agmon, E. (1995). Functional harmony revisited: A prototype-theoretic approach. *Music Theory Spectrum*, *17*(2), 196–214.

Agon, C., Haddad, K., & Assayag, G. (2002). Representation and rendering of rhythm structures. *Proceedings Second International Conference on WEB Delivering of Music*, 109–113.

Agón, C. (1998). *OpenMusic : Un langage visuel pour la composition musicale assistée par ordinateur* (These de doctorat). Paris 6.

Andreatta, M. (2004). On group-theoretical methods applied to music: Some compositional and implementational aspects. *Perspectives in mathematical and computational music theory* (pp. 169–193).

Ballet, G., Borghesi, R., Hoffmann, P., & Lévy, F. (1999). Studio online 3.0: An internet "killer application" for remote access to IRCAM sounds and processing tools. *Proceedings of the Journées d'Informatique Musicale*, 123–132.

Barat, C., Ducottet, C., & Jourlin, M. (2003a). Line pattern segmentation using morphological probing. *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis*, *1*, 417–422.

Barat, C., Ducottet, C., & Jourlin, M. (2003b). Pattern matching using morphological probing. *Proceedings of the 2003 International Conference on Image Processing*, *1*, I–369.

Beineke, L. W. (1968). Derived graphs and digraphs. *Proceedings of Beiträge zur graphentheorie*, 17–33.

Beineke, L. W. (1970). Characterizations of derived graphs. *Journal of Combinatorial Theory*, *9*(2), 129–135.

Beineke, L. W., & Bagga, J. S. (2021). *Line graphs and line digraphs*. Springer.

Beineke, L. W., & Zamfirescu, C. M. (1982). Connection digraphs and second-order line digraphs. *Discrete Mathematics*, *39*(3), 237–254.

Berlioz, H. (1844). *Grand traité d'instrumentation et d'orchestration modernes*. Schonenberger.

Bilmes, J. A. (1993). *Timing is of the essence: Perceptual and computational techniques for representing, learning, and reproducing expressive timing in percussive rhythm* (Master's Thesis). Massachusetts Institute of Technology.

Birkhoff, G. (1948). *Lattice theory* (2nd ed., Vol. 25). American Mathematical Society.

Bloch, I. (2009). Duality vs. adjunction for fuzzy mathematical morphology and general form of fuzzy erosions and dilations. *Fuzzy Sets and Systems*, *160*(13), 1858–1867.

Bloch, I. (2012). Mathematical morphology on bipolar fuzzy sets: General algebraic framework. *International Journal of Approximate Reasoning*, *53*(7), 1031–1060.

Bloch, I., Heijmans, H., & Ronse, C. (2007). Mathematical morphology. *Handbook of spatial logics* (pp. 857–944).

Bloch, I., & Maître, H. (1994). Fuzzy mathematical morphology. *Annals of Mathematics and Artificial Intelligence*, *10*(1), 55–84.

Boashash, B. (2016). Time-frequency synthesis and filtering. *Time-frequency signal analysis and processing* (2nd ed., pp. 637–691).

Bonada, J. (2000). Automatic technique in frequency domain for near-lossless time-scale modification of audio. *Proceedings of the 2000 International Computer Music Conference*.

Bonada, J. (2002). *Audio time-scale modification in the context of professional post-production* (Pre-Doctoral research). Universitat Pompeu Fabra.

Bradski, G. (2000). The OpenCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, *25*(11), 122–125.

Brown, J. C. (1991). Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, *89*(1), 425–434.

Butterworth, S. (1930). On the theory of filter amplifiers. *Experimental Wireless and the Wireless Engineer*, *7*, 536–541.

Cadore, J., Gallardo-Antolín, A., & Peláez-Moreno, C. (2011). Morphological processing of spectrograms for speech enhancement. *Proceedings of the 5th International Conference on Nonlinear Speech Processing*, 224–231.

Carnovalini, F., Rodà, A., Harley, N., Homer, S. T., & Wiggins, G. A. (2021). A new corpus for computational music research and a novel method for musical structure analysis. *Proceedings of the 16th International Audio Mostly Conference*, 264–267.

Chen, T.-P., & Su, L. (2018). Functional harmony recognition of symbolic music data with multi-task recurrent neural networks. *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 90–97.

Cheuk, K. W., Anderson, H., Agres, K., & Herremans, D. (2020). nnAudio: An on-the-fly GPU audio to spectrogram conversion toolbox using 1d convolutional neural networks. *IEEE Access*, *8*, 161981–162003.

Chomsky. (1965). *Aspects of the theory of syntax*. MIT Press.

Concas, A., Reichel, L., Rodriguez, G., & Zhang, Y. (2021). Chained graphs and some applications. *Applied Network Science*, *6*(1), 1–29.

Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, *19*(90), 297–301.

Couturier, L., Bigo, L., & Levé, F. (2022a). Annotating symbolic texture in piano music: A formal syntax. *Proceedings of Sound and Music Computing 2022*, 577–584.

Couturier, L., Bigo, L., & Levé, F. (2022b). A dataset of symbolic texture annotations in mozart piano sonatas. *Proceedings the 23rd International Society for Music Information Retrieval Conference*, 509–516.

De Haas, W. B., Magalhães, J. P., Wiering, F., & C. Veltkamp, R. (2013). Automatic functional harmonic analysis. *Computer Music Journal*, *37*(4), 37–53.

Deng, T.-Q., & Heijmans, H. J. (2002). Grey-scale morphology based on fuzzy logic. *Journal of Mathematical Imaging and Vision*, *16*(2), 155–171.

Devaney, J., Arthur, C., Condit-Schultz, N., & Nisula, K. (2015). Theme and variation encodings with roman numerals (TAVERN): A new data set for symbolic music analysis. *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 728–734.

Dorran, D. (2005). *Audio time-scale modification* (Doctoral dissertation). Dublin Institute of Technology.

Driedger, J., Müller, M., & Disch, S. (2014). Extending harmonic-percussive separation of audio signals. *Proceedings of the 15th Conference of the International Society for Music Information Retrieval*, 611–616.

Engel, J., Hantrakul, L., Gu, C., & Roberts, A. (2020). DDSP: Differentiable digital signal processing. *Proceedings of the 8th International Conference on Learning Representations*.

Fierro, L., & Välimäki, V. (2023). Enhanced fuzzy decomposition of sound into sines, transients, and noise. *Journal of the Audio Engineering Society*, *71*(7), 468–480.

Fourier, J. B. J. (1888). *Theorie analytique de la chaleur*. Gauthier-Villars et fils.

Füg, R., Niedermeier, A., Driedger, J., Disch, S., & Müller, M. (2016). Harmonic-percussive-residual sound separation using the structure tensor on spectrograms. *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, 445–449.

Galatos, N., Jipsen, P., Kowalski, T., & Ono, H. (2007). *Residuated lattices: An algebraic glimpse at substructural logics*. Elsevier.

Giraud, M., Levé, F., Mercier, F., Rigaudière, M., & Thorez, D. (2014). Towards modeling texture in symbolic data. *Proceedings of the 15th Conference of the International Society for Music Information Retrieval*, 59–64.

Goto, M., Hashiguchi, H., Nishimura, T., & Oka, R. (2002). RWC music database: Popular, classical and jazz music databases. *Proceedings of the 3rd International Conference on Music Information Retrieval*, 287–288.

Gröchenig, K. (2001). *Foundations of time-frequency analysis*. Birkhäuser.

Guigue, D., & de Paiva Santana, C. (2018). The structural function of musical texture: Towards a computer-assisted analysis of orchestration. *Proceedings of the Journées d'Informatique Musicale*.

Gustafsson, F. (1996). Determining the initial states in forward-backward filtering. *IEEE Transactions on Signal Processing*, *44*(4), 988–992.

Harary, F., & Norman, R. Z. (1960). Some properties of line digraphs. *Rendiconti del Circolo Matematico di Palermo*, *9*(2), 161–168.

Heijmans, H. J. A. M., & Ronse, C. (1990). The algebraic basis of mathematical morphology i. dilations and erosions. *Computer Vision, Graphics, and Image Processing*, *50*(3), 245–295.

Helmholtz, H. v. (1865). *Die Lehre von den Tonempfindungen als physiologische Grundlage für die Theorie der Musik*. Friedrich Vieweg.

Herold, N. (2012). Timbre et analyse musicale : Les possibilités d'intégration du timbre dans l'analyse formelle des œuvres pour piano du dix-neuvième siècle. *L'interprétation musicale* (pp. 79–103).

Illescas, P., Rizo, D., & Quereda, J. (2007). Harmonic, melodic, and functional automatic analysis. *Proceedings of the 2007 International Computer Music Conference*.

ISO. (1975). *Acoustics — standard tuning frequency (standard musical pitch)* (Standard ISO 16:1975). International Organization for Standardization.

ISO. (2019a). *Date and time — representations for information interchange — part 1: Basic rules* (Standard ISO 8601-1:2019). International Organization for Standardization.

ISO. (2019b). *Quantities and units — part 2: Mathematics* (Standard ISO 80000-2:2019). International Organization for Standardization.

Jacquemard, F., Donat-Bouillud, P., & Bresson, J. (2015). A structural theory of rhythm notation based on tree representations and term rewriting. *5th international conference of mathematics and computation in music* (pp. 3–15).

Jacquemard, F., Ycart, A., & Sakai, M. (2017). Generating equivalent rhythmic notations based on rhythm tree languages. *Proceedings of the 3rd International Conference on Technologies for Music Notation and Representation*, 145–153.

Jipsen, P., & Tsinakis, C. (2002). A survey of residuated lattices. *Ordered Algebraic Structures: Proceedings of the Gainesville Conference Sponsored by the University of Florida 28th February*, 19–56.

Juillerat, N., Arisona, S. M., & Schubiger-Banz, S. (2008). Enhancing the quality of audio transformations using the multi-scale short-time fourier transform. *Proceedings of the 10th IASTED International Conference on Signal and Image Processing*.

Karvonen, M. (2008). *Using mathematical morphology for geometric music retrieval* (Master's Thesis). University of Helsinki.

Karvonen, M., Laitinen, M., Lemström, K., & Vikman, J. (2011). Error-tolerant content-based music-retrieval with mathematical morphology. *Proceedings of the 7th International Symposium on Computer Music Modeling and Retrieval*, 321–337.

Karvonen, M., & Lemström, K. (2008). Using mathematical morphology for geometric music information retrieval. *Proceedings of the 2008 International Workshop on Machine Learning and Music*.

Koelsch, S., Rohrmeier, M., Torrecuso, R., & Jentschke, S. (2013). Processing of hierarchical syntactic structure in music. *Proceedings of the National Academy of Sciences*, *110*(38), 15443–15448.

Krausz, J. (1943). Démonstration nouvelle d'une théoreme de whitney sur les réseaux. *Matematikai és fizikai lapok*, *50*(1), 75–85.

Krull, W. (1924). Axiomatische begründung der allgemeinen idealtheorie. *Sitzungsberichte der physikalisch medizinischen Societat der Erlangen*, *56*, 47–63.

Lantuéjoul, C. (1978). *La squelettisation et son application aux mesures topologiques des mosaiques polycristallines* (Doctoral dissertation). École des Mines.

Lascabettes, P., Bloch, I., & Agon, C. (2020). Analyse de représentations spatiales de la musique par des opérateurs simples de morphologie mathématique. *Proceedings of Journées d'Informatique Musicale 2020*.

Lascabettes, P. (2019). *Mathematical morphology applied to music* (Master's Thesis). École Normale Supérieure - Paris Saclay.

Lascabettes, P. (2023). *Mathematical models for the discovery of musical patterns, structures and for performances analysis* (Doctoral dissertation). École Normale Supérieure - Paris Saclay.

Lascabettes, P., Agon, C., Andreatta, M., & Bloch, I. (2022). Computational analysis of musical structures based on morphological filters. *Proceedings of the 8th International Conference Mathematics and Computation in Music*, 267–278.

Lerdahl, F. (1983). *A generative theory of tonal music*. MIT Press.

Lerdahl, F., & Jackendoff, R. (1983). An overview of hierarchical structure in music. *Music Perception: An Interdisciplinary Journal*, *1*(2), 229–252.

Levine, S., Verma, T., & Smith, J. (1998). Multiresolution sinusoidal modeling for wideband audio with modifications. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, *6*, 3585–3588.

Lewin, D. (1987). *Generalized musical intervals and transformations*. Oxford University Press.

López, N. N. (2017). *Automatic harmonic analysis of classical string quartets from symbolic score* (Master's Thesis). Universitat Pompeu Fabra.

López, N. N., Gotham, M., & Fujinaga, I. (2021). AugmentedNet: A roman numeral analysis network with synthetic training examples and additional tonal tasks. *Proceedings of the 22nd International Society for Music Information Retrieval Conference*, 404–411.

Maragos, P. (2005). Lattice image processing: A unification of morphological and fuzzy algebraic systems. *Journal of Mathematical Imaging and Vision*, *22*(2), 333–353.

Marsden, A., Hirata, K., & Tojo, S. (2013). Towards computable procedures for deriving tree structures in music : Context dependency in GTTM and schenkerian theory. *Proceedings of the Sound and Music Computing Conference 2013*, 360–367.

Mateo, C., & Talavera, J. A. (2020). Bridging the gap between the short-time fourier transform (STFT), wavelets, the constant-q transform and multi-resolution STFT. *Signal, Image and Video Processing*, *14*(8), 1535–1543.

Matheron, G. (1965). *Granulométrie en place et milieu poreux aléatoire*. École des Mines de Paris.

Matheron, G. (1967). *Eléments pour une théorie des milieux poreux*. Masson et Cie.

Matheron, G. (1975). *Random sets and integral geometry*. J. Wiley & Sons.

Maxwell, H. J. (1992). An expert system for harmonic analysis of tonal music. *Understanding music with AI* (pp. 147–160).

Micchi, G., Gotham, M., & Giraud, M. (2020). Not all roads lead to rome: Pitch representation and model architecture for automatic harmonic analysis. *Transactions of the International Society for Music Information Retrieval*, *3*(1), 42–54.

Moreira, D. (2019). *Textural design: A compositional theory for the organization of musical texture* (Doctoral dissertation). Universidade Federal do Rio de Janeiro.

Naegel, B., Passat, N., & Ronse, C. (2007). Grey-level hit-or-miss transforms—part II: Application to angiographic image processing. *Pattern Recognition*, *40*(2), 648–658.

Najman, L., & Talbot, H. (2010). *Mathematical morphology: From theory to applications*. Wiley-ISTE.

Neuwirth, M., Harasim, D., Moss, F. C., & Rohrmeier, M. (2018). The annotated beethoven corpus (ABC): A dataset of harmonic analyses of all beethoven string quartets. *Frontiers in Digital Humanities*, *5*.

Nordgren, Q. R. (1960). A measure of textural patterns and strengths. *Journal of Music Theory*, *4*(1), 19–31.

Orio, N., & Roda, A. (2009). A measure of melodic similarity based on a graph representation of the music structure. *Proceedings of the 10th International Society for Music Information Retrieval Conference*, 543–548.

Papadopoulos, A. (2015). Mathematics and group theory in music. *Handbook of group actions, vol. II (ed. l. ji, a. papadopoulos and s.-t. yau)* (p. 525–572.).

Parada-Cabaleiro, E., Schmitt, M., Batliner, A., Schuller, B. W., & Schedl, M. (2021). Automatic recognition of texture in renaissance music. *Proceedings of the 22nd international society for music information retrieval conference* (pp. 509–516).

Pardo, B., & Birmingham, W. P. (2002). Algorithms for chordal analysis. *Computer Music Journal*, *26*(2), 27–49.

Piston, W. (1955). *Orchestration*. Norton.

Rameau, J.-P. (1722). *Traité de l'harmonie : réduite à des principes naturels*. Jean-Baptiste-Christophe Ballard.

Riba, E., Mishkin, D., Ponsa, D., Rublee, E., & Bradski, G. (2020). Kornia: An open source differentiable computer vision library for PyTorch. *Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision*, 3663–3672.

Riemann, H. (1893). *Vereinfachte harmonielehre; oder, die lehre von den tonalen funktionen der akkorde*. Augener.

Ronse, C. (1996). A lattice-theoretical morphological view on template extraction in images. *Journal of Visual Communication and Image Representation*, *7*(3), 273–295.

Ronse, C., & Heijmans, H. J. A. M. (1991). The algebraic basis of mathematical morphology: II. openings and closings. *CVGIP: Image Understanding*, *54*(1), 74–97.

Ronse, C. (1990). Why mathematical morphology needs complete lattices. *Signal processing*, *21*(2), 129–154.

Rossum, G. V., & Drake, F. L. (2009). *Python 3 reference manual: Python documentation manual part 2*. CreateSpace.

Rotman, J. J. (1994). *An introduction to the theory of groups*. Springer.

Sathiamoorthy, G. (2020). Labeling of chain bipartite graphs. *National Academy Science Letters*, *43*(7), 639–641.

Schörkhuber, C., & Klapuri, A. (2010). Constant-q transform toolbox for music processing. *Proceedings of the 7th Sound and Music Computing Conference*, 3–64.

Serra, J. (1982). *Image analysis and mathematical morphology*. Academic Press.

Serra, X., & Smith, J. (1990). Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, *14*(4), 12–24.

Shepard, R. N. (1964). Circularity in judgments of relative pitch. *The Journal of the Acoustical Society of America*, *36*(12), 2346–2353.

Smith, J. (2011). *Spectral audio signal processing*. W3K Publishing.

Soille, P. (2002). Advances in the analysis of topographic features on discrete images. *Proceedings of the 10th international conference on discrete geometry for computer imagery* (pp. 175–186).

Soille, P. (2013). *Morphological image analysis: Principles and applications*. Springer.

Soum-Fontez, L., Giraud, M., Guiomard-Kagan, N., & Levé, F. (2021). Symbolic textural features and melody/accompaniment detection in string quartets. *Proceedings of the 15th International Symposium on Computer Music Multidisciplinary Research*, 175–184.

Steinberg, R., & O'Shaughnessy, D. (2008). Segmentation of a speech spectrogram using mathematical morphology. *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1637–1640.

Temperley, D. (1997). An algorithm for harmonic analysis. *Music Perception*, *15*(1), 31–68.

Temperley, D. (2002). A bayesian approach to key-finding. *Proceedings of the 2nd international conference music and artificial intelligence* (pp. 195–206).

Temperley, D. (2004). *The cognition of basic musical structures*. MIT Press.

Temperley, D. (2009). A unified probabilistic model for polyphonic music analysis. *Journal of New Music Research*, *38*(1), 3–18.

Toussaint, G. T. (2013). *The geometry of musical rhythm*. Routledge.

Tymoczko, D. (2009). Generalizing musical intervals. *Journal of Music Theory*, *53*(2), 227–254.

Tymoczko, D. (2016). In quest of musical vectors. *Mathemusical conversations* (pp. 256–282).

Tymoczko, D., Gotham, M., Cuthbert, M. S., & Ariza, C. (2019). The romantext format: A flexible and standard method for representing roman numeral analyses. *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 123–129.

Verma, T. S., Levine, S. N., & Meng, T. H.-Y. (1997). Transient modeling synthesis: A flexible analysis/synthesis tool for transient signals. *Proceedings of the 1997 International Computer Music Conference*, 48–51.

Verma, T. S., & Meng, T. H. Y. (2000). Extending spectral modeling synthesis with transient modeling synthesis. *Computer Music Journal*, *24*(2), 47–59.

Verma, T., & Meng, T. (1998). An analysis/synthesis tool for transient signals that allows a flexible sines+transients+noise model for audio. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, 3573–3576.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Vázquez-Baeza, Y. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, *17*(3), 261–272.

Walt, S. v. d., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). Scikit-image: Image processing in python. *PeerJ*, *2*, n.p.

Wang, X., Takaki, S., & Yamagishi, J. (2020). Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 402–415.

Ward, M., & Dilworth, R. P. (1938). Residuated lattices. *Proceedings of the National Academy of Sciences*, *24*(3), 162–164.

Ward, M., & Dilworth, R. P. (1939). Residuated lattices. *Transactions of the American Mathematical Society*, *45*(3), 335–354.

Weber, G. (1832). *Versuch einer geordneten theorie der tonsezkunst zum selbstunterricht. 1: Grammatik der tonsezkunst*. B. Schott's Söhne.

Whitney, H. (1932). Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, *54*(1), 150–168.

Winograd, T. (1968). Linguistics and the computer analysis of tonal harmony. *Journal of Music Theory*, *12*(1), 2–49.

Xu, S., Zhao, X., Duan, C. H., Cao, X. L., Li, H. Y., Liang, S. L., & Wang, S. W. (2014). A mathematical morphological processing of spectrograms for the tone of chinese vowels recognition. *Applied Mechanics and Materials*, *571-572*, 665–671.

Ycart, A., Jacquemard, F., Bresson, J., & Staworko, S. (2016). A supervised approach for rhythm transcription based on tree series enumeration. *Proceedings of the 42nd International Computer Music Conference*, n.p.

Young, R. W. (2005). Terminology for logarithmic frequency units. *The Journal of the Acoustical Society of America*, *11*(1), 134–139.

Youngberg, J., & Boll, S. (1978). Constant-q signal analysis and synthesis. *Proceedings of the 1978 IEEE International Conference on Acoustics, Speech, and Signal Processing*, *3*, 375–378.

Zhang, K., Wang, X., Han, F., & Zhao, H. (2015). The detection of crackles based on mathematical morphology in spectrogram analysis. *Technology and Health Care*, *23*(2), 489–494.

Appendix A

Order theory

The foundations of lattice theory is *order theory*. We recall here some basic concepts.

Definition A.1 (Binary relation)**.** A **binary relation** R between X and Y is a subset of $X \times Y$. We notate $R \subset X \times Y$. It is usually expressed by means of a symbol like $\prec$ that points out the importance of the order in the relation. In this case, we use the notation

$$a \prec b :\Leftrightarrow (a, b) \in R\,.$$

In an abuse of notation, we will say that the symbol $\prec$ *is* the binary relation.
A **binary relation on** X is a binary relation between X and X.

Definition A.2 (Partial order)**.** Let L be a set. A **partial order** $\preceq$ on L is a binary relation on L that satisfies the three following properties:

1. $\forall a \in L,\ a \preceq a$ (Reflexivity)
2. $\forall a, b \in L,\ a \preceq b \wedge b \preceq a \Rightarrow a = b$ (Antisymmetry)
3. $\forall a, b, c \in L,\ a \preceq b \wedge b \preceq c \Rightarrow a \preceq c.$ (Transitivity)

We call $(L, \preceq)$ a **partially ordered set**.

We also recall the definitions of the supremum and infimum.

Definition A.3 (Supremum)**.** Let $(L, \preceq)$ be a partially ordered set. Let $A \subseteq L$. We say that $a_0 \in L$ is the **supremum** of A, and we notate $a_0 = \bigvee A$, if a_0 is the least upper bound of A, i.e.,

1. $\forall a \in A,\ a \preceq a_0,$ (Upper bound)

2. $\forall b \in \{c \in L : \forall a \in A,\, a \preceq b\},\, b \preceq a_0$. (Least upper bound)

We use the notation $a \vee b$ for $\bigvee\{a, b\}$.

Definition A.4 (Infimum)**.** Let $(L, \preceq)$ be a partially ordered set. Let $A \subseteq L$. We say that $a_0 \in L$ is the **infimum** of A, and we notate $a_0 = \bigwedge A$, if a_0 is the greatest lower bound of A, i.e.,

1. $\forall a \in A,\, a_0 \preceq a$, (Lower bound)

2. $\forall b \in \{c \in L : \forall a \in A,\, c \preceq a\},\, b \preceq a_0$. (Greatest lower bound)

We use the notation $a \wedge b$ for $\bigwedge\{a, b\}$.

A direct consequence of the definitions that is used extensively is the following.

Proposition A.5. *Let $(L, \preceq)$ be a partially ordered set. Let $A \subseteq L$. Let $b \in L$.*

1. If $\bigvee A \in L$, then

$$\forall a \in A,\, a \preceq b \Leftrightarrow \bigvee A \preceq b\,. \tag{A.1}$$

2. If $\bigwedge A \in L$, then

$$\forall a \in A,\, b \preceq a \Leftrightarrow b \preceq \bigwedge A\,. \tag{A.2}$$

Appendix B

Functional Analysis

Functional Analysis is the study of spaces made of functions and the operators between them. Let us recall the classical functional spaces.

B.1 Common functional spaces

Let D and C be two sets. We recall that we call C^D the space of functions $f : D \to C$ that have D as domain and C as codomain.

Definition B.1 ($\mathcal{L}^p$ spaces)**.** Let $p \in [1, \infty]$. If $p \in [1, \infty[$, we call the $\mathcal{L}^p(\mathbb{R}; \mathbb{C})$ **space** the set

$$\mathcal{L}^p(\mathbb{R}; \mathbb{C}) = \{f \in \mathbb{C}^{\mathbb{R}} : \int_{\mathbb{R}} |f(x)|^p \, dx < \infty\}/_{\sim} \tag{B.1}$$

where $f \sim g \Leftrightarrow \int_{\mathbb{R}} |f(x) - g(x)|^p \, dx = 0$.

We call the p-norm of a function $f \in \mathcal{L}^p$ to $||f||_p = \int_{\mathbb{R}} |f(x)|^p \, dx$

For $p = \infty$, we use

$$\mathcal{L}^\infty(\mathbb{R}; \mathbb{C}) = \{f \in \mathbb{C}^{\mathbb{R}} : \sup_{x \in \mathbb{R}} |f(x)| < \infty\}/_{\sim} \tag{B.2}$$

where $f \sim g \Leftrightarrow \int_{\mathbb{R}} \sup_{x \in \mathbb{R}} |f(x) - g(x)| = 0$.

We call the ∞-norm of a function $f \in \mathcal{L}^\infty(\mathbb{R}; \mathbb{C})$ to $||f||_\infty = \sup_{x \in \mathbb{R}} |f(x)|$.

All the $\mathcal{L}^p(\mathbb{R}; \mathbb{C})$ spaces are Banach spaces, i.e., complete normed vector spaces. Furthermore, the space $\mathcal{L}^2$ is a Hilbert space with scalar product defined in Definition B.13.

B.2 Common operators

If we have a subtraction operator defined in the domain, we can define the translation operator.

Definition B.2 (Translation operator)**.** Let C be a set and let E and G be two sets such that a subtraction $- : E \times G \to E$ is defined. Let $a \in G$. Then, the **translation operator** is defined as

$$\begin{array}{rccl} T_a : & C^E & \to & C^E \\ & f & \mapsto & T_a[f] : \begin{array}{ccc} E & \to & C \\ x & \mapsto & f(x-a) \end{array} \end{array} . \tag{B.3}$$

Definition B.3 (Reflection operator)**.** Let C be a set and $(G, +)$ be a group. Then, the **reflection operator** is defined as

$$\begin{array}{rccl} \mathcal{R} : & C^G & \to & C^G \\ & f & \mapsto & \mathcal{R}[f] : \begin{array}{ccc} G & \to & C \\ x & \mapsto & f(-x) \end{array} \end{array} . \tag{B.4}$$

When the set C has a ring structure, we can define the addition and multiplication operators.

Definition B.4 (Addition and multiplication operators)**.** Let D be a set. Let $(R, +, \cdot)$ a ring. Then, the **addition operator** is defined as

$$\begin{array}{rccl} + : & (R^D, R^D) & \to & R^D \\ & (f, g) & \mapsto & f + g : \begin{array}{ccc} D & \to & R \\ x & \mapsto & f(x) + g(x) \end{array} \end{array} . \tag{B.5}$$

The **multiplication operator** is defined as

$$\begin{array}{rccl} \cdot : & (R^D, R^D) & \to & R^D \\ & (f, g) & \mapsto & f \cdot g : \begin{array}{ccc} D & \to & R \\ x & \mapsto & f(x) \cdot g(x) \end{array} \end{array} . \tag{B.6}$$

When we have a field structure in the domain, we can define the scaling operator.

Definition B.5 (Scaling operator)**.** Let $(\mathbb{K}, +, \cdot)$ be a field and C a set. Let $s \in \mathbb{K} \setminus \{0\}$. Then, the **scaling operator** is defined as

$$\begin{array}{rccl} D_s : & C^{\mathbb{K}} & \to & C^{\mathbb{K}} \\ & f & \mapsto & D_s[f] : \begin{array}{ccc} \mathbb{K} & \to & C \\ x & \mapsto & f\left(\frac{x}{s}\right) \end{array} \end{array} . \tag{B.7}$$

The scaling operator may be defined differently depending on what norm you want to preserve.

Definition B.6 (p-unitary scaling operator)**.** Let $s \in \mathbb{K} \setminus \{0\}$. Let $p \in [1, \infty]$. We define the **p-unitary scaling operator** of scale s as

$$\begin{array}{rccl} D_s^p : & \mathbb{C}^{\mathbb{R}} & \to & \mathbb{C}^{\mathbb{R}} \\ & f & \mapsto & D_s^p[f] : \begin{array}{ccc} \mathbb{R} & \to & \mathbb{C} \\ x & \mapsto & |s|^{-\frac{1}{p}} f\left(\frac{x}{s}\right) \end{array} \end{array} \tag{B.8}$$

where we assume $\frac{1}{\infty} = 0$.

Proposition B.7. *The p-unitary scaling operator is unitary for the p norm, i.e.,*

$$||D_s^p[f]||_p = ||f||_p \,. \tag{B.9}$$

When the codomain is $\mathbb{C}$, we can define the conjugate operator.

Definition B.8 (Conjugate operator)**.** Let D be a set. Then, the **conjugate operator** is defined as

$$\begin{array}{rccl} \bar{\cdot} : & \mathbb{C}^D & \to & \mathbb{C}^D \\ & f & \mapsto & \overline{f} : \begin{array}{ccc} D & \to & \mathbb{C} \\ x & \mapsto & \overline{f(x)} \end{array} \end{array} \,. \tag{B.10}$$

If the domain is a group, we can define the involution operator.

Definition B.9. Let $(G, +)$ be a group. The **involution operator** is defined as

$$\begin{array}{rccl} \cdot^* : & \mathbb{C}^G & \to & \mathbb{C}^G \\ & f & \mapsto & f^* : \begin{array}{ccc} G & \to & \mathbb{C} \\ x & \mapsto & \overline{f(-x)} \end{array} \end{array} \,. \tag{B.11}$$

When the domain is $\mathbb{R}$ and the codomain is $\mathbb{C}$, we can define the modulation operator.

Definition B.10 (Modulation operator)**.** Let $\xi \in \mathbb{R}$. Then, the **modulation operator** is defined as

$$\begin{array}{rccl} M_\xi : & \mathbb{C}^{\mathbb{R}} & \to & \mathbb{C}^{\mathbb{R}} \\ & f & \mapsto & M_\xi[f] : \begin{array}{ccc} \mathbb{R} & \to & \mathbb{C} \\ t & \mapsto & f(t) \cdot e^{2\pi \imath t \xi} \end{array} \end{array} \,. \tag{B.12}$$

Proposition B.11. *Let $\tau \in \mathbb{R}$. Let $\xi \in \mathbb{R}$. Then, $\forall f \in \mathbb{C}^{\mathbb{R}}$,*

$$T_\tau M_\xi = e^{-2\pi\imath\tau\xi} M_\xi T_\tau \qquad\qquad M_\xi T_\tau = e^{2\pi\imath\tau\xi} T_\tau M_\xi \,.$$

Another important operator is the convolution.

Definition B.12 (Convolution operator)**.** The **convolution operator** is defined as

$$\begin{array}{rccl} * : & \mathcal{L}^1(\mathbb{R};\mathbb{C}) \times \mathcal{L}^p(\mathbb{R};\mathbb{C}) & \to & \mathcal{L}^p(\mathbb{R};\mathbb{C}) \\ & (f,g) & \mapsto & f*g : \begin{array}{ccc} \mathbb{R} & \to & \mathbb{C} \\ t & \mapsto & \int_{\mathbb{R}} f(u)\cdot g(t-u)\,du \end{array} \end{array} . \tag{B.13}$$

When the functions belong to $\mathcal{L}_2(\mathbb{R};\mathbb{C})$, we can define the scalar product of two functions.

Definition B.13 (Scalar product)**.** Let $f, g \in \mathcal{L}_2(\mathbb{R};\mathbb{C})$. Then, the **scalar product** between f and g is defined as

$$\langle f, g\rangle = \int_{\mathbb{R}} f(x)\cdot\overline{g(x)}\,dx\,. \tag{B.14}$$

The convolution can be expressed in terms of the scalar product.

Proposition B.14. *Let $f, g \in \mathcal{L}_2(\mathbb{R};\mathbb{C})$. Then, $\forall x \in \mathbb{R}$,*

$$(f*g)(x) = \langle f, T_x g^*\rangle\,. \tag{B.15}$$

B.3 Fourier theory

In this section we will recall the basics of Fourier transformations, in particular to establish the notations used throughout this thesis.

The *Fourier Transform* is defined for functions in $\mathcal{L}_1(\mathbb{R};\mathbb{C})$ as follows.

Definition B.15. Let $f \in \mathcal{L}_1(\mathbb{R};\mathbb{C})$. Then, the **Fourier Transform** of f is defined by

$$\begin{array}{rccc} \hat{f} : & \mathbb{R} & \to & \mathbb{C} \\ & \xi & \mapsto & \int_{\mathbb{R}} f(t) e^{2\pi\imath t\xi}\,dt \end{array} . \tag{B.16}$$

The **Fourier Transform** is the linear operator $\mathscr{F}$ defined by

$$\begin{array}{rccc} \mathscr{F} : & \mathcal{L}_1(\mathbb{R};\mathbb{C}) & \to & \mathcal{L}_\infty(\mathbb{R};\mathbb{C}) \\ & f & \mapsto & \mathscr{F}[f] = \hat{f} \end{array} . \tag{B.17}$$

Proposition B.16. *The Fourier transform is an unitary operator in* $\mathcal{L}_2(\mathbb{R};\mathbb{C})$*, i.e.,*

$$\begin{array}{rcccccc} \mathscr{F}: & \mathcal{L}_2(\mathbb{R};\mathbb{C}) & \to & \mathcal{L}_2(\mathbb{R};\mathbb{C}) & & & \\ & f & \mapsto & \mathscr{F}[f]: & \mathbb{R} & \to & \mathbb{C} \\ & & & & \xi & \mapsto & \langle f, M_\xi[\mathbf{1}]\rangle \end{array} \tag{B.18}$$

where $\mathbf{1}$ *is the function* $\mathbf{1}:\mathbb{R}\to\mathbb{C}$, $t\mapsto 1$*, and*

$$||\mathscr{F}[f]||_2 = ||f||_2\,. \tag{B.19}$$