Dynamical and topological tools for (modern) music analysis

Mattia Giuseppe Bergomi

UNIVERSITÀ DEGLI STUDI DI MILANO

UPMC SORBONNE UNIVERSITÉS

ircam Centre Pompidou

Tesi di dottorato in cotutela tra l'Università degli Studi di Milano
e l'Université Pierre et Marie Curie

Thèse de doctorat en cotutelle entre l'Università degli Studi di Milano
et l'Université Pierre et Marie Curie

Programma in Informatica
Scuola di Dottorato in Informatica, Università degli Studi di Milano

Specialité
Mathématiques
École doctorale Informatique, Télécommunications et Électronique (Paris)

Presentata da / *Présentée par*
Mattia Giuseppe Bergomi

Per il conseguimento del titolo di
Dottorato di ricerca dell'Università di Milano

Pour obtenir le grade de
Docteur de l'Université Pierre et Marie Curie

Titolo della tesi / *Sujet de tèse*

Dynamical and Topological Tools for (Modern) Music Analysis

Discussa il 10 dicembre 2015/ *Soutenue le 10 décembre 2015*

di fronte alla commissione composta da / *devant le jury composé de*

Goffredo Haus	Direttore di tesi	*Directeur de thèse*
Moreno Andreatta	Codirettore di tesi	*Codirecteur de thèse*
Davide Luigi Ferrario	Referee	*Rapporteur*
Elaine Chew	Referee	*Rapporteur*
Massimo Ferri	Esaminatore	*Examinateur*
Jean-Louis Giavitto	Esaminatore	*Examinateur*

Dynamical and Topological

Tools for

(Modern) Music Analysis

Mattia G. Bergomi

2015

I read in a book that the objectivity of human thought can be expressed by using the verb to think *in impersonal form. Could we ever say* "it plays" *as we say* "it rains"*, or* "today it is windy"*? [...] And may we also say* "it listens" *as we say* "it rains"*?*

— *Freely translated and adapted by the author from* Se una notte d'inverno un viaggiatore*, Italo Calvino.*

Abstract

Is it possible to represent the horizontal motions of the melodic strands of a contrapuntal composition, or the main ideas of a jazz standard as mathematical entities? In this work, we suggest a collection of novel models for the representation of music that are endowed with two main features. First, they originate from a topological and geometrical inspiration; second, their low dimensionality allows to build simple and informative visualisations.

Here, we tackle the problem of music representation following three non-orthogonal directions. We suggest a formalisation of the concept of voice leading (the assignment of an instrument to each voice in a sequence of chords) suggesting a horizontal viewpoint on music, constituted by the simultaneous motions of superposed melodies. This formalisation naturally leads to the interpretation of counterpoint as a multivariate time series of partial permutation matrices, whose observations are characterised by a degree of complexity. After providing both a static and a dynamic representation of counterpoint, voice leadings are reinterpreted as a special class of partial singular braids (paths in the Euclidean space), and their main features are visualised as geometric configurations of collections of 3-dimensional strands.

Thereafter, we neglect this time-related information, in order to reduce the problem to the study of *vertical* musical entities. The model we propose is derived from a topological interpretation of the *Tonnetz* (a graph commonly used in computational musicology) and the deformation of its vertices induced by a harmonic and a consonance-oriented function, respectively. The 3-dimensional shapes derived from these deformations are classified using the formalism of persistent homology. This powerful topological technique allows to compute a *fingerprint* of a shape, that reflects its *persistent* geometrical and topological properties. Furthermore, it is possible to compute a distance between these fingerprints and hence study their hierarchical organisation. This particular feature allows us to tackle the problem of automatic classification of music in an innovative way. Thus, this novel representation of music is evaluated on a collection of heterogenous musical datasets.

Finally, a combination of the two aforementioned approaches is proposed. A model at the crossroad between the signal and symbolic analysis of music uses multiple sequences alignment to provide an encompassing, novel viewpoint on the musical inspiration transfer among compositions belonging to different artists, genres and time. To conclude, we shall represent music as a time series of topological fingerprints, whose metric nature allows to compare pairs of time-varying shapes in both topological and in musical terms. In particular the dissimilarity scores computed by aligning such sequences shall be applied both to the analysis and classification of music.

Contents

Introduction

Modelling a creative process is a daunting task, since it is not yet possible to define an operator capable of judging its aesthetics in an objective way. This is one of the main reasons that renders the realisation of formal models for the analysis and classification of music such a challenging endeavour. It is necessary to investigate the compositional process, in order to provide a coherent analysis and a robust classifier of music.

Often, the core of a piece of music is made of a small collection of strong, recognisable musical concepts, that are grasped by the majority of the listeners (Dowling, 1972; Folgieri et al., 2014; Tulipano and Bergomi, 2015). These *musical concepts* are shaped by varying levels of tension over time, drawing the attention of the listener to particular moments thanks to specific choices, frustrating his or her intuition through unexpected changes, or confirming his or her expectation with, for instance, a well-known cadence leading to resolution.

Our approach to the analysis of music composition stems from the assumption that it is based on two main actions used by the composer to describe musical concepts and shape his or her piece. The philosopher and musicologist Ernst Kurth in *Grundlagen des Linearen Kontrapunkts* (Kurth and Rothfarb, 1991) describes the counterpoint as *an equilibrium among streaming linear forces (kinetic energy) and congealing harmonics forces (potential energy).* These terms, that are not meant to be interpreted as scientific definitions, suggest a twofold interpretation of music. On one side, the horizontal point of view, intended as the behaviour of superposed independent melodic *strands* of counterpoint; on the other side, a vertical perspective where music is compressed in a harmonic form, and chords summarise the information otherwise ordered in time.

From a scientific viewpoint, the analysis of music has largely been attacked on the symbolical side with algebraic tools (Andreatta, 2003; Zabka, 2009) and the category theory (Mazzola and Andreatta, 2006; Mazzola et al., 2002; Popoff et al., 2015), while its audio signals have been largely explored in computer science, leading to the field of *Music Information Retrieval* (Casey et al., 2008). Recently, the mathematical community witnessed a surprising growth of the field of *Topological Persistence* (d'Amico et al., 2006; Edelsbrunner and Harer, 2008). This theory provides a rigorous approach to the problem of shapes recognition, allowing to compare complex forms, while giving a simple and robust representation of their geometrical and topological properties. As the models for the analysis of audio signals take advantage of the strategies developed for image analysis (Smaragdis and Brown, 2003; Wang et al., 2003; Li et al., 2010), it is possible to borrow some tools from the topological analysis of shapes and data to tackle the problem of music

analysis.

The main aim of this work lies in the introduction of low-dimensional topological and geometrical models in order to describe, albeit in a extremely simplified form, the compositional process. This task has been split into two smaller problems, following the approach described by Kurth. On one side, the analysis of multidimensional time series as a concatenation of events in time, which finds its natural musical counterpart in the voice-leading theory. On the other side, the representation of *persistent* features of static and time-varying shapes, encoding in their geometry the information carried by the symbolic and signal-based nature of music.

The structure of this work reflects these considerations. After an introductory part, aimed at defining some basic musical and mathematical concepts, it is developed in three parts. In Parts II and III, these horizontal and vertical approaches are described separately, although they are far from being orthogonal. Consequently, two strategies considering both these aspects are proposed in Part IV. In the following paragraphs the main contributions of each parts are described.

Musical and mathematical preliminaries

In this first part, we introduce the main musical and mathematical characters that shall intervene in this work. First, a quick historical presentation of the concepts of monody and polyphony is presented and the links between these classical compositional techniques and their modern counterparts are discussed. Second, we place our research at the crossroad between mathematics and music. Two state-of-the-art approaches that inspired our investigation are discussed. On one hand, we introduce the geometrical representation of harmonic objects provided by the chords space (Tymoczko, 2011; Callender et al., 2008), together with the interpretation of voice leadings as trajectories in this space, which inspired the research described in Part II. On the other hand, we introduce the notion of simplex and simplicial complex, two standard objects in algebraic topology, that will be used to provide a topological definition of the *Tonnetz*.

Algebraic and geometric models for the voice leading theory

Given a sequence of chords, the voice leading process corresponds to their transformation in a superposition of melodies, endowed with a certain degree of independence. The main contribution of this part is the introduction of a mathematical formalisation of the concept of voice leading and the representation of simultaneous motions of voices as partial permutations. An algorithm to univocally compute the partial permutation matrix associated to a voice leading is proposed. In particular, we demonstrate how this simple representation suffices to describe the behaviour of the voices, with a focus on voice crossing.

As a geometrical counterpart to this first algebraic interpretation, voice leadings are interpreted as piecewise geodesics in several spaces. The different types of voice motions are analysed in each space, pointing out how minimal geodesic paths represent non-crossing voice leadings among two chords. Then, consecutive simultaneous motions of voices are represented as concatenation of geodesics.

Once the essential role played by the juxtaposition of voice leadings as a concatenation of linear function has been modelled, the concatenations of geodesics

are substituted by concatenations of partial permutation matrices. Associating to each permutation matrix a 4-dimensional complexity vector describing the main features of the voice leading, we provide a representation of the counterpoint of the first species. After generalising this model to the study of the concatenation of voice leadings containing rests, a naïve extension to other contrapuntal species is suggested.

Finally, in order to provide a visualisation of voices motions, voice leadings are described as trajectories in 3-dimensional Euclidean space by using the mapping which is naturally defined between partial braids and partial permutations. This representation shall prove to be very efficient for the visualisation of voice leadings between n-notes chords, when a particular class of trajectories is considered.

Persistent musical features

In this section, the ordering of melodic or harmonic states that represented the core of the previous part is neglected. Music is seen as composed by vertical, unordered entities, as a pianist could interpret a scale as a cluster, to grasp at first sight its intervallic properties.

The main idea is to introduce a metric representation of the *Tonnetz* interpreted as a planar polyhedral surface, whose vertices are displayed along a third dimension, through a specific function. In particular, we shall consider two deformation functions. The first one is defined in the symbolic domain and takes into account the pitch classes and durations of a series of notes. The second, based on the interaction between signal and symbol, is constructed on the consonance function as it has been defined by (Plomp and Levelt, 1965).

The shapes obtained via these deformations are then classified by computing their *persistent homology*. The novel and lively field of computational topology provides a series of tools allowing to associate a fingerprint to a shape, describing its geometrical and topological features as a simple diagram. After a preliminary section introducing the basic definitions and theorems of persistent homology, this formalism is applied to the analysis of music. The results are interpreted in a musical context. Moreover, the distance between persistence diagrams is used to classify several datasets of compositions, modal scales and triads.

Harmonic time series and persistence snapshots

The dynamic and time-dependent nature of music is one of the main ingredients of this last part. In the first chapter, we suggest a novel approach to the analysis of pop music. At the intersection of the symbolic and signal-analysis domains, this method consists in the interpretation of automatically transcribed harmonic progressions as symbolic sequences. Such sequences shall be analysed by computing their multiple alignment. Widely used in phylogenetic, this technique shall provide an encompassing representation of the harmonic features characterising a dataset of 138 Pop songs. The analysis of statistically recurrent motifs of these aligned sequences allows to quantify and analyse the shared inspiration and the contamination over time among compositions.

Thereafter, we propose an adaptation of the model introduced in the previous section, in order to include time in the geometrical and topological analysis of

music. Static shapes are now considered as time-varying systems, whose evolution is describable as a sequence of observations in time. Thus, we shall consider the time series formed by topological fingerprints computed on a sampling of the *Tonnetz*'s deformation in time. A musical interpretation of the meaning of these topology-based time series is followed by an application of this technique to a music classification task on three different datasets.

Part I

Musical and mathematical preliminaries

Table of Contents

Abstract

The aim of this first section is to introduce the ingredients of music theory that inspired our research, in order to provide a practical musical setting for the whole work and to give some important music-oriented bibliographic references. In Chapter 1, a brief history of monody, polyphony, counterpoint and its relationship to western common practice tradition is discussed.

In Section 2.1 we introduce the concept of *simplicial complex*, a core object in algebraic topology representing one of the main mathematical ingredients of this work. Then, we place our investigation in the mathematical/musical domain: we introduce the chord space, which inspired a model describing the complexity of voice leading presented in Part II and the *Tonnetz*, that will be used in Part III.

One

Music theory preliminaries

Monody and *polyphony* allow to introduce two apparently orthogonal approaches to music analysis. The former suggests the well-known interpretation of chords as unordered sets of notes (referred hereafter as a *vertical* analysis). The latter, can be defined as the study of voices moving independently as a superposition of melodies (referred hereafter as *horizontal* analysis). Although both approaches encode relevant information, we shall observe that it is not possible, even on a historical basis, to order these approaches chronologically, nor to define them as independent techniques.

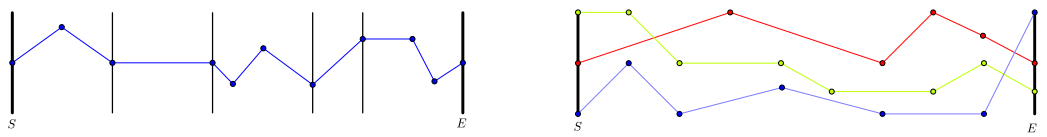
In Figure 1.1 an intuitive representation of these viewpoints is depicted. Monody can be depicted as a set of horizontal lines in simultaneous motions, while polyphonic music can be represented as a series of independent lines in terms of height (pitches) and time (duration). The superposition of several melodies allows the composer to enrich and emphasise a main melody, which is preferred among the others.

Shortly, we shall provide a quick historical overview on monophony and polyphony. This section aims at supplying the reader with the basic information concerning what shall be developed in the next parts of this work, to provide the essential music theory bibliographic references and some examples linking the classical concepts of monody and polyphony with *modern music*.

1.1 Monody, polyphony and modern notation

1.1.1 Monody and lead sheet

In the fourth century, when the first monastic communities were created, the *psalmodic practices* arose as ancestors of the Gregorian chant (Apel, 1958; Chanan, 1994). The monophonic monastic psalmody was used as a metaphor of discipline, to



(a) Monody. (b) Polyphony.

Figure 1.1: Intuitive representation of monody and polyphony. (a) Monody is intended as a melodic line supported by a harmonic progression. (b) The polyphonic approach allows to create superpositions of independent melodic strands, that affect the listener both as a whole and separated entities.

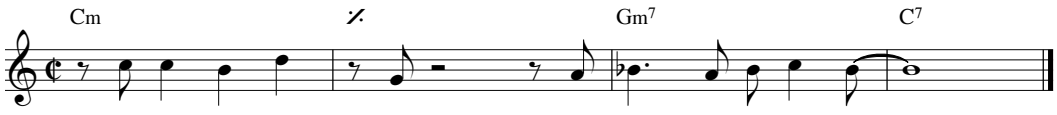


Figure 1.2: An example of lead sheet.

create and enforce the bond among the members of the community, as it is described in (Basil, 1963).

It is important to note that, in the context described above, monophony represents a specific choice, rather than an ancestor of polyphonic music, representing the rejection of earlier —presumably polyphonic— practices. Indeed, polyphony has never supplanted monophony in the history of western music. If the term monophony is used to describe music consisting of a single (generally vocal) melodic part, *monophony* is a melody sustained by a harmonic progression. This term was introduced in (Galilei, 1569), in order to describe a single voice supported by the chords played by a lute. We refer interested reader to (Taruskin, 2009, Chapter 1) for further details about the passage from monastic psalmody to monophony.

In a modern music context, the idea of monody and its notation are widely used. It is common practice to use the lead sheet notation to represent music in a concise form, as it is depicted in Figure 1.2. The melody is written in standard notation, while chords appear above the staff as symbols (see Appendix E for an introduction to chord notation). This kind of harmonic notation provides no information concerning the voicings that should be used and both the rhythmical and dynamical aspects are also neglected.

In lead sheet notation, chords are represented as *vertical* structures. In this case, it is natural to think about them, as pitch-class sets, i. e. collections of notes in which neither the octave, nor the order of the notes composing the chord are specified. The mathematical model describing this construction will be detailed in Chapter 2. However, it is clear that the style and the time in which a song has been composed, arranged or re-arranged, lead the performer to certain musical choices, that at least partially, fill the notation's gaps. This vertical approach to music inspired the model, that we will describe in Part III.

1.1.2 Polyphony, modal harmony and melodic voicings

Polyphony has always been present in European music. However, we can identify the 12^{th} century as the moment in which polyphonic composition became the standard technique in Western music. As we claimed above, polyphony and monophony are not terms in opposition, but answers to different needs.

The practice of polyphony was firstly described in the treatise *Musica Enchiriadis* and its contemporary commentary *Schola Enchiriadis*, see (Erickson and Palisca, 1995). These treatises depict two polyphonic techniques that can be used to enrich a given melody. It is interesting to note how these two techniques can be reinterpreted in a modern context.

The first one is the *ison* chanting, in which the tonic note of the melody, explicitly notated, is supposed to be held while the main melody is sung. In a modern music

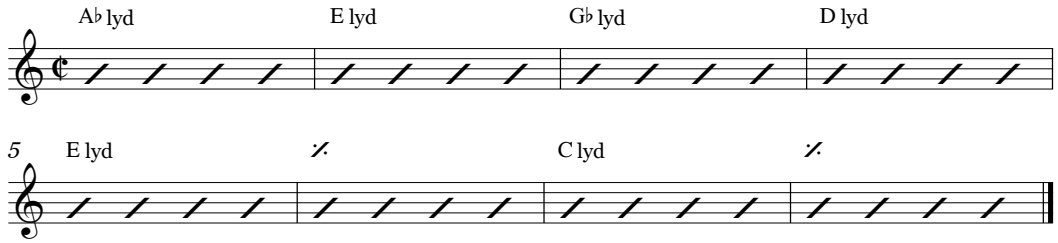


Figure 1.3: Chord symbols are substituted by mode names. *The Law of Diminishing Returns - Alan Pasqua. Solos part B.*

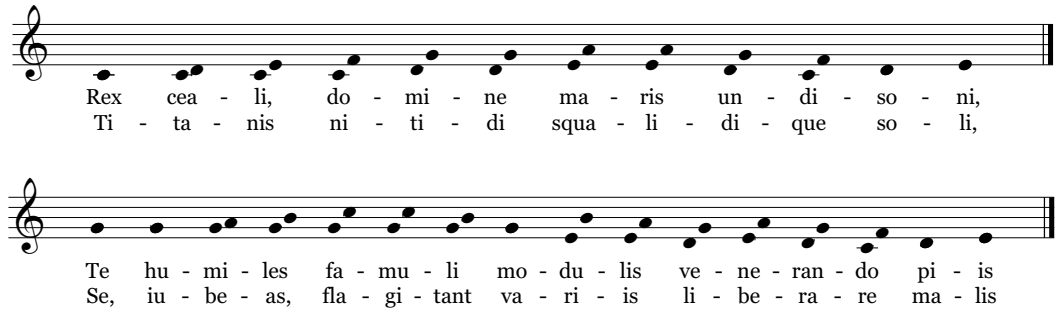


Figure 1.4: Example of polyphony from Musica Enchiriadis. Transcription from (Taruskin, 2009, Chapter 2).

context, this type of practice is analogous to the modal harmony notation. For example, consider the notation used in *The Law of Diminishing Returns*' solo section depicted in Figure 1.3. In this case, the notation specifies a particular modal choice (lydian) and its *ison*, i. e. the root of the modal scale and the reference pitch that allows to identify the lydian mode. See Appendix A.1 for an introduction to modal theory.

The second technique describes the harmonisation of a given melody through parallel doubling, i. e. the accompaniment of a melody with another one consisting of its transposition to a fixed consonant interval. The modern analogous to this technique is the arrangement process known as the *voicing of a melody* or *block voicing*. This practice lies at the intersection between monody and polyphony. Given a lead sheet, the melody can be *voiced* using its harmonic structure (see Figure 1.5 for an example). We refer to (Wei, 2008) for a detailed explanation of this technique in its modern version and to (Taruskin, 2009, Chapter 5) for details on its classical use. It is possible to find something similar in the two partitions of Figure 1.4. They are not examples of polyphonic composition, but a reinforcement of the *vox principalis* through a lower melody, *organum*, producing an intuitive contrapuntal harmony.

As the enrichment of a melody using voicings is strictly related to a chord the two techniques described in *Musica Enchiriadis* are far from the compositional independence that characterises a polyphonic composition. Two important innovations are described in the *Micrologus* (d'Arezzo et al., 1993). First, as it is depicted in Figure 1.6, more than one contrapuntal solution can be given as harmonisation of the same melody. In particular, in Figure 1.6b the *final* is reached with a passing

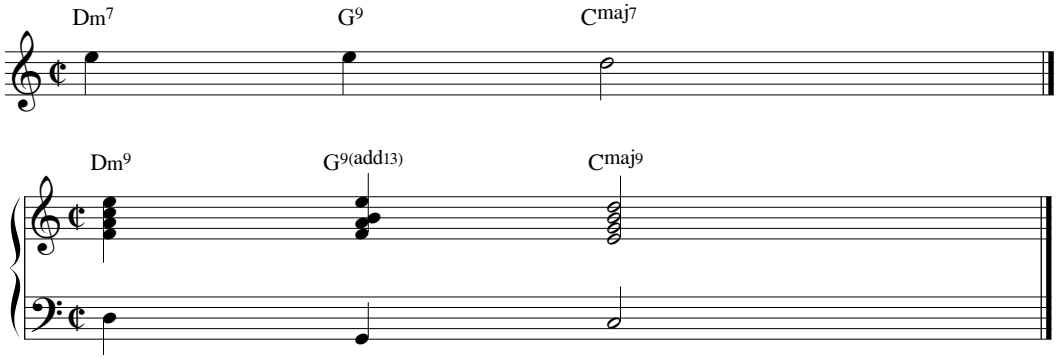
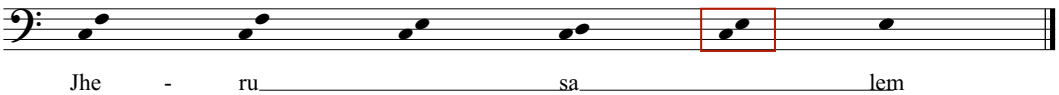
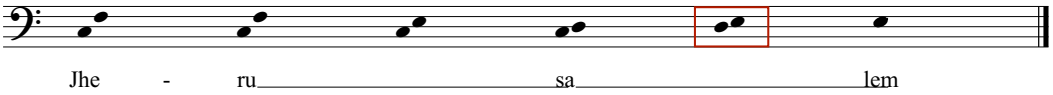


Figure 1.5: Melody voicing.



(a) Solution 1.



(b) Solution 2.

Figure 1.6: Two different harmonizations of Jerusalem. Guido d'Arezzo, *Micrologus*.

note and the major second is used as a secondary consonance, giving rise to a smoother passage to the final than the direct leap used in Figure 1.6a. Second, although some intervals like the perfect fifth are still judged as *hard-sounding*, in *Micrologus* the contrapuntal technique is based on the pleasantness of a certain harmonic choice, rather than on a *natural law*. Thus, not only the process of voicing has been brought to a more human level, but even the concept of parsimony of voice leading is introduced, as one may notice from the movement of the organum in both the examples of Figure 1.6.

At the same time, the examples given in the *Micrologus* stress a preference for contrary motion at cadences, while the parallel doubling represents a sporadical choice, thanks to the new degrees of freedom the voices are endowed with. See also (Rankin, 1993) for *pre-guidonian* evidences on the use of parsimonious voice leadings and contrary motion. In Figure 1.7 it is possible to observe an example of this relative independence of voices.

This independence has been inherited by modern music, representing the typical behaviour of the melody against a bass line. The former moving more or less freely depending on the context, and the latter linked to the harmonic choices made by the composer. In Figure 1.8 the bars 1-4 of *Interplay* by Bill Evans are depicted. The harmonic progression is stressed by the movement of the bass line and enriched with a higher voice, in a harmonisation of major and minor twelfths. This choice states both a tonal ($B\flat$ minor) and modal (F phrygian) choice. The melody moves with a high degree of independence, often in contrary motion and crossing the *tenor*

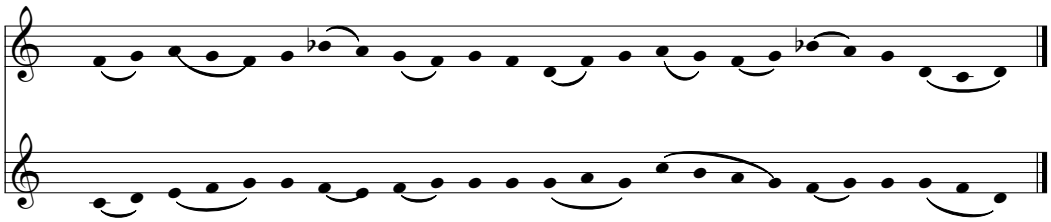


Figure 1.7: Independent voice leading and contrary motion. A fragment of *Alleluia: Angelus Domini - Chartres 109, fol. 75.*

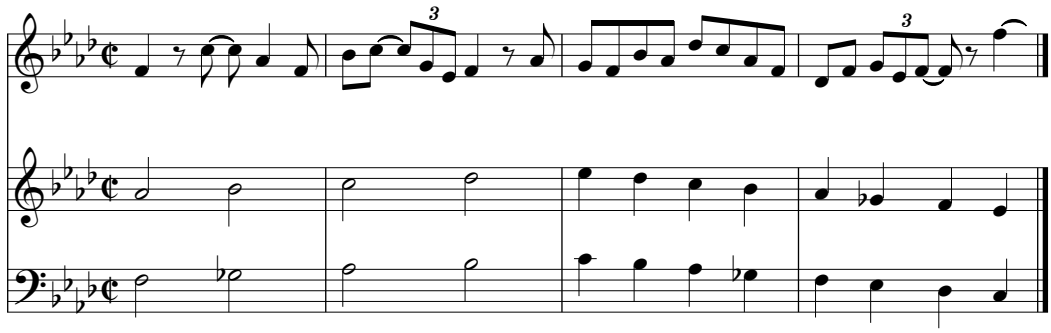


Figure 1.8: Polyphonic Jazz standard. Segregation among the melody and the bass voices. *Interplay*, Bill Evans, bars 1-4.

voice. Voice leading techniques shall be detailed later in Section 1.2. However the degree of independence of the voices either in rhythmical or intervallic terms, allows to classify counterpoint into five species, depicted in Figure 1.9. To conclude this comparison among classical and modern monodic and polyphonic techniques, we show how the first and fifth contrapuntal species has been used in two jazz composition in Figures 1.10 and 1.11, respectively.

The necessity of a representation of simultaneous motion of voices and its visualisation inspired the work we describe in Part II. The time-dependent nature of Music, suggested the time-series-oriented representation of Music, will be described in Part IV.

1.2 Voice leading practice

Harmony and the study of counterpoint provide some theoretical axioms to guarantee the smoothness of a composition (where smoothness is intended in this context as understandability). We refer to (Aldwell et al., 2010, Chapter 6) for a list of phenomena occurring in the voice leading process in four-part writing. The following list aims at describing some compositional strategies, that shall be used in the remainder of this work.

Vocal range. Each voice has to be settled in a range that can be sung without excessive effort. Thus the construction of a melody associated to each voice has to take into account this particular feature:

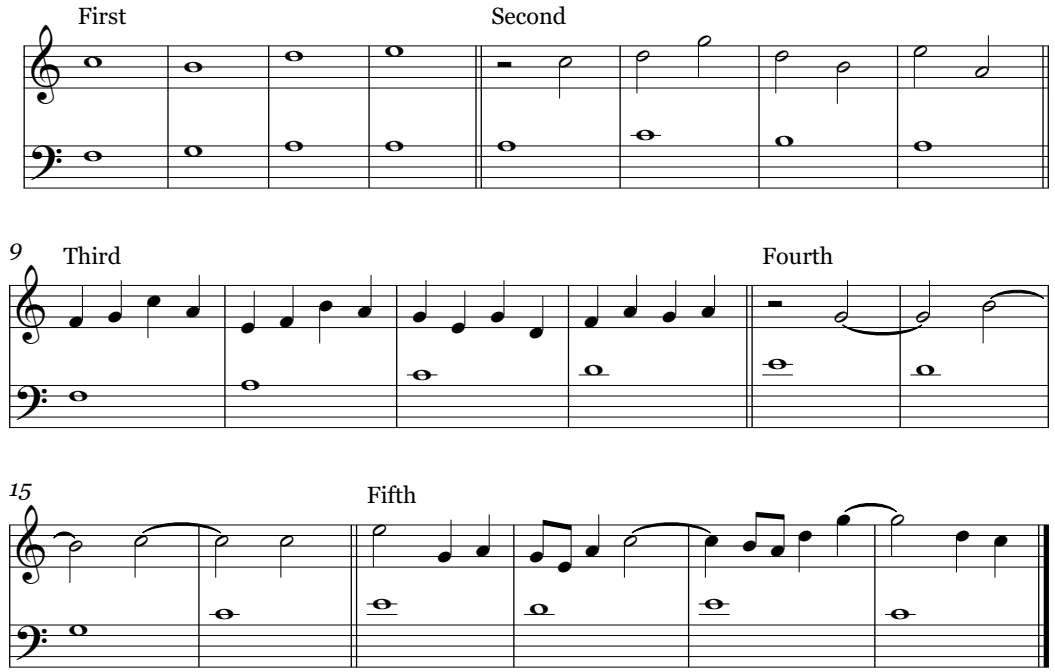


Figure 1.9: Five different degree of independence among voices. From note against note in the first specie of counterpoint, to the complete degree of independence of the fifth specie.

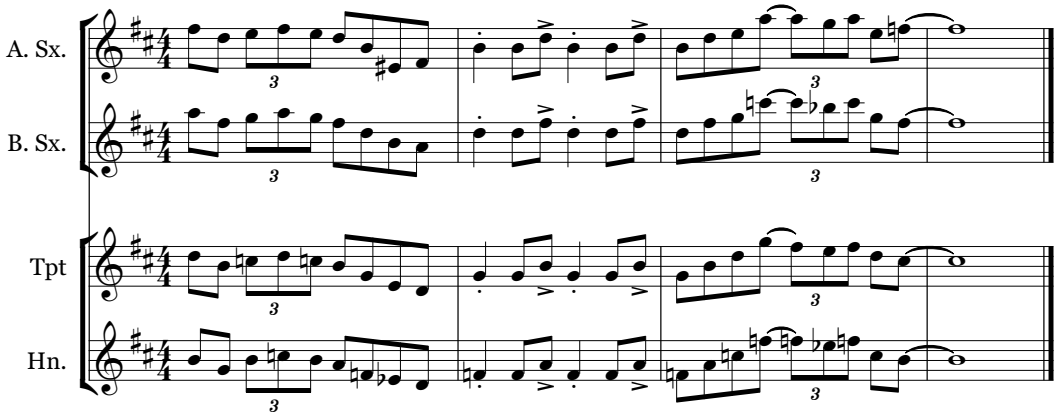


Figure 1.10: A reduced orchestration of *Boplicity bars* 1-4. *Birth of the Cool*, by Miles Davis.

i) Soprano : $C_4 \to G_6$,

ii) Alto : $G_3 \to C_5$,

iii) Tenor : $C_3 \to G_4$,

iv) Bass : $E_2 \to C_4$.

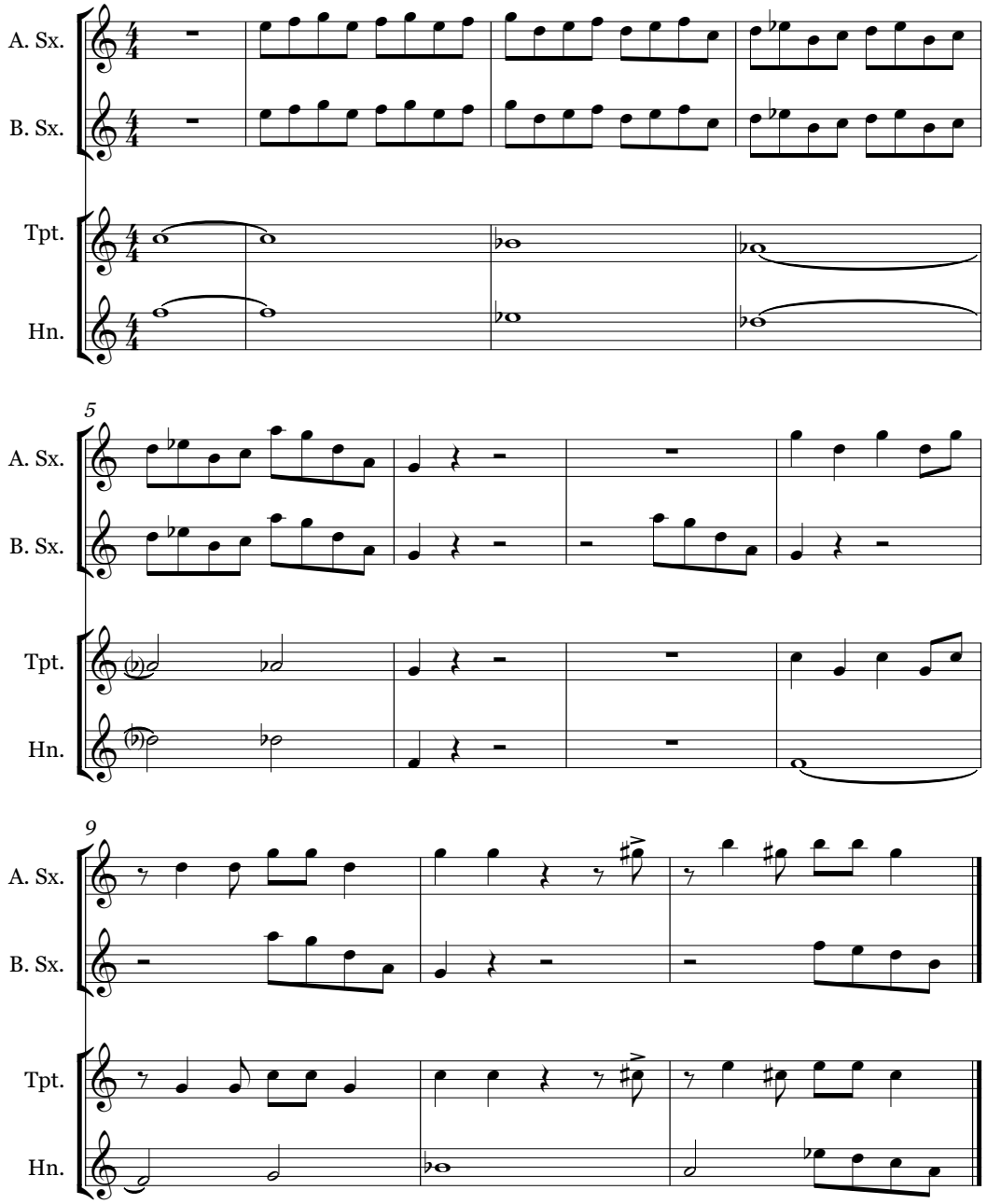


Figure 1.11: Alto sax, baritone sax, trumpet and horn voices in *Move*, bars 1-11, by Miles Davis.

Doubling. We assume that the only absolute rule to augment the complexity of a voice leading is the doubling of a tension of a chord. The fifth can be omitted in root-position chords, since they do not add information concerning the *genre* of the chord. Thus, in seventh-chords the doubled root could take the place of the fifth, while the doubling of a seventh has to be considered as dissonant. The third can only be omitted to achieve special effects.

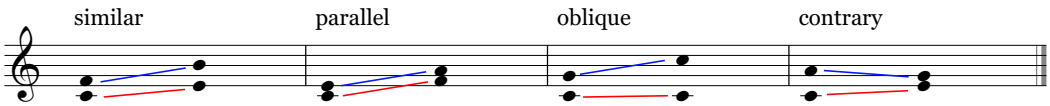


Figure 1.12: Motion classes for two voices. *Similar:* same direction but different intervals; *parallel:* same direction and same intervals; *oblique:* only one voice is moving; *contrary:* opposite directions.

Microscopic spacing rules. A wide spacing among the upper voices can create an effect of thinness mostly if it is continued for two or three chords. Normally, adjacent upper voices should not be more than an octave apart, while even a two octaves separation is acceptable among the tenor and the bass voices. Furthermore a soprano voice segregated from the other voices or the excessive proximity of the tenor and the alto voices can create confusion.

Voice crossing. Crossing occurs when two voices exchange positions. It is less problematic when it involves inner voices and for a small amount of time.

Overlap. Overlap occurs when a voice moves above or under the former state of an adjacent voice. Thus the difference between voice crossing and overlap is that in the latter the relative positions of the voices are maintained, but their ranges intersect in two consecutive moments. This practice can lead to confusing voice leadings.

Leap. The degree of complexity of a leap depends on its intervallic size and its consequent consonant or dissonant nature. Here follows a simple classification:

- Minor and major third: consonant leaps.
- Sixth or seventh: dissonant leaps, usually followed by a change of motion.
- Larger than an octave: not permitted, rarely used to create interest.
- Perfect fourth and perfect fifth: consonant and often followed by a motion change.

Two consecutive leaps in the same direction are usually avoided, with the exception of two consecutive third leaps.

Melodic motion. Generally, the soprano line tends to move by conjunct motion avoiding leaps. The bass line is normally in charge to support the other voices, clarifying the harmony of the piece, thus it can move disjointedly. Inner voices have to complete the tones of the chord framed by the bass and soprano lines. In conclusion, leaps of the soprano voice increase the complexity of a voice leading, but their complete absence would create repetitive and static melodies and would make the harmonic structure vague.

Simultaneous motion. It is possible to classify the simultaneous motion of two voices as follows (refer to Figure 1.12 for an intuitive representation):

- Similar Motion: same direction, different spacing.
- Parallel Motion: same direction and same spacing.
- Oblique Motion: only one voice is moving.
- Contrary Motion: opposite directions.

Contrary motion provides contrast and independence to the voices, creating an interesting soundscape for the listener. Parallel motion in thirds, sixths and tenths can be considered among the most powerful voice leading techniques. In some cases, parallel motion bounds the possible configurations of the voices, thus it is forbidden for unisons, octaves, fifths.

Consecutive fifths and octaves by contrary motion are normally avoided. Hidden fifths and octaves are to be avoided in few voices contexts (forbidden in two parts writing). A complex texture or a dissonant context mitigate the effect of parallel fifths and octaves. The general rule holds, hidden octaves have to be avoided in the outer voices.

Two

Mathematical models: state of the art

In this section we present two important music representation models. First, the chord space which has the interesting mathematical structure of an orbifold. This space has been recently introduced in (Tymoczko, 2006) and it is characterised by a metric, continuous structure. Second, in a sort of *mathematical opposition* to this model, we describe the *Tonnetz*. It was represented, at its origin, as a table (Euler, 1739b), aiming at stressing the acoustic relationships among pitches. It has been described as an abstract graph in (Zabka, 2009). We shall suggest a topological representation of the *Tonnetz*. In order to safely define these music representation spaces, we shall introduce two basic concepts of algebraic topology: simplices and simplicial complexes.

2.1 Simplicial complexes

2.1.1 Simplices

A standard object in Topology is the *gluing diagram*: a collection of topological polygons, whose edges are labeled and oriented. Such a diagram represents the space obtained by gluing the sides labeled with the same letters, and matching orientations. Geometrical entities like the torus $\mathbb{T}^2$, the Möbius strip M, the projective plane $\mathbb{R}\mathrm{P}^2$ and the Klein bottle K can be obtained by attaching two triangles as it is depicted in Figure 2.1.

Simplices generalise this idea to higher dimensions: it is possible to think about the n-dimensional simplex as an equivalent of the n-dimensional triangle.

Let $V = \{ v_0, v_1, \ldots, v_n \}$ be a set of points in $\mathbb{R}^m$. The points in V are *affinely independent* if and only if the vectors $v_i - v_0$ for $i \in \{ 1, \ldots, n \}$ are linearly independent. An *affine combination* of the points v_i is given by $x = \sum_{i=0}^{n} \alpha_i v_i$ with $\sum_{i=0}^{n} \alpha_i = 1$. A *convex combination* of the v_i is an affine combination such that $\alpha_i \geqslant 0$ for all i.

Definition 2.1.1. The *convex hull* of a set of points $V = \{ v_0, \ldots, v_n \} \subset \mathbb{R}^m$ is the set of all convex combinations of points in V:

$$C = \left\{ \sum_{i=0}^{n} \alpha_i v_i \;\middle|\; \sum_i \alpha_i = 1,\ \alpha_i \geqslant 0 \right\}.$$

Definition 2.1.2. Let $V = \{ v_0, \ldots, v_n \} \subset \mathbb{R}^m$ be a set of $n+1$ affinely independent points. The convex hull $\mathrm{conv}\,(V)$ is said to be a *simplex* of dimension n, denoted by $\sigma = [v_0, \ldots, v_n]$.

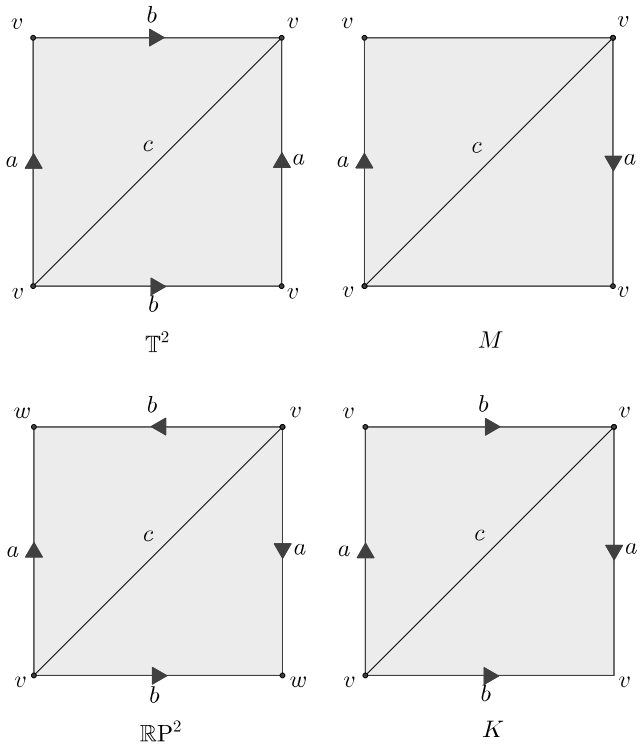


Figure 2.1: Gluing diagrams of the torus $\mathbb{T}^2$, the Möbius strip M, the projective plane $\mathbb{R}\mathrm{P}^2$ and the Klein bottle K.

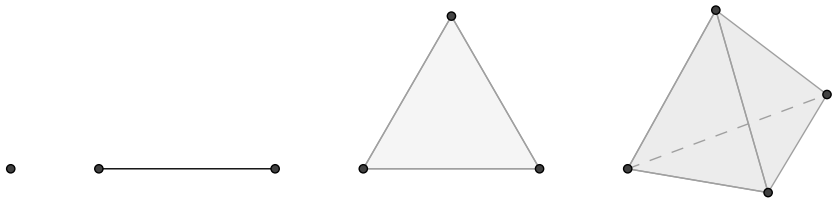


Figure 2.2: Representation of low-dimensional simplices.

The 0, 1 and 2-dimensional simplices are called vertices, edges and triangles. The 3-simplex, a tetrahedron, corresponds to the 3-dimensional extension of the triangle. These simplices are depicted in Figure 2.2.

Let σ be a simplex generated by the set of affinely independent points V. A *face* τ of σ, is the convex hull of a non-empty subset $S \subseteq V$. In particular a face is said to be *proper* if S is a proper subset of V. We will use the notation $\tau < \sigma$ if τ is a proper face of σ and $\tau \leqslant \sigma$ otherwise.

The boundary of σ, denoted bd σ is the union of all its proper faces and its interior is int $\sigma = \sigma - \text{bd } \sigma$.

2.1.2 Simplicial complexes

Simplicial complexes are particular collections of simplices, that are closed under the operation of *taking faces* and in which improper intersections of simplices are

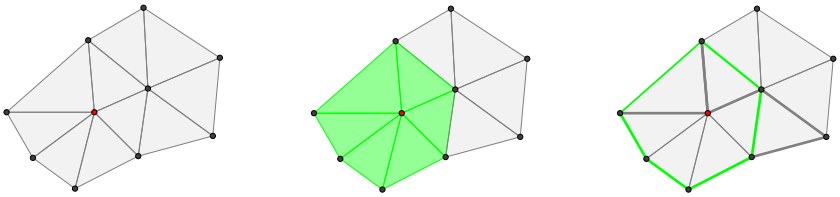


Figure 2.3: Star and link of a vertex of a simplicial complex.

forbidden. Formally, we have

Definition 2.1.3. A *simplicial complex* K is a finite collection of simplices, such that for $\sigma, \sigma_0 \in K$:

(i) if $\tau < \sigma$ then $\tau \in K$;

(ii) $\sigma \cap \sigma_0$ is a face of both simplices or empty.

The dimension of a simplicial complex K is the maximum dimension of its simplices. A subcomplex of K is a simplicial complex $L \subseteq K$. Particular subcomplexes of K are its *skeleta*, in particular the k-skeleton is defined as the set containing all simplices of K of dimension at most k. The *underlying space* of K, denoted as $|K|$, is the *polyhedron* given by the union of its simplices with the topology it inherits from $\mathbb{R}^m$. Let $\mathbb{X}$ be a topological space, it is said *triangulable* if it has a *triangulation* given by a homeomorphism $\Phi : \mathbb{X} \to |K|$, where K is a simplicial complex.

The *star* of a simplex $\tau \in K$ is the set of its cofaces, i. e. $\operatorname{St} \tau = \{ \sigma \in K \mid \tau \leqslant \sigma \}$ which is generally not a subcomplex of K. Hence, we can consider its closure, the *closed star* of τ denoted by $\overline{\operatorname{St}}\, \tau$, which is the smallest subcomplex containing $\operatorname{St} \tau$. The *link* of τ is the collection of all simplices in its closed star that does not intersect τ. See Figure 2.3 for a representation of the star and the link of a vertex of a simplicial complex.

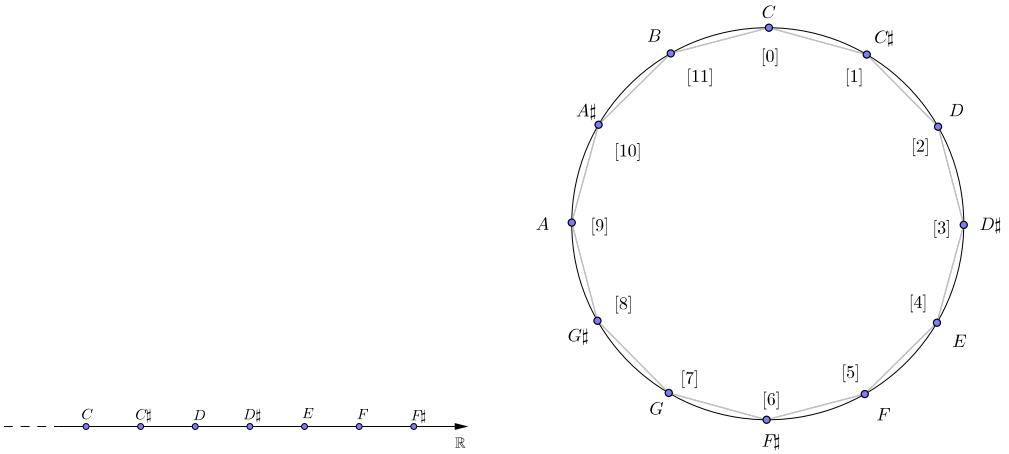
A simplicial complex of dimension 2 can be described as a purely combinatorial object, starting with a set of vertices, then attaching the edges to obtain a graph and finally, adding triangles to the graph's structure. In the case of higher dimensional simplicial complexes, according to (Hatcher, 2002, Sec. 2.1), since the simplices of a simplicial complex K are univocally determined by their vertices, it is possible to give a combinatorial interpretation of K, as a set K_0 of vertices, with sets K_n of n-simplices, i. e. $(n+1)$-element subsets of K_0. In addition, every subset of $(k+1)$-element subset of the vertices of K_n has to be a k-simplex, in K_k.

2.2 The geometrical approach: continuous models

2.2.1 From pitch labels to continuous frequencies

When considering the equal temperament, given the fundamental frequency ν of a note it is possible to represent its pitch as a real number through the function $p : (0, +\infty) \to \mathbb{R}$ defined by

$$p(\nu) := 69 + 12 \log_2 \left(\frac{\nu}{440} \right). \tag{2.2.1}$$



(a) $\mathbb{R}$ (b) $S^1 = \mathbb{R}/12\mathbb{Z}$.

Figure 2.4: The linear space of pitches and the space of pitch classes.

The majority of humans, including either trained listeners or musicians are not sensitive to *absolute* note frequencies but rather to their ratios. This suggests that a notion of *distance* in the mathematical space of notes should be defined in terms of ratios of their fundamental frequencies. The advantage brought by Equation (2.2.1) is that we are able to deal with subtractions (which are handier, compared to the ratios). It is thus reasonable to interpret the *pitch space* as the metric space $(\mathbb{R}, d)$, where d is the distance induced by the absolute value: $d(p, q) \coloneqq |p - q|$. Observe that this model implies the existence of infinitely many notes between any two pitches p and q. A way to visualise this concept is to image a *continuous glissando* of an instrument such as the violin or the trombone, or even a human voice. However, the values corresponding to the notes actually played in music (by a piano or a clarinet, for instance) are in fact specific *integer* numbers. This is due to the choice of working in the equal temperament framework, where the octave is subdivided into 12 equally spaced subintervals, so that the ratio of two consecutive semitones is $2^{1/12}$.

In this work, we assume continuous trajectories among notes (represented as points of a space) to be paths between one discrete state of the space to another, as they are defined by equal tuning.

In order to carry out a more qualitative and deeper analysis, hence reaching a visualisation of the harmonic essence of a piece, we must consider *pitch classes*, obtained by identifying pitches modulo octave:

$$[p] \coloneqq \{\, p + 12k \mid k \in \mathbb{Z} \,\}\,. \tag{2.2.2}$$

This amounts to take the quotient space $\mathbb{R}/12\mathbb{Z} \cong \mathcal{S}^1 =: \mathbb{T}^1$, which we endow with the distance

$$\bar{d}([p], [q]) \coloneqq \min \{\, |p - q| \mid p \in [p], q \in [q] \,\}\,;$$

we call $(\mathbb{T}^1, \bar{d})$ the *pitch class space*.

Thanks to the definitions given above, it is possible to start modelling objects belonging to the domain of harmony. Several studies aiming at a geometric description

of the chord space have already been developed, in particular by D. Tymoczko and others in (Tymoczko, 2006, 2008; Callender et al., 2008; Tymoczko, 2011). In Music, a *chord* is the simultaneous execution of two or more notes (say n, in general) modulo octave, which translates in mathematical language into an n-tuple of real numbers (i. e. a point of $\mathbb{R}^n$). Since in harmony, one is not sensitive to octaves when studying the relations between chords or notes, we actually think in terms of pitch classes. Hence, an n-tuple of pitch classes is, in principle, a point of the n-dimensional torus $\mathbb{T}^n$. However, chords where notes (or pitch classes) are permuted are considered equivalent from the harmonic point of view. Therefore, if we ignore the order in which the notes are arranged, we have to quotient $\mathbb{T}^n$ by the symmetric group $\mathcal{S}_n$, and we come up with the mathematical definition of chord. In what follows we shall always assume $n \geqslant 2$.

Definition 2.2.1 (n-dimensional pitch space)**.** A tuple of n notes $(p_1, \ldots, p_n)$, where $P = \{p_i\}_{i=1}^n \subseteq \mathbb{Z}_{12}$ is a point in the space

$$\mathbb{T}^n = \left(\mathcal{S}^1\right)^n.$$

The idea is to neglect the order in which notes are listed in P, thus

Definition 2.2.2 (Chord space)**.** A chord is a point in the space

$$\mathbb{A}_n = \mathbb{T}^n/\mathcal{S}_n,$$

where $\mathcal{S}_n$ is the symmetric group, that acts by permutation of the coordinates:

$$\sigma\left(x_1, \ldots, x_n\right) = \left(x_{\sigma(1)}, \ldots, x_{\sigma(n)}\right).$$

2.2.2 Geometrisation of the chord space

The n-dimensional torus can be viewed as a quotient space with respect to integer translations: $\mathbb{T}^n \cong \mathbb{R}^n/(12\mathbb{Z})^n$. Since the action of $(12\mathbb{Z})^n$ on $\mathbb{R}^n$ has no fixed points, the projection $\pi : \mathbb{R}^n \to \mathbb{T}^n$ is a covering map and therefore it preserves the local topology. Furthermore, the symmetric group S_n acts on the n-torus via diffeomorphisms (isometries) by permuting the coordinates of each point. Thus $\mathbb{A}_n$ inherits from $\mathbb{T}^n$ the structure of metric space. Moreover, since it has been obtained from a differentiable manifold through the action of a finite group, it is also an *orbifold*. We refer to (Thurston, 2002) for details on this topic. However, $\mathbb{A}_n$ is *not* a differentiable manifold, because the points fixed by the action of S_n are *singular*.[1]

The following result was proven in (Slavich, 2010) and provides a geometric characterisation of the chord spaces. The proof has been rewritten in Appendix B since the original document is written in Italian.

Theorem 2.2.1. *The space of chords $\mathbb{A}_n$ is a metric space, obtained by gluing the $(n-1)$-dimensional tetrahedral bases of a right n-dimensional prism via the equivalence relation induced by a cyclic permutation of the vertices.*

[1] A point in $\mathbb{A}_n$ is *singular* if at least 2 of its coordinates have the same value: in this case the action of the permutation group admits fixed points.

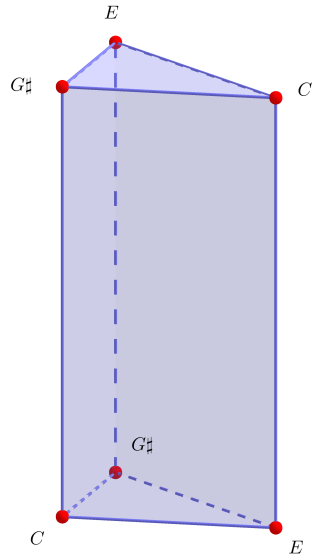


Figure 2.5: The space $\mathbb{A}_3$.

It is possible to characterise the points of the chord space by considering the number of repeated pitch classes they contain. For instance, the points of the space $\mathbb{A}_3$ depicted in Figure 2.5 are structured as follows:

(a) the points representing chords with no repeated pitch classes lie in the interior of the prism $\mathring{\mathbb{A}}_n$.

(b) the chords whose representatives are tuples of the form (x, x, y) lie on the 2-dimensional faces of the prism.

(c) the edges of the prism are constituted by unisons (modulo octave).

The voice leading between two n-notes chords can be represented as a trajectory in the chord space. The singular boundaries of the prism acts as mirrors on the trajectory (this particular feature of the chord space will be discussed in more details in Section 3.3). To help the reader's intuition, it is possible to think about these reflections in the simplified representation of the billiard table orbifold in Figure 2.6 on the facing page. The action of the group of isometries of the plane on the four sides of the table generates infinitely many collections of balls in $\mathbb{R}^2$ and the edges of the rectangle R act as mirrors respect to the trajectory of the ball.

2.3 The *Tonnetz*

The *Tonnetz* has been largely studied in computational musicology. Its structure mirrors the acoustical properties of pitch classes and the connections between its vertices highlight relevant tonal, harmonic objects, such as major and minor triads. In the following sections, we will sketch its history and we define it both as an abstract graph and a simplicial complex.

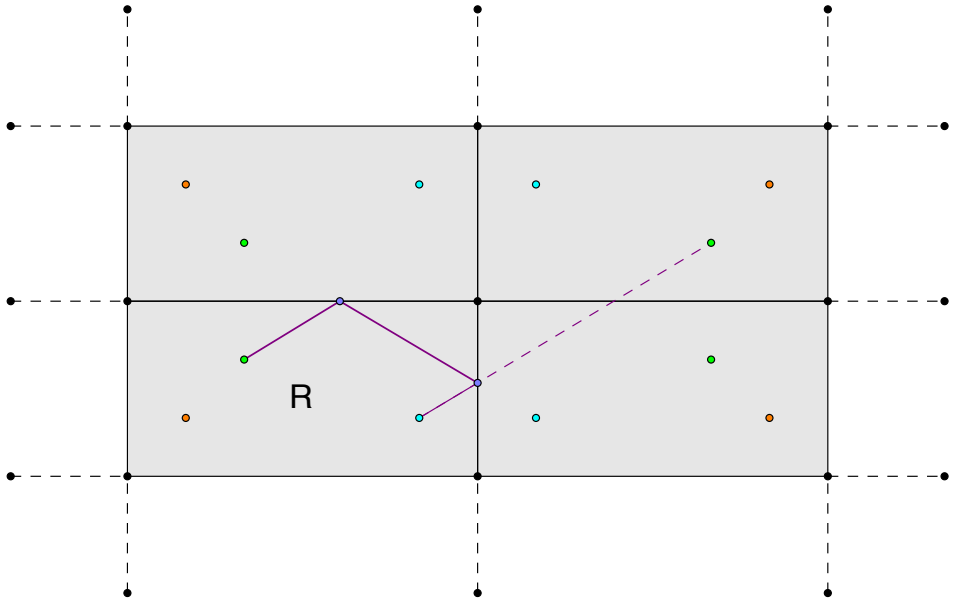


Figure 2.6: The billiard table orbifold is generated by the group of isometries of $\mathbb{R}^2$ reflecting a rectangle along its four sides. The borders of the rectangle R act as mirrors on the dashed trajectory.

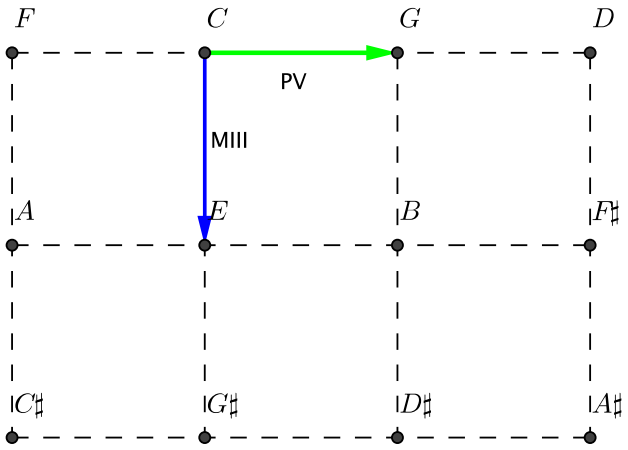


Figure 2.7: The Euler *Tonnetz*. Two pitch classes are connected by an edge, if they form a *consonant* interval. The horizontal arrow (PV) links two pitch classes a perfect fifth apart, while the two pitch classes connected by the vertical arrow (MIII) forms a major third interval.

2.3.1 An overview on tone-networks

Leonhard Euler was the first to describe a *Tonnetz* in (Euler, 1774). Although this structure has been largely generalised, see for instance (Douthett and Steinbach, 1998; Tymoczko, 2012), the original idea was to create a diagram mirroring the acoustical proximity of the pitch classes of the chromatic scale in just intonation temperament. This representation of the *Tonnetz* is depicted in Figure 2.7. Two

consecutive notes on the horizontal axis, equipped with the orientations of the arrows showed in the figure, form a perfect fifth interval (PV). On the vertical axis, a couple of consecutive notes form a major third (MIII) from top to bottom[2].

The *Tonnetz* has inspired important modern musical models. For instance, the spiral array (Chew, 2002) (in equal temperament) can be described as a *spiralisation* generalising the Euler 3×4 diagram. It is defined as a 3-dimensional helix where the position of the ith pitch class has cylindrical coordinates

$$p(i) = (\sin(i\pi/2), \cos(i\pi/2), ih),$$

where h is a fixed height parameter and $i \in \mathbb{Z}$.

Hence, consecutive pitches on the helix are arranged to form perfect fifth intervals. Moreover, the periodicity of the trigonometric functions implies that

$$\pi_{x,y}(p(i)) = \pi_{x,y}(p(i+4)),$$

where $\pi_{x,y} : \mathbb{R}^3 \to \mathbb{R}^2$ is the canonical projection. Thus, two such points differ only in their last coordinate, and represent a major third interval. See Figure 2.8 for a representation of the spiral array and an example of the two configuration of pitch classes described above.

If the aim of the *Tonnetz* was to represent the acoustical nearness among the 12 notes of the chromatic scale, the first infinite *Tonnetz* was introduced by von Oettingen in 1866. A new direction on the graph can be considered as relevant: the notes on the left-bottom/right-top diagonals in the Euler's matrix are minor third intervals. Thus it is possible to extend the diagram of Euler as a infinite triangular planar lattice.

To safely define the *Tonnetz* in the Graph Theory formalism, we introduce the following definitions.

Definition 2.3.1 (Abstract graph)**.** An *abstract unoriented graph* is a pair (V, E) where V is a finite non-empty set and E is a non-empty set of unordered pairs of different elements of V. Thus, an element of E is of the form $\{v, w\}$ where v and w belong to V and $v \neq w$. We call vertices the elements of V and edges the elements $\{v, w\}$ of E connecting v and w.

Pitches can be associated to the *Tonnetz*' vertices by defining a labelling function $l_V : V \to L$. It is clear how it is possible to associate to the Euler's diagram a set of vertices, which in terms of pitch classes correspond to the chromatic scale, and associate an edge to every couple of pitches with intervallic distance equal to 7, 3 or 4 half-steps[3].

A formal definition of the *Tonnetz* as an abstract graph is given in (Zabka, 2009).

Definition 2.3.2 (Realization of a graph)**.** Let (V, E) be an abstract graph. A *realization* of (V, E) is a set of points in $\mathbb{R}^N$, whose elements are associated to vertices in V and edges are realized as segments joining the pairs $e \in E$. Such a realization is termed a *graph*. We require that the following two *intersection conditions* hold:

[2] A change of the orientation of the axis will *reverse* the intervals. A perfect fifth's inversion is a perfect fourth, while the inversion of a major third is a minor sixth.

[3] We shall always consider an octave to be splitted in 12 half-steps

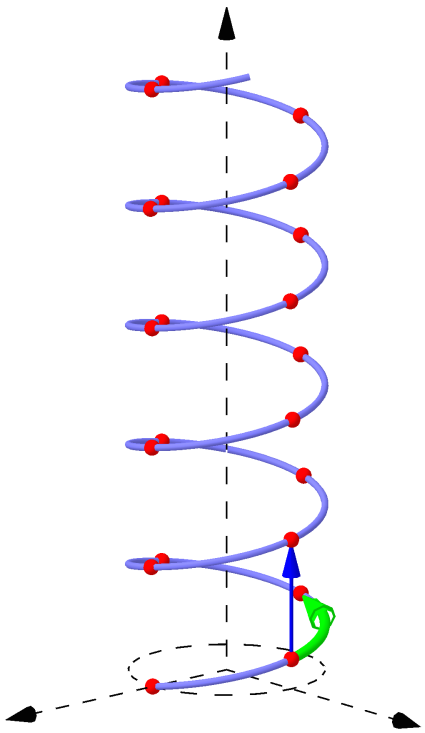


Figure 2.8: The spiral array. Two consecutive pitch classes lying on the helix are a perfect fifth apart (considering the orientation of the curved arrow), while the vertical arrow connects two pitch classes a major third far from each other.

1. two edges meet either in a common end-point or not at all;
2. no vertex lies on an edge except at one of its ends.

It is possible to represent the *Tonnetz* as a geometric realisation of an abstract graph corresponding to a 2-dimensional triangular lattice, whose edges are determined by three translation functions of the form

$$\begin{aligned} \tau_i : \mathbb{Z}/12\mathbb{Z} &\to \mathbb{Z}/12\mathbb{Z} \\ p &\mapsto p + i \mod 12, \end{aligned}$$

where $i \in \{3, 4, 5\}$ and $p \in L_V$ is the set of labels equipped with the labelling function l_V. See Figure 2.9 for a visualization of the *Tonnetz*.

The cardinality of the set of unique vertices of the *Tonnetz* $T(\tau_1, \tau_2, \tau_3)$ is determined by the order of the translations involved in its construction. In particular, it is the maximum of the orders of the translation maps involved, and corresponds to the whole chromatic scale if and only if τ_i generates $\mathbb{Z}/12\mathbb{Z}$ for some $i \in \{1, 2, 3\}$. In particular $T(3, 4, 5)$ contains the whole chromatic scale since 5 is a generator of $\mathbb{Z}/12\mathbb{Z}$.

2.3.2 The *Tonnetz* as a Simplicial Complex

Thanks to the theory introduced in Section 2.1, it is possible to give a simplicial complex interpretation of the *Tonnetz*, as originally suggested in (Bigo et al., 2013).

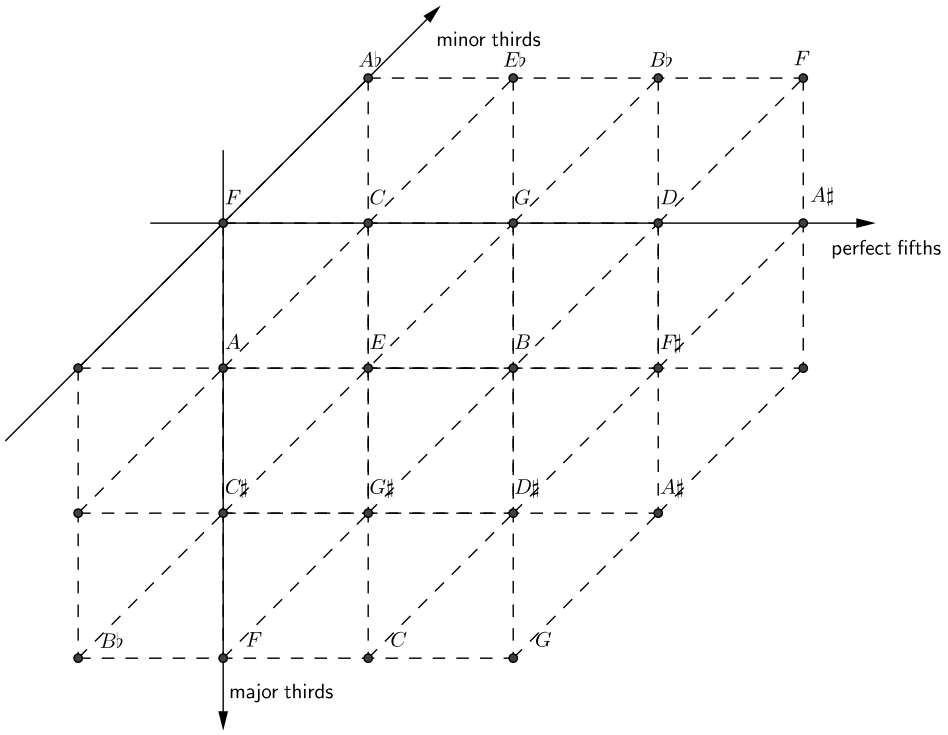


Figure 2.9: Realization of the *Tonnetz* as a tiling of the plane.

The vertices of the graph in Figure 2.9 correspond to 0-simplices, edges to 1-simplices and the 2-simplices are attached to the structure defined by the 1-skeleton we just provided. In particular, considering the labels inherited by the graph we have that the 0-simplices correspond to pitch classes, 1-simplices to perfect fifth, major third and minor third intervals[4] and 2-simplices to major and minor triads. In Figure 2.10 the 2-simplices are labeled as triads. The label corresponds to the triad generated by the superposition of the notes on the triangle's vertices. For instance, the triad of C major corresponds to $[C, E, G]$, while C minor is associated to $[C, E\flat, G]$.

In the remainder of this work, we will refer to the *Tonnetz* as a simplicial complex, denoting it by T and its underlying space by $|T|$. In particular, we define an extended shape E of the *Tonnetz* as a subcomplex $E \subset T$.

Given a topological space $\mathbb{X}$ and a discrete group G acting on it, a *fundamental domain* of the action of G on $\mathbb{X}$ is an open set $S \subset \mathbb{X}$, such that the projection $\pi : \mathbb{X} \to \mathbb{X}/G$ is injective when restricted on S and surjective on $\bar{D}$. Observe that a *fundamental domain* of the *Tonnetz* corresponds to a region which is the torus generated by the major and minor third intervals. Geometrically, it is realised by identifying the horizontal and vertical edges of the square represented in Figure 2.10 on the next page, according to the labels of the vertices. In the remainder of this work, we shall denote such a region by F.

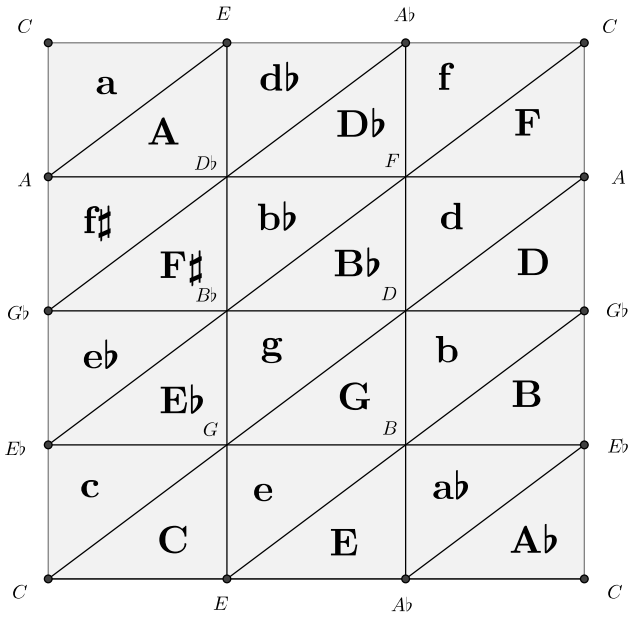


Figure 2.10: The gluing diagram of the *Tonnetz* torus. Pitch classes correspond to 0-simplex. Each triangle represents either a major or a minor triad denoted by a bold label, with major triads indicated by capital letters.

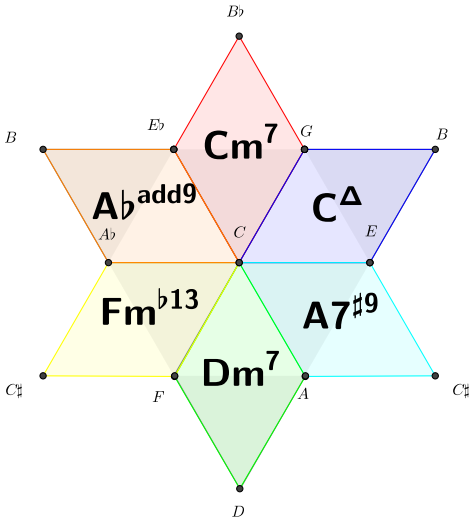
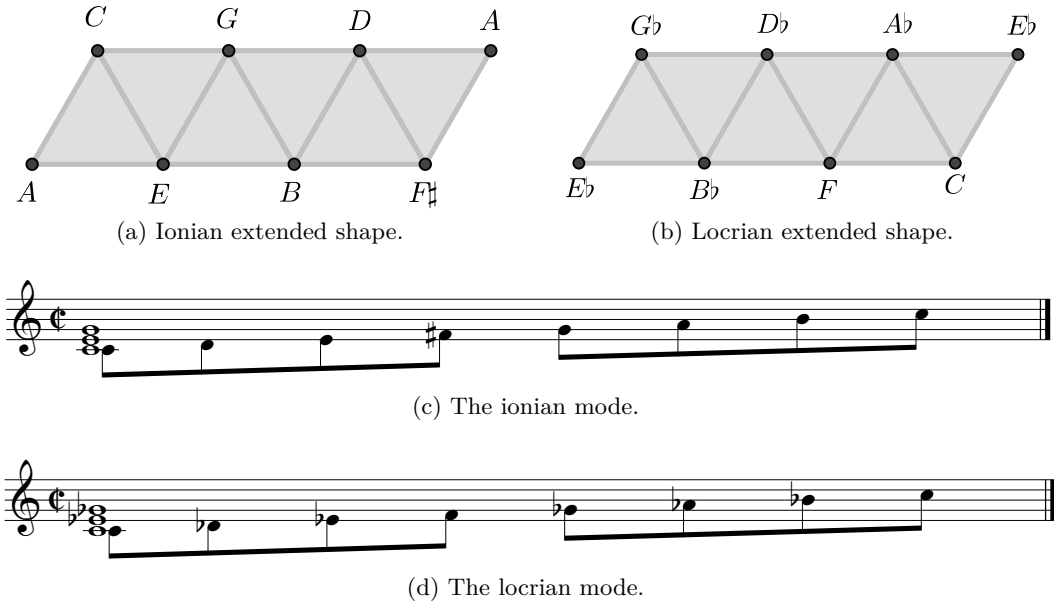


Figure 2.11: Simple shapes and four notes chords.

Extended Shapes on the Tonnetz

The extended shape generated by a trace of the pitch classes played in a music phrase on the *Tonnetz* depends on the intervals among the notes involved in the phrase. However, it would not be possible to distinguish geometrically the subcomplexes associated to a **C△** and **Cm7** (modern chord notations and the definition of triad, seventh chord and altered chord are detailed in Appendix E), both corresponding to

[4]Or their inversions depending on the orientation of the edges.



(a) Ionian extended shape.

(b) Locrian extended shape.

(c) The ionian mode.

(d) The locrian mode.

Figure 2.12: Extended shapes on the *Tonnetz*. Two different modes are represented by the same extended shape.

the subcomplex generated attaching two adjacent triangles of T sharing an edge. In particular, more exotic chords correspond to the same shape. In Figure 2.11 some of the possible subcomplexes given by the attachment of two triangles on T are depicted. It is possible to observe in the figure, that altered chords appear next to the standard ones.

The same phenomenon occurs for modes by analysing extended shapes generated considering different modal scales. In this context, we refer to a mode as a scale supported by a fundamental note or a chord defining the set of resolutions and tensions in the scale. (See Appendix A for details on modern modal theory). In Figures 2.12a and 2.12b we show how the same extended shape is associated to two different modes. Figures 2.12a and 2.12b have been realised with the software Hexachord[5] from MIDI files corresponding to the partitions of Figures 2.12c and 2.12d. The idea that led to the model we shall present in Part III is to define a *preferred subcomplex* of the fundamental domain of the *Tonnetz*, generated considering the pitch classes and the durations of musical phrases.

[5]Developed by Louis Bigo during his Ph.D. thesis and available at `http://www.lacl.fr/~lbigo/recherche`.

Part II

The horizontal dynamics of music: an algebraic and topological viewpoint on voice leading theory

Table of Contents

Abstract

This part focuses on the analysis of voice leadings, i. e. the transformation of a sequence of chords into a collection of superposed melodies in simultaneous motion. In Chapter 3, the musical idea of voice leading is formalised from a mathematical viewpoint as a multiset: an unordered collection of pitches where repetitions of the same element are allowed. Thereafter, a representation of voice leadings as partial permutations is described.

This algebraic approach is re-interpreted geometrically in Section 3.3. Voice leadings become geodesics and their concatenation a piecewise geodesic path in the space of pitches, pitch classes and the chord space. The different kind of simultaneous voice motions are analysed in each space, pointing out how minimal geodesics paths represents non-crossing voice leadings among two chords. In Section 3.4 we show how partial permutation matrices encode the information concerning simultaneous motions of n voices, including possible crossings among pairs of voices. We propose a method to represent different kinds of voice leadings used in a piece as a multiset of points endowed with a multiplicity. Then, we suggest a simple extension of this model to other contrapuntal species than the first one. Sequences of voice leadings, described as 5-dimensional points, are seen as multi-dimensional time series and compared using dynamic time warping.

Finally, in Chapter 4, partial singular braids are introduced as a tool for the visualisation of partial permutations and, hence, voice leadings. Indeed, by selecting a particular class of braids it is possible to visualise voice leading among chords of n notes in the 3-dimensional Euclidean space. Then, the first model is extended to take into account the intervallic leap of each voice. In conclusion of this chapter, we analyse the behaviour of the model in the space of pitch classes, analysing four examples previously discussed in (Tymoczko, 2011).

This part represents joint work with Alessandro Portaluri and Riccardo Jadanza.

Three

Voice leadings, partial permutations and geodesics

Counterpoint represents the melodic point of view of composition, reflecting a horizontal way of thinking. In the particular case of simultaneous motion of voices, the attention is centered on the composition of multiple, independent melodies that end up forming a sequence of chords. This choice allows to compose melodies affecting the listener both as a whole (chords) and as different autonomous fluxes of notes (parts). In the following sections, we shall focus on the formalisation of the *voice leading* process, also called *part writing*, that is the evolution and the interaction of parts or voices in a sequence of chords.[1] Intuitively, we can think of it as the assignment of a melody to a certain instrument, when more than one melody is played by more than one instrument at the same time.[2]

3.1 Defining the voice leading

In general, it is possible to describe a melody as a finite sequence of ordered pairs of pitches or pitch classes $(p_i, p_{i+1})_{i\in I}$, where I is a finite set of indices. See Section 2.2 for the definition of pitch and pitch class. In order to model the voice leading in a mathematical way it is necessary to introduce first the concept of *multiset*, a generalisation of the idea of set. This approach was already considered in (Tymoczko, 2006). Formally, a *multiset* M is a couple (X, μ) composed of an *underlying set* X and a map $\mu : X \to \mathbb{N}$, called the *multiplicity* of M, such that for every $x \in X$ the value $\mu(x)$ is the number of times that x appears in M. In layman terms, we can think of a multiset as of a list, where an object can appear more than once, whilst the elements of a set are necessarily unique. As an example consider $L = [a, a, a, b, b, c]$. The underlying set of L is $X = \{\, a, b, c \,\}$ and the multiplicity function μ takes values $\mu(a) = 3,\ \mu(b) = 2,\ \mu(c) = 1$.

We define the *cardinality* $|M|$ of M to be the sum of the multiplicities of the elements of its underlying set X. Observe, however, that a multiset is in fact completely defined by its multiplicity function: it suffices to set $M \coloneqq \big(\mathrm{dom}(\mu), \mu\big)$.

Definition 3.1.1. Let L and M be two finite multisets, such that $|L| = |M|$. A

[1] Here the term "chord" is used in the musical sense, not necessarily as a point of the space $\mathbb{A}_n$.

[2] It is possible to think in terms of voice leading even in non-compositional contexts: for instance, a guitarist reading a partition makes a part-writing choice, deciding to play a note on a certain string. Thus we can imagine the six strings as a choir composed by six singers playing together.

bijection between L and M is the multiset $\Phi \subset L \times M$, such that

$$\Phi = \{(l_1, m_1), \ldots, (l_n, m_n)\},$$

where $L = \{l_1, \ldots, l_n\}$ and $M = \{m_1, \ldots, m_n\}$.

If we interpret a set of n singing voices (or parts played by n instruments, or both) as a multiset of pitches of cardinality n, then a voice leading can be mathematically described as follows.

Definition 3.1.2. Let $M \coloneqq (X_M, \mu_M)$ and $L \coloneqq (X_L, \mu_L)$ be two multisets of pitches with same cardinality n and arrange their elements into n-tuples $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ respectively.[3] A *voice leading* of n voices between M and L, denoted by $(x_1, \ldots, x_n) \to (y_1, \ldots, y_n)$, is the (bijection) multiset

$$Z \coloneqq \{(x_1, y_1), \ldots, (x_n, y_n)\},$$

whose underlying set is $X_Z \coloneqq X_M \times X_L$ and whose multiplicity function μ_Z is defined accordingly, by counting the occurrences of each ordered pair.

Remark 1. Observe that the definition just given is not linked to the particular type of object (pitches): it is possible to describe voice leadings also between pitch classes, for instance.

Note that it is also possible to describe a voice leading as a bijective map from the multiset M to the multiset L, i. e. as a partial permutation of the *union multiset*

$$M \cup L \coloneqq (X_M \cup X_L, \mu_{M \cup L}),$$

where

$$\mu_{M \cup L} \coloneqq \max\{\mu_M \chi_M, \mu_L \chi_L\}$$

and χ_M and χ_L are the characteristic functions of X_M and X_L, respectively.[4]

3.2 Partial permutations

A *partial permutation* of a finite (multi)set S is a bijection among two fixed sub(multi)set of S. For instance, this function can be a string of n symbols, in which we admit $\diamond$ as a special character to denote the empty character. In this definition the domain of the partial permutation is constituted by the position indices of the non-empty elements of the string. For instance the string "1 1 $\diamond$ 2" represents the partial permutation of domain $\{1, 2, 4\}$. The symbol 1 fixed, 2 is mapped to 1 and 4 to 2. The corresponding cycle notation is

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & \diamond & 2 \end{pmatrix},$$

where the two sub(multi)sets corresponds to the rows of the matrix, the mapping to the columns and $\diamond$ is associated to *unmapped* elements.

[3] These are in fact the images of two bijective maps $\psi_M : \{1, \ldots, n\} \to M$ and $\psi_L : \{1, \ldots, n\} \to L$., where M and L are understood as "sets" with (possibly) repeated elements.

[4] For a multiset S we assume that $\mu_S(x) = 0$ if $x \notin X_S$. Under this assumption, the function $\mu_{M \cup L}$ is defined on the entire set $X_M \cup X_L$.

Remark 2. In order to be able to do computations with partial permutations, it is fundamental to fix an *ordering* among the elements of the union multiset $M \cup L$. We give to $M \cup L$ the natural ordering $\leqslant$ of real numbers, being its elements pitches. Indeed, in classical music with equal temperament, one defines the pitch p of a note as a function of the fundamental frequency using the Equation (2.2.1). This can be done also in the case where the elements of the union multiset are pitch classes: the ordering is induced by the ordering of their representatives belonging to a same octave.

Example 3.2.1. The voice leading

$$(G_2, G_3, B_3, D_4, F_4) \to (C_3, G_3, C_4, C_4, E_4) \tag{3.2.1}$$

is described by the partial permutation of the ordered union multiset

$$(G_2, C_3, G_3, B_3, C_4, C_4, D_4, E_4, F_4)$$

defined by

$$\begin{pmatrix} G_2 & C_3 & G_3 & B_3 & C_4 & C_4 & D_4 & E_4 & F_4 \\ C_3 & \diamond & G_3 & C_4 & \diamond & \diamond & C_4 & \diamond & E_4 \end{pmatrix}. \tag{3.2.2}$$

Thus, a voice leading between two multisets of n voices can be seen as a partial permutation of a multiset whose cardinality is less than or equal to $2n$.

The next step is to associate a representation matrix with the partial permutation. Let V be an n-dimensional vector space over a field $\mathbb{F}$ and let $\mathcal{E} \coloneqq \{e_1, \ldots, e_n\}$ be a basis for V. The symmetric group S_n acts on $\mathcal{E}$ by permuting its elements: the corresponding map $S_n \times \mathcal{E} \to \mathcal{E}$ assigns $(\sigma, e_i) \mapsto e_{\sigma(i)}$ for every $i \in \{1, \ldots, n\}$. We consider the well-known *linear representation* $\rho : S_n \to \mathrm{GL}(n, \mathbb{F})$ of the group S_n given by

$$\rho(1\ i) \coloneqq \begin{pmatrix} 0 & & & 1 & & & \\ & 1 & & & & & \\ & & \ddots & & & & \\ & & & 1 & & & \\ 1 & & & & 0 & & \\ & & & & & 1 & \\ & & & & & & \ddots \\ & & & & & & & 1 \end{pmatrix},$$

where the 1's in the first row and in the first column occupy the positions $1, i$ and $i, 1$ respectively. The map ρ sends each 2-cycle of the form $(1\ i)$ to the corresponding permutation matrix that swaps the first element of the basis $\mathcal{E}$ for the i-th one. Note that each row and each column of a permutation matrix contains exactly one 1 and all its other entries are 0. Following this idea and (Horn and Johnson, 1991, Definition 3.2.5, p. 165), we say that a matrix $P \in \mathrm{Mat}(m, \mathbb{R})$ is a *partial permutation matrix* if for any row and any column there is at most one non-zero element (equal to 1). When dealing with a voice leading $M \to L$, the dimension m of the matrix P is equal to the cardinality of the multiset $M \cup L$.

Remark 3. In general, the partial permutation matrix associated with a given voice leading is not unique. This is due to the fact that we are dealing with multisets: if $M \to L$ is a voice leading it is possible that some components of L have the same value, i. e. that different voices are playing or singing the same note.

For this reason we introduce the following convention.

Convention 1. Let $M \coloneqq (x_1, \ldots, x_n) \to L \coloneqq (y_1, \ldots, y_n)$ be a voice leading and suppose that more than one voice is associated with a same note of L. To this end, let $(x_{i_1}, \ldots, x_{i_k})$ be the pitches of M (with $i_1 < \cdots < i_k$) that are mapped to the pitches $(y_{j_1}, \ldots, y_{j_k})$ of L, with $y_{j_1} = \cdots = y_{j_k}$ and $j_1 < \cdots < j_k$. In order to uniquely associate a partial permutation matrix $P \coloneqq (a_{ij})$ with the above voice leading, we assign the value 1 to the corresponding entries of P by following the order of the indices, that is by setting $a_{i_1 j_1} = 1, \ldots, a_{i_k j_k} = 1$.

Thus, we shall henceforth speak of *the* partial permutation matrix associated with a given voice leading.

Example 3.2.2. The partial permutation matrix associated with the cycle representation of Equation (3.2.2) of voice leading represented in Equation (3.2.1) is

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Therefore, if $M \to L$ is a voice leading, if both M and L are thought of as ordered tuples and if P is its partial permutation matrix, we have that $PM = L$; in addition, the "reversed" voice leading $L \to M$ is obviously described by the transposed P^{T} of P: $P^{\mathsf{T}} L = M$.

This representation has the advantage of providing objects that are much handier than a multiset of pairs, speaking in computational terms. Algorithm 3.1 presents the pseudocode for the computation of the partial permutation matrix of a voice leading.

Algorithm 3.1 Computing the partial permutation matrix.

Input:
$M \to L$ ▷ Source (M) and target (L) multisets describing the voice leading
Output:
P ▷ Partial permutation matrix associated with the voice leading

Evaluate multiplicities of all $x \in M$ and all $y \in L$;
Generate the *ordered* multiset $U := M \cup L$;
Initialise $P \in \mathrm{Mat}(|U|, \mathbb{R})$ by setting $P(i, j) = 0$ for all i, j;
1: **for** $i, j \in \{1, \ldots, |U|\}$ **do**
2: **if** $U(i) \to U(j)$ **then**
3: $P(i, j) = 1$
4: **end if**
5: **end for**

3.3 Voice leading and piecewise geodesic paths

We can imagine a voice leading of n voices as a sequence of n-dimensional vectors (points in $\mathbb{R}^n$), whose components are the pitches associated with each note played by each voice. An important feature of this visualisation is that the melody of a certain voice is always represented by the *same* coordinate (say the ith one) in every vector of the sequence: we can thus read it very simply by looking at the projections

$$\begin{aligned} \pi_i : \mathbb{R}^n &\to \mathbb{R} \\ (v_1, \dots, v_n) &\mapsto v_i \text{ for } i = 1, \dots, n. \end{aligned}$$

A useful way to represent this idea is to take an oriented segment joining two consecutive points u and v of $\mathbb{R}^n$, that is a path $\gamma : [0, 1] \to \mathbb{R}^n$ given by

$$\gamma(s) \coloneqq u + s(v - u). \tag{3.3.1}$$

Note that this is just a convenient graphical tool and does not mean at all that every point constituting the path is effectively "played": the only ones that are involved in the melody are the endpoints $\gamma(0) = u$ and $\gamma(1) = v$.

The main characteristic of the path presented just above is that it is a *geodesic* between the points u and v, being the n-dimensional Euclidean space flat. There are infinitely many ways to connect two points in $\mathbb{R}^n$, and we are not interested in the particular way they are joined. It makes sense to set the convention that they are linked in the simplest way possible; this choice will bring advantages also in the following, as the reader will see.

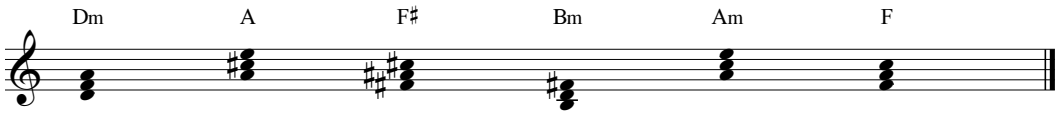
If we iterate this process for each note and for each voice we obtain a polygonal chain in $\mathbb{R}^n$, which is *not* a geodesic but rather a *piecewise geodesic*. This is not surprising and in fact quite desirable, because if we considered a melody of more than 2 notes (per voice) and if we joined the endpoints with a segment, then we would lose all the information between the two, that is we would erase the melody itself! For this reason it is meaningful to consider a *concatenation of geodesics*, which allows to reproduce every step of the music.

This is the geometric representation of what has been presented above in the algebraic form through partial permutation matrices. Indeed, if we consider a melody as a finite sequence of points (say k) in $\mathbb{R}^n$, with n the number of voices, then we can describe it geometrically through a piecewise linear path and algebraically as the product $P_k \cdots P_1$, where P_i is the partial permutation matrix of the i-th voice leading. As an example, consider the progression of triads in Figure 3.1a on the following page: each of them is represented as a triple (p_1, p_2, p_3) in $\mathbb{R}^3$, with $p_1 < p_2 < p_3$. In general it is possible to build a voice leading by associating each note of a given chord with a note of the following one, respecting the order induced by $<$. This rule has been used to draw the path in Figure 3.1b.

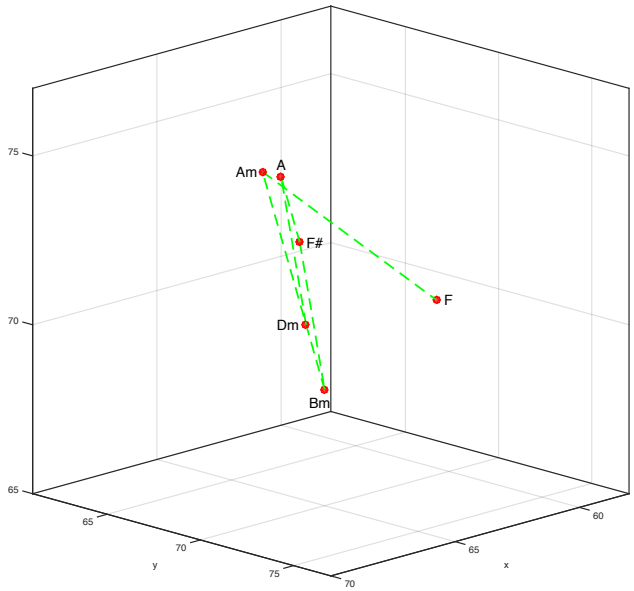
Let us now consider the four voice leadings

$$\begin{aligned} &(B_3, F_5) \to (C_4, E_4) \quad \text{and} \quad (F_5, B_3) \to (E_4, C_4), \\ &(B_3, F_5) \to (E_4, C_4) \quad \text{and} \quad (F_5, B_3) \to (C_4, E_4), \end{aligned}$$

depicted in Figure 3.2 on page 45: from the musical and perceptual viewpoint they are completely equivalent in pairs (each row describes the same voice leading).



(a) Voice leading among triads in root position.



(b) Graphic representation in $\mathbb{R}^3$ of the above voice leading.

Figure 3.1: Voice leading and corresponding piecewise geodesic path.

Generalising this fact to n voices, it is natural to identify the paths in $\mathbb{R}^n$ that are symmetric with respect to the diagonal of this space.
An immediate generalisation of this situation leads to the conclusion that if we apply the *same* permutation to both the endpoints of the paths representing a voice leading, then we obtain *de facto* the *same* voice leading.

The above discussion about symmetry, points out that it makes sense to represent a voice leading of n voices as a geodesic on the Riemannian manifold with boundary $\mathbb{R}^n/S_n$. In the special case $n = 2$ the space $\mathbb{R}^2/S_2$ is isomorphic to the half-plane

$$\mathbb{H} \coloneqq \{(x, y) \in \mathbb{R}^2 \mid x \leqslant y\}.$$

Figure 3.2 shows the voice leading between two dyads in $\mathbb{R}^2$.

It possible to represent and to analyse voice leadings as paths on more *harmony-oriented* spaces, such as the pitch class space $\mathbb{T}^n$ (whose points tuples of pitch classes) or the chord space $\mathbb{A}_n$, introduced in Section 2.2. From the harmonic point of view it is indeed admissible to ignore the octave which a certain note of a chord belongs to, and to identify each chord with the whole set of its possible voicings. We are therefore interested in geodesics on these spaces, as they will be the representation of voice leading also in this fairly general setting. The paths that we are seeking will be easily constructed once we note that $\mathbb{T}^n$ and $\mathbb{A}_n$ are obtained as identification spaces from $\mathbb{R}^n$. Therefore it suffices to draw the segments connecting the endpoints

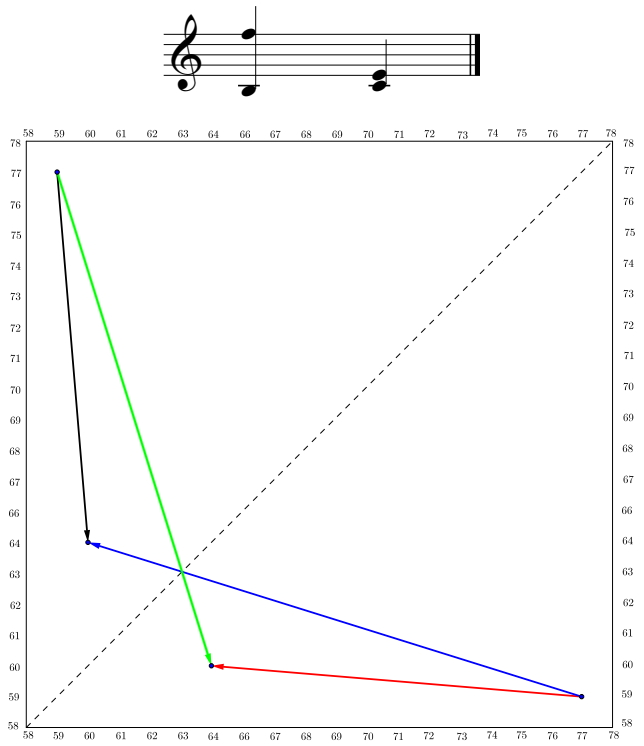


Figure 3.2: The four possible voice leadings between the notes of the score depicted above. Observe the symmetric nature of the paths with respect to the dashed line $(y = x)$.

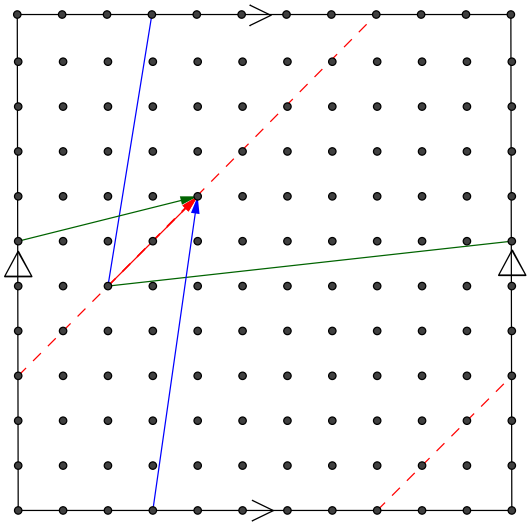
of the voice leading in $\mathbb{R}^n$, just like before, and then project them via the covering map that gives rise to the desired space. Here are some illustrated examples.

Example 3.3.1 (Voice Leading on $\mathbb{T}^2$)**.** In Figure 3.3a on the following page, the torus is described as a gluing space (see also Section 2.1). Thus, a trajectory crossing the upper border of the square in a certain point (x, u) will re-enter in the square at the point (x, l), where $y = u$ and $y = l$ are the lines where the horizontal edges lie, respectively. Symmetrically, the same argument holds for the vertical edges. Counting how may times a trajectory crosses the opposite edges of the square[5], it is possible to retrieve the number of octave leaps made by one or more voices during a voice leading (which *a priori* is lost, since we are considering pitch classes). Let us consider for this purpose the following four voice leadings in $\mathbb{R}^2$:

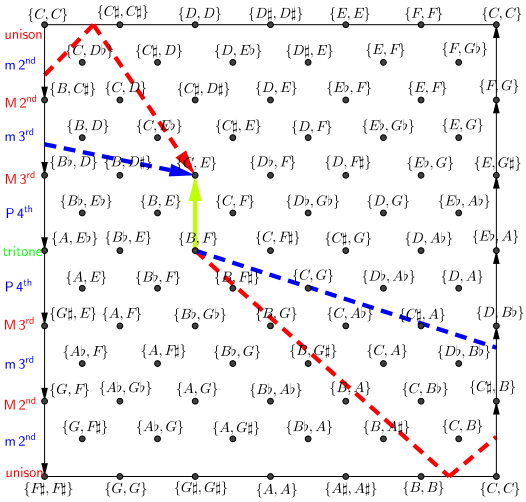
$$\text{i) } (D_0, F_0) \to (E_0, G_0), \qquad \text{ii) } (D_0, F_0) \to (E_1, G_0),$$
$$\text{iii) } (D_0, F_0) \to (E_0, G_1), \qquad \text{iv) } (D_0, F_0) \to (E_1, G_1).$$

They all represent the same voice leading $(D, F) \to (E, G)$ in the pitch class space, but their path realisations are different. Figure 3.3a on the next page displays these four paths on $\mathbb{T}^2$:

[5]It would be equivalent to consider the generators of the fundamental group of the torus $\pi_1\left(\mathbb{T}^2\right) = \mathbb{Z} \times \mathbb{Z}$. See (Hatcher, 2002, Ch. 1).



(a) Geodesics and octave leaps in $\mathbb{T}^2$.



(b) Geodesics in $\mathbb{A}_2$.

Figure 3.3: Voice leading paths in the pitch class space $\mathbb{T}^2$ and in its relative chord space $\mathbb{A}_2$. (a) The torus $\mathbb{T}^2$ is represented as a square with the usual identification rule on its sides expressed by the symbols $>$ and $\triangle$. (b) The identification rule on $\mathbb{A}_2$ is represented by the arrows on the vertical edges of the square.

Path i) is drawn as the shortest red arrow, since the jump between the dyads does not exceeds the 0-th octave;

Path ii) is represented by the green arrow, exiting the square from its right side and coming back in from its left side: this reflects the fact that the first voice makes a leap of one octave;

Path iii) is associated with the blue arrow, pointing to the top and re-entering the figure from the bottom: in this case it is the first voice that jumps to the next octave;

Path iv) is rendered by the dashed red arrow: it jumps from the top to the bottom and then from the right to the left of the square, because both voices exceed the 0-th octave.

Example 3.3.2 (Voice Leading on $\mathbb{A}_2$)**.** In the 2-dimensional case of dyads $\mathbb{A}_2 = \mathbb{T}^2/S_2$ is the *Möbius strip*. See (Bergomi et al., 2014b) for details about the construction and the positioning of the chords on the lattice. Figure 3.3b shows three different geodesic paths corresponding to the voice leading $\{B, F\} \to \{C, E\}$, where the curly braces mean that we are identifying all possible assignments of parts to each voice. Observe the identification of the left and right side of the square with inverted orientation and note the *singular boundary* of the unisons, constituted by the upper and lower side of the square. As we intuitively described in Section 2.2, the paths bounce back when touching it because of the quotient by the symmetric group. Here, it is clear that harmony is favoured over melody, as neglecting both octaves and ordering leads to focus on the *ensemble* of voices.

These examples share and show one important feature: the shortest paths joining the two pairs of pitch classes $\mathbb{T}^2$ or dyads $\mathbb{A}_2$ (the *minimal* geodesics between those two points) represent voice leading *with neither crossing nor octave leaps*, whilst the paths that touch the singular boundary correspond to part writings where at least one of these phenomena occurs.

Although the voice crossing is not advised as a standard practice in harmony manuals, it is a useful technique to avoid repeated notes, parallel fifths and hidden octaves and to assure a high degree of independence to each voice. For further details on orchestration and the use of voice crossing, see (Prout, 2012; Boland and Link, 2012; Russo, 1997; Sussman and Abene, 2012; Notley, 2007).

3.4 Simultaneous motions of the voices and complexity of a voice leading

We have seen in the previous section how the partial permutation matrix associated with a voice leading contains the information describing the path leading from one note to the next for each voice. Here, we are going to illustrate that, in fact, the tool that we have built also encodes the direction of motion of the different voices, including their crossings.

On the one hand, in music one distinguishes between three main behaviours (cf. Figure 1.12 on page 18; here, we omit parallel motion because it is not involved in our analysis):

(i) *Similar motion*, when the voices move in the same direction;

(ii) *Contrary motion*, when the voices move in opposite directions;

(iii) *Oblique motion*, when only one voice is moving.

On the other hand, with reference to a partial permutation matrix (a_{ij}), it is possible to describe the motion of a voice by noting three conditions, which are immediate consequences of the ordering of the union multiset:

1) If there exists an element $a_{ij} = 1$ for $i < j$ then the i-th voice is moving "upwards";

2) If there exists an element $a_{ij} = 1$ for $i > j$ then the i-th voice is moving "downwards";

3) If there exists an element $a_{ii} = 1$ then the i-th voice is constant.

The connection between the two worlds is the following:

- If either Condition 1) or Condition 2) is verified by two distinct elements then we have *similar motion*;
- If both Condition 1) and Condition 2) hold for two distinct elements then we are facing *contrary motion*;
- The case of *oblique motion* involves Conditions 1) and 3) or Conditions 2) and 3), for at least two distinct elements.

As we mentioned in Section 1.2, *voice crossing* is a particular case of these motions where the voices swap their relative positions. This phenomenon can be described in terms of multisets as follows.

Definition 3.4.1. Let $(x_1, \ldots, x_n) \to (y_1, \ldots, y_n)$ be a voice leading ($n \in \mathbb{N}$). If there exist two pairs (x_i, y_i) and (x_j, y_j) such that $x_i < x_j$ and $y_i > y_j$ or such that $x_i > x_j$ and $y_i < y_j$ then we say that a *(voice) crossing* occurs between voice i and voice j.

The partial permutation matrix retrieves this information, as the following proposition shows.

Proposition 3.4.1. *Consider a voice leading of n voices and let $P \coloneqq (a_{ij})$ be its associated partial permutation matrix. Choose indices $i, j, k, l \in \{1, \ldots, n\}$ such that $a_{ij} = 1$ and $a_{kl} = 1$. Then there is a crossing between these two voices if and only if one of the following conditions hold:*

i) $i < k$ and $j > l$;

ii) $i > k$ and $j < l$.

Furthermore, the total number of voices that cross the one represented by a_{ij} is equal to the number of 1's in the submatrices (a_{rs}) and (a_{tu}) of P determined by the following restrictions on the indices: $r > i$, $s < j$ and $t < i$, $u > j$.

Proof. In a partial permutation matrix the row index of a non-zero entry denotes the initial position of a certain voice in the ordered union multiset, whereas the column index of the same entry represents its final position after the transition. It is then straightforward from Definition 3.4.1 that for a voice crossing to exist either condition i) or condition ii) must be verified. Every entry a_{kl} satisfying one of those conditions refers to a voice that crosses the one represented by a_{ij}. Hence, the number of crossings for a_{ij} equals the amount of 1's in positions (r, s) such that $r > i$ and $s < j$, summed to the number of 1's in positions (t, u) such that $t < i$ and $u > j$. □

Remark 4. The fact that the number of crossings with a given voice equals the number of 1's in the submatrices determined by the entry corresponding to that voice (as explained in the previous proposition) holds true only because we assumed Convention 1. Indeed, if we did not make such an assumption, the submatrices could contain positive entries referring to voices ending in the same note but that do not produce crossings.

From what we have shown thus far it emerges that it is possible to give a qualitative description of a voice leading by counting the voices that are moving upwards, those that are moving downwards, those that remain constant and the number of crossings. We summarise these features in a 4-dimensional *complexity vector* c defined by

$$c \coloneqq (\#\text{upward voices}, \#\text{downward voices}, \#\text{constant voices}, \#\text{crossings}), \quad (3.4.1)$$

so that we are now able to classify and distinguish voice leadings by simply looking at these four aspects.

Remark 5. The notion of complexity we defined above is not equivalent nor related to the standard definitions of complexity.

Example 3.4.1. *Similar motion.* The voice leading $(C_1, E_1, G_1) \to (D_1, F_1, A_1)$ is represented by

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and its complexity vector is $(3, 0, 0, 0)$.

Oblique motion. The voice leading $(G_2, G_2, C_3) \to (C_3, C_3, C_3)$ is associated with

$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and its complexity vector is $(2, 0, 1, 0)$.

Voice crossing. The voice leading $(C_1, E_1, G_1) \to (G_1, C_1, E_1)$ is represented by

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

and its complexity vector is $(1, 2, 0, 2)$.

By virtue of these tools it is straightforward to analyse an entire first species counterpoint, by considering the concatenation of its voice leadings and thereafter we retrieve a sequence of complexity vectors. This last piece of information can be

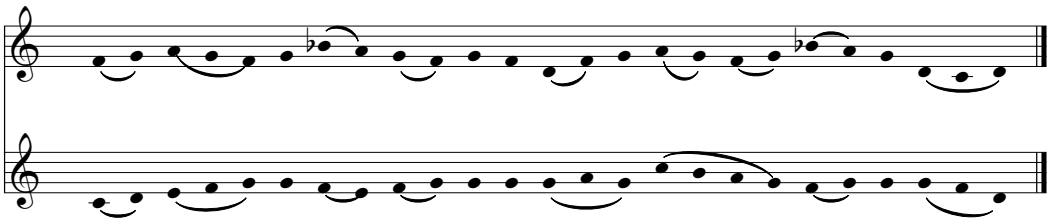


Figure 3.4: *Alleluia, Angelus Domini*, Chartres fragment n. 109, fol. 75.

visualised as a set of points in a 4-dimensional space — or rather as one or more of its 3-dimensional projections (see Subsection 3.5). In fact, if one wants to represent the complexity of the whole composition as a point cloud, one should take into account that different matrices can produce the same complexity vector. Therefore, we have a *multiset* of points in $\mathbb{R}^4$ (with non-negative integer components).

3.5 Complexity analysis of two *Chartres Fragments*

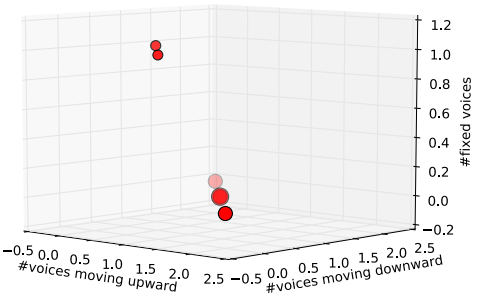
We are going to analyse two pieces that are parts of the *Chartres Fragments*, an ensemble of compositions dating back to the Middle Ages: *Alleluia, Angelus Domini* and *Dicant nunc Judei*. Both of them are counterpoints of the first species and involve only two voices. The musical interest in these compositions consists in the introduction of a certain degree of independence between the voices and the use of a *parsimonious voice leading*, i. e. an attempt to make the passage from a melodic state to the next as smooth as possible. Note how the independence of the voices is reflected by the presence of contrary motions and crossings, which can then be interpreted as a rough measure of this feature. For a complete treatise on polyphony and a historical overview we refer the reader to Taruskin (2009).

In what follows, we represent the multiplicity of each complexity vector c as a circle of centre $c \in \mathbb{R}^4$ and radius equal to the *normalised multiplicity* $\mu(c)/n$ of c, where $\mu(c)$ is the number of occurrences of c in the analysed piece and n is the total number of notes played or sung by each voice in the whole piece.

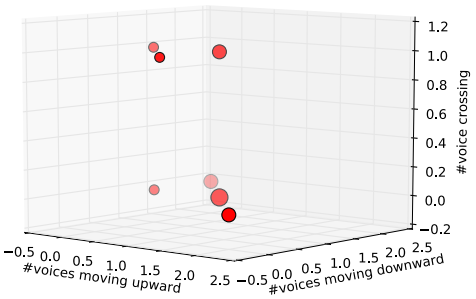
Alleluia, Angelus Domini. The fragment under examination is depicted in Figure 3.4; here is the list of its first four voice leadings, as they are generated by the pseudocode described in Algorithm 3.1:

```
Voice Leading: ['F4', 'C4'] ['G4', 'D4']
[2, 0, 0, 0] - similar motion up
Voice Leading: ['G4', 'D4'] ['A4', 'E4']
[2, 0, 0, 0] - similar motion up
Voice Leading: ['A4', 'E4'] ['G4', 'F4']
[1, 1, 0, 0] - contrary motion
Voice Leading: ['G4', 'F4'] ['F4', 'G4']
[1, 1, 0, 1] - contrary motion - 1 crossing
```

Table 3.1 on page 52 contains the the complexity vectors and their occurrences in the piece. The point cloud associated with this multiset is represented in Figure 3.5.



(a) Projection neglecting the *crossing* component of the complexity vectors.



(b) Projection neglecting the *constant voices* component of the complexity vectors.

Figure 3.5: Three-dimensional projections of the complexity cloud of the paradigmatic voice leading *Alleluia, Angelus Domini*. The radius of each circle represents the normalised multiplicity of the corresponding complexity vector.

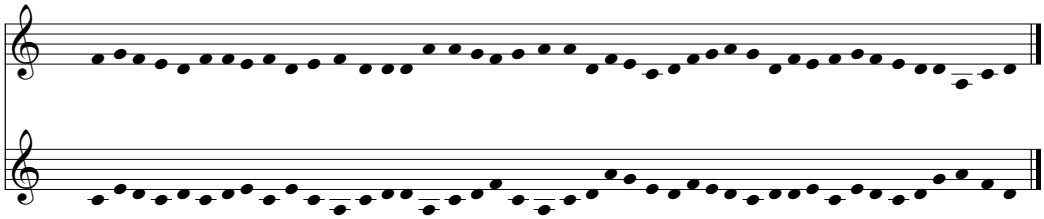


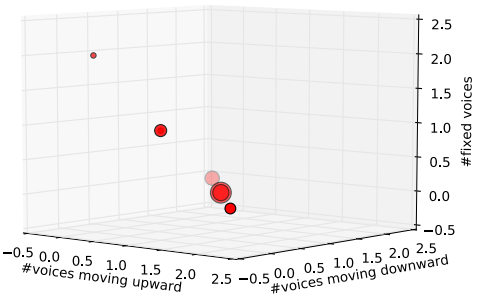
Figure 3.6: *Dicant nunc Judei*, Chartres fragment.

Observe how the projection that neglects the component of c corresponding to the number of constant voices (Figure 3.5b) gives an immediate insight on the relevance of voice crossing in the piece.

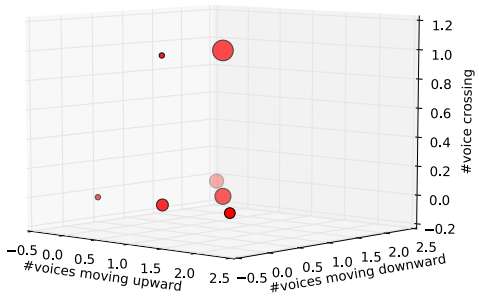
Dicant nunc Judei. The first part of the output of Algorithm 3.1 produces the following analysis:

```
Voice Leading: ['F4', 'C4'] ['G4', 'E4']
[2, 0, 0, 0] - similar motion up
Voice Leading: ['G4', 'E4'] ['F4', 'D4']
[0, 2, 0, 0] - similar motion down
Voice Leading: ['F4', 'D4'] ['E4', 'C4']
[0, 2, 0, 0] - similar motion down
Voice Leading: ['E4', 'C4'] ['D4', 'D4']
[1, 1, 0, 1] - contrary motion - 1 crossing
```

The complexity vectors arising in the whole piece and their multiplicities are again collected in Table 3.1; see Figure 3.7 instead for a visualisation of the point cloud describing the piece. Note how the voice crossing is more massive than in the point cloud describing *Alleluia, Angelus Domini*. In addition, the point $(0, 0, 0)$ in Figure 3.7b corresponds to the point $(0, 0, 2, 0) \in \mathbb{R}^4$, that represents trivial voice leadings where both parts do not vary.



(a) Projection neglecting the *crossing* component of the complexity vectors.



(b) Projection neglecting the *constant voices* component of the complexity vectors.

Figure 3.7: Three-dimensional projections of the complexity cloud of the paradigmatic voice leading *Dicant nunc Judei*. The radius of each circle represents the normalised multiplicity of the corresponding complexity vector.

Alleluia, Angelus Domini	
c	$\mu(c)$
$(0,1,1,0)$	2
$(0,1,1,1)$	2
$(0,2,0,0)$	4
$(1,0,1,1)$	2
$(1,1,0,0)$	6
$(1,1,0,1)$	4
$(2,0,0,0)$	4

Dicant nunc Judei	
c	$\mu(c)$
$(0,0,2,0)$	1
$(0,2,0,0)$	7
$(1,0,1,0)$	5
$(1,0,1,1)$	1
$(1,1,0,0)$	9
$(1,1,0,1)$	15
$(2,0,0,0)$	4

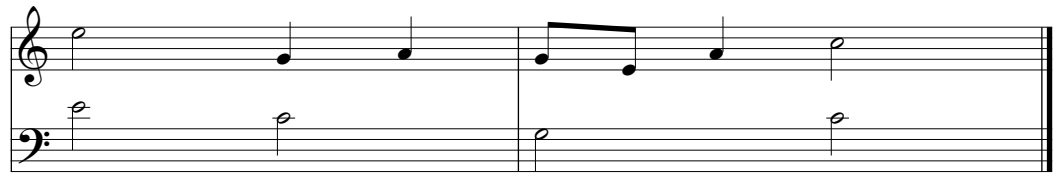
Table 3.1: Complexity vectors of the analysed fragments and their occurrences.

3.6 Rhythmic independence and rests

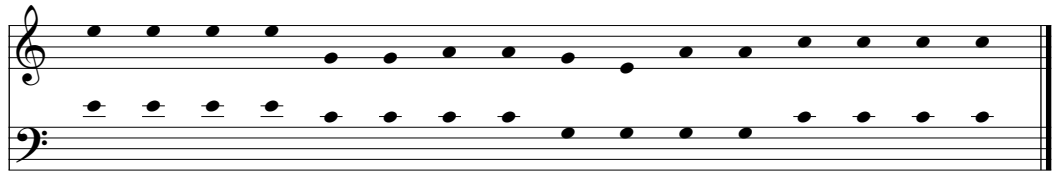
The examples analysed in Subsection 3.5 are counterpoints of the first species — which is the simplest case, voices follow a *note-against-note* flow. It is, however, possible to study more complex scenarios by introducing *rhythmic independence* between voices and *rests* in the melody, reducing non-simultaneous voices to the simplest case.

If the voices play at different rhythms or follow rhythmically irregular themes, we consider the minimal rhythmic unit u appearing in the phrase and *homogenise* the composition based on that unit: if a note has duration ku, with $k \in \mathbb{N}$, we represent it as k repeated notes of duration u (see Figure 7.4 for an example). This transformation of the original counterpoint introduces only oblique motions and does not alter the number of the other types of motion.

In musical terms, if a voice is silent it is neither moving nor being constant and it cannot cross other voices. Therefore, in order to include rests in our model it is necessary to slightly modify Algorithm 3.1 by introducing a new symbol (p) in the dictionary of pitches. We also choose to indicate a rest in the matrices associated with a voice leading by the entry -1. We adopt the following convention concerning the ordered union multiset.



(a) Counterpoint of the fifth species.



(b) Reduction to the first species.

Figure 3.8: Reduction of rhythmically independent voices to a counterpoint of the first species.

Convention 2. We choose rests to be the last elements in the ordered union multiset associated with a voice leading. In other words, we declare p to be strictly greater than any other pitch symbol.

Example 3.6.1. The voice leading $(p, D_4, D_5) \to (D_4, C_3, C_3)$ corresponds to the matrix

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}.$$

Remark 6. Note that when introducing the -1's in the matrix associated with a voice leading, we are no longer dealing with partial permutation matrices. However, to study voice leadings with rhythmic independence of the voices as before (thus ignoring rests) it is enough to consider the minor of the matrix obtained by deleting all rows and columns containing -1 (which is obviously again a partial permutation matrix).

We extend the complexity vector defined previously in Formula (3.4.1) by adding a fifth component that counts the number of voices that are silent at least once in the voice leading, i. e. it counts the number of negative (-1) entries of the associated matrix. Furthermore, we slightly modify also the notion of *normalised multiplicity* of a complexity vector c, needed for the representation of the complexity of a piece in the form of a point cloud, now dividing the number $\mu(c)$ of occurrences of c in the piece by the total number of notes per voice *after the homogenisation*.

3.6.1 Example: the *Retrograde Canon* by J. S. Bach

We consider the *Retrograde Canon* (also known as *Crab Canon*), a palindromic canon with two voices belonging to the *Musikalisches Opfer* by J. S. Bach, the beginning of which is reproduced in Figure 3.9.

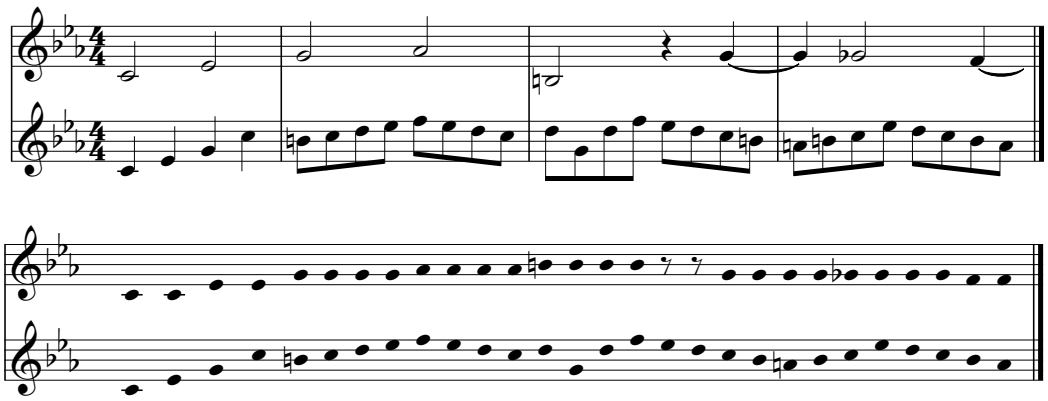


Figure 3.9: The *Retrograde Canon* (bars 1–4), a palindromic canon belonging to the *Musikalisches Opfer* by J. S. Bach, and its reduction to first species counterpoint (unisons have been omitted).

Retrograde Canon			
c	$\mu(c)$	c	$\mu(c)$
$(0,0,1,0,1)$	2	$(1,0,0,0,1)$	2
$(0,0,2,0,0)$	8	$(1,0,1,0,0)$	43
$(0,1,0,0,1)$	2	$(1,0,1,1,0)$	1
$(0,1,1,0,0)$	43	$(1,1,0,0,0)$	14
$(0,1,1,1,0)$	1	$(1,1,0,1,0)$	3
$(0,2,0,0,0)$	11	$(2,0,0,0,0)$	11

Table 3.2: Complexity vectors of the *Retrograde Canon* and their occurrences.

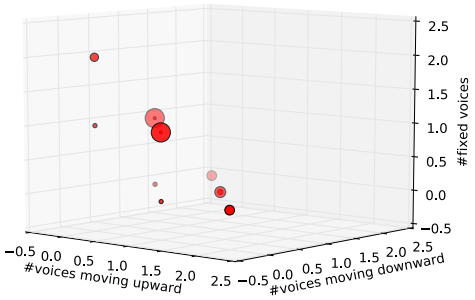
We homogenise the rhythm by expressing each note in eighths and we apply Algorithm 3.1. Here is the output of the first four meaningful voice leadings:

```
Voice Leading: ['D4', 'D4'] ['D4', 'F4']
c = [1, 0, 1, 0, 0] - oblique motion
Voice Leading: ['D4', 'F4'] ['F4', 'A4']
c = [2, 0, 0, 0, 0] - similar motion up
Voice Leading: ['F4', 'A4'] ['F4', 'D5']
c = [1, 0, 1, 0, 0] - oblique motion
Voice Leading: ['F4', 'D5'] ['A4', 'C#5']
c = [1, 1, 0, 0, 0] - contrary motion
```

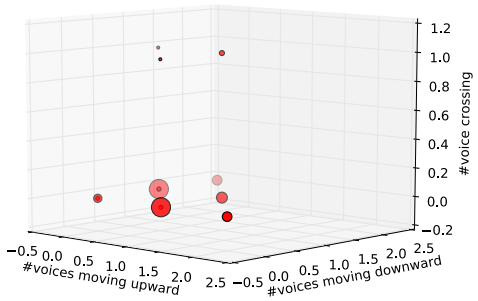
Table 3.2 collects the complexity vectors and their multiplicities; they are displayed in the form of point clouds in Figure 3.10.

3.7 Concatenation of voice leadings and time series

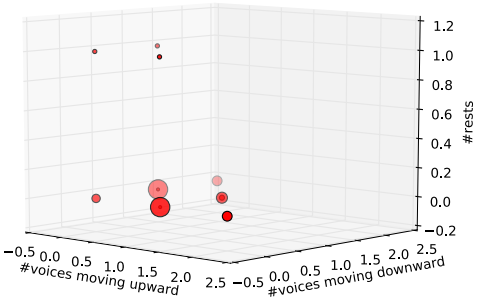
The paradigmatic point cloud associated with a voice leading gives a useful 3-dimensional representation of the piece; however, this analysis is just structural, as it does not take into account the way in which voice leadings have been concatenated by



(a) Projection on the first three components of the complexity vector.



(b) Projection on the *upward, downward* and *crossing* components of c.



(c) Projection on the *upward, downward* and *rest* components of c.

Figure 3.10: Three-dimensional projection of the 5-dimensional point cloud representing the complexity of the *Retrograde Canon*. The radius of each circle represents the normalised multiplicity of each complexity vector.

the composer. It is possible to introduce this temporal dimension by looking at the sequence of complexity vectors from a different viewpoint.

The concatenation of observations in time can be seen as a *time series*, that is a sequence of data concerning observations ordered according to time. In our case each piece of music can be described as a 5-dimensional time series, whose observations are the complexity vectors associated with each voice leading. More specifically, we use the so-called *Dynamic Time Warping (DTW)*, a method for comparing time-dependent sequences of different lengths: it returns a measure of similarity between two given sequences by "warping" them non-linearly (see Figure 3.11 for an intuitive representation) along the temporal axis. We invite the reader to consult Senin (2008) for a detailed review of DTW algorithms.

3.7.1 Dynamic Time Warping analysis

Let $\mathcal{F}$ be a set, called the *feature space*, and take two finite sequences $X := (x_1, \ldots, x_n)$ and $Y := (y_1, \ldots, y_m)$ of elements of $\mathcal{F}$, called *features* (here n and m are natural numbers). In order to compare them, we need to introduce a notion of distance between features, that is a map $\mathcal{C} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$, also called a *cost function*, that meets at least the following requirements:

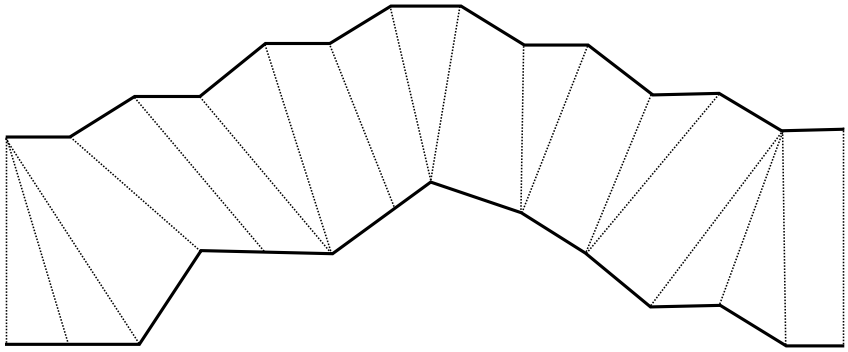


Figure 3.11: Dynamic Time Warping among two series of observation.

i. $\mathcal{C}(x, y) \geqslant 0$ for all $x, y \in \mathcal{F}$;

ii. $\mathcal{C}(x, y) = 0$ if and only if $x = y$;

iii. $\mathcal{C}(x, y) = \mathcal{C}(y, x)$ for all $x, y \in \mathcal{F}$.

Now, if we apply $\mathcal{C}$ to the features X and Y, we can arrange the values in an $n \times m$ real matrix $C \coloneqq (\mathcal{C}(x_i, y_j))$, where i ranges in $\{1, \ldots, n\}$ and j in $\{1, \ldots, m\}$.

A (n, m)-*warping path* in C is a finite sequence $\gamma \coloneqq (\gamma_1, \ldots, \gamma_l) \in \mathbb{R}^l$, with $l \in \mathbb{N}$, such that:

1. $\gamma_k \coloneqq (\gamma_k^x, \gamma_k^y) \in \{1, \ldots, n\} \times \{1, \ldots, m\}$ for all $k \in \{1, \ldots, l\}$;

2. $\gamma_1 \coloneqq (1, 1)$ and $\gamma_l \coloneqq (n, m)$;

3. $\gamma_k^x \leqslant \gamma_{k+1}^x$ and $\gamma_k^y \leqslant \gamma_{k+1}^y$ for all $k \in \{1, \ldots, l-1\}$;

4. $\gamma_{k+1} - \gamma_k \in \{(1, 0), (0, 1), (1, 1)\}$ for all $k \in \{1, \ldots, l-1\}$.

The *total cost* of a (n, m)-warping path γ over the features X and Y is defined as

$$\mathcal{C}_\gamma(X, Y) \coloneqq \sum_{k=1}^{l} \mathcal{C}(x_{\gamma_k^x}, y_{\gamma_k^y}).$$

An *optimal warping path* on X and Y is a warping path realising the minimum total cost (see Figure 3.12). We are now ready to define the *DTW distance* between X and Y:

$$DTW(X, Y) \coloneqq \min \{\, \mathcal{C}_\gamma(X, Y) \mid \gamma \text{ is a } (n, m)\text{-warping path} \,\}.$$

Remark 7. Note that the minimum always exists because the set is finite.

We computed the DTW distance between each pair of the three examples that we analysed in Subections 3.5 and 3.6.1, choosing as cost function the Euclidean distance in $\mathbb{R}^5$. We embedded the 4-dimensional complexity vectors in $\mathbb{R}^5$ by adding a fifth component and setting it to 0. The results of the comparison are shown in Table 3.3. Although we analysed only three compositions, it is possible to observe how the DTW distance segregates the two pieces belonging to the Chartres fragments.

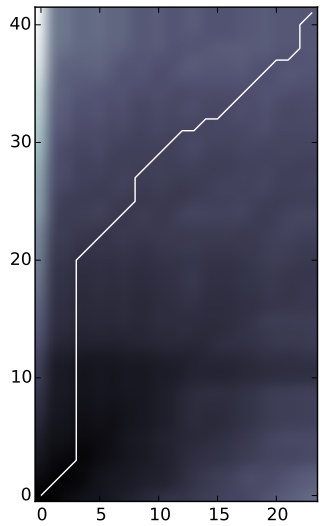


Figure 3.12: Optimal warping path on *Alleluia, Angelus Domini* and *Dicant nunc Judei*.

	Alleluia	*Dicant*	*Canon*
Alleluia	0.00	0.62	1.34
Dicant	0.62	0.00	1.16
Canon	1.34	1.16	0.00

Table 3.3: DTW distance matrix for the three time series of complexity vectors.

3.8 Discussion

Our analysis showed that our definition of complexity in terms of the relative movements of the voices and especially of crossing is suitable for characterising a musical piece. The point-cloud representation yields a "photograph" of complexity, a sort of fingerprint that lets clearly emerge the main features of the examined composition, noticeable even at first glance.

Albeit the extension of this method to the whole set of contrapuntal species forces a naïve simplification of the compositions, it provides a measure of their dissimilarity. The information collected by the time series of complexity vectors associated to a composition grabs the organisation of voice leadings in time, encoding the information concerning the motion of each voice, the overall configuration of the voices (relative motions and crossings) and the distribution of rests in time and among the parts. The DTW provides a direct measure of the dissimilarity between the complexity time series of two pieces. The optimal warping path points out the regions of the compositions that can be considered *comparable* with respect to the set of properties listed above.

Four

A braid-oriented visualisation of voice leadings

Partial permutations and their representation as low-dimensional sparse matrices are a handy computational tool to describe voice leadings, however they cannot provide a intuitive visualisation of the horizontal motions of the voices. Often describing voice leadings, musicologists refer to the superposed melodies naming them *strands* (Kurth and Rothfarb, 1991; Lester, 1994; Larson, 2012). Following this natural approach, we shall shortly define simultaneous melodic strands as a set of 3-dimensional paths, that is representative of a partial permutation. Furthermore, allowing these strands to intersect either at their starting or ending point, it is possible to visualise unisons. This comfortable representation of repeated pitches shall allow us to represent additional information such us leaps among voices. Finally, we take advantage of this representation, in order to tackle the problem of the interpretation of *voice leadings* between pitch-class sets.

4.1 The braid group and the partial singular braid monoid

In this section the basic background concerning braids and partial singular braids are recalled. Our main references are (Hansen, 1989) and (East, 2007, 2010).

4.1.1 The braid group

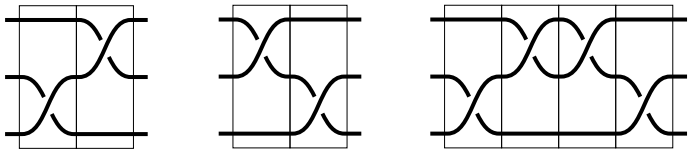
Definition 4.1.1. A *braid* β on n strands is a collection of embeddings

$$\mathscr{B} := \{\beta^\alpha : [0,1] \to \mathbb{R}^3,\ \alpha = \{1,\dots,n\}\},$$

with disjoint images such that:

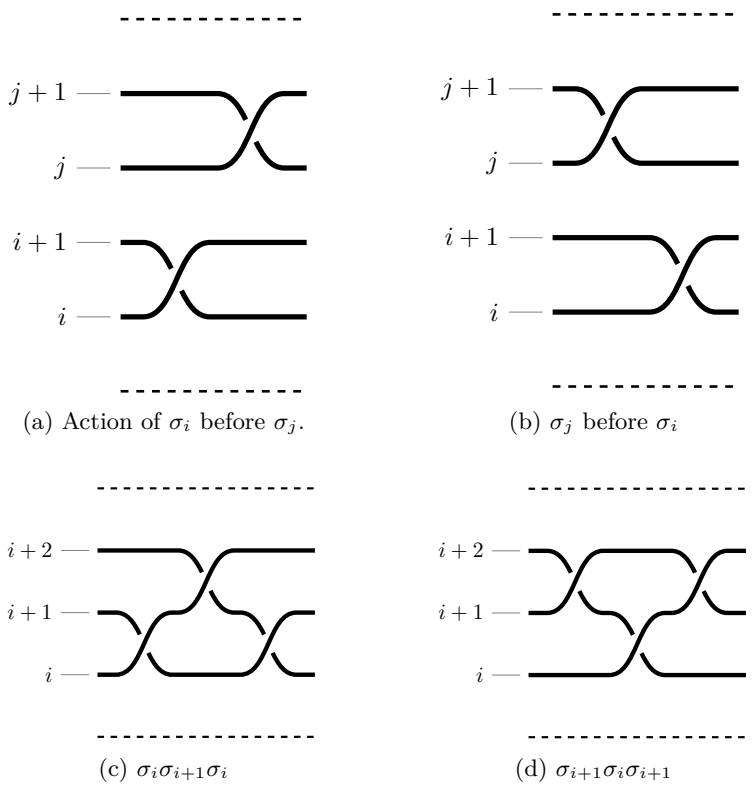
- $\beta^\alpha(0) = (0, \alpha, 0)$;
- $\beta^\alpha(1) = (1, \tau(\alpha), 0)$ for some permutation τ;
- the images of each β^α is transverse to all planes $\{x = const\}$.

Definition 4.1.2. Two braids are said to lie in the same *topological braid class* if they are homotopic relative to the endpoints in the sense of braids: one can deform a braid into the other without any intersection among the strands.



(a) First braid (b) Second braid (c) Their concatenation

Figure 4.1: Concatenation of braids.



(a) Action of σ_i before σ_j. (b) σ_j before σ_i

(c) $\sigma_i\sigma_{i+1}\sigma_i$ (d) $\sigma_{i+1}\sigma_i\sigma_{i+1}$

Figure 4.2: Graphical representation of the Braids properties in Equations (4.1.1) and (4.1.2).

There is a natural group structure on the space of topological braids with n strands, B_n, given by concatenation Figure 4.1. Using generators σ_i which interchanges the i-th and $(i+1)$-th strands with a positive crossing yields the presentation for B_n.

Let

$$p_1 : \quad \sigma_i\,\sigma_j = \sigma_j\,\sigma_i; \quad |i-j| > 1 \tag{4.1.1}$$

$$p_2 : \quad \sigma_i\,\sigma_{i+1}\,\sigma_i = \sigma_{i+1}\,\sigma_i\,\sigma_{i+1}; \quad i < n-1. \tag{4.1.2}$$

These two properties are depicted in Figure 4.2. We are ready to give a presentation of the group B_n as

$$B_n := \langle \sigma_1, \ldots, \sigma_{n-1} : \ p_1 \text{ and } p_2 \text{ hold} \rangle.$$

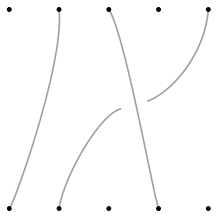


Figure 4.3: A partial braid $\beta \in \mathcal{IB}_5$

Let s_i be the permutation (i i+1), and the symmetric group $\mathcal{S}_n$ can be presented as

$$\langle s_1, \ldots, s_{n-1} \mid s_i s_j = s_j s_i \text{ for } |i-j| > 1,$$
$$s_i^2 = 1,\ s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1} \text{ for } 1 \leqslant i < n \,\rangle,$$

thus the projection defined on the group of braids on n strands, on the symmetric group is given by

$$\pi : \mathcal{B}_n \to \mathcal{S}_n$$
$$\sigma_i \mapsto (i\ i+1).$$

4.1.2 Partial braids and partial permutations

The braid group is too structured to represent generic voice leadings where voices can *collide* in unisons or can be rested. Our solution is to consider a weaker structure in which strands are endowed with higher degrees of freedom. Here, we introduce the inverse monoid of partial permutation and the partial singular braid monoid.

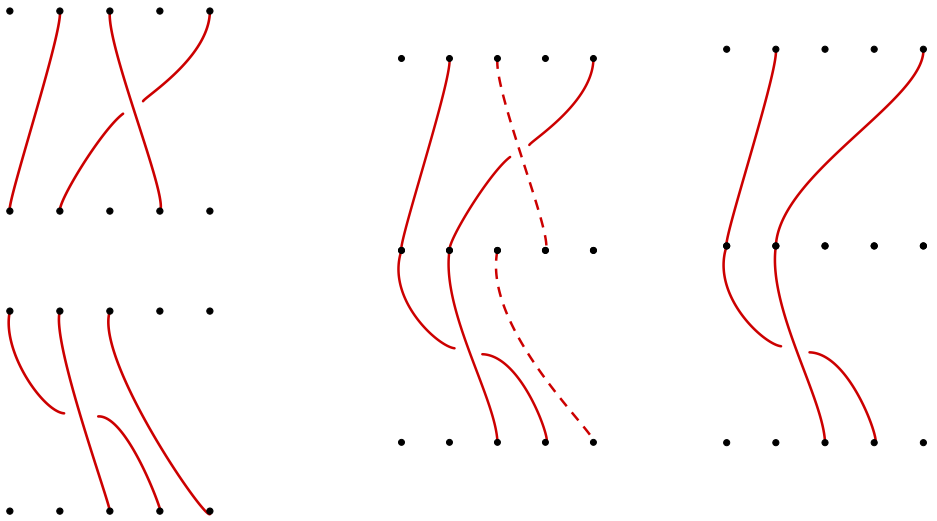
Definition 4.1.3 (Monoid)**.** A *monoid* is a couple $(S, +)$, where S is a set, $+$ is an associative binary operation and it exists $e \in S$ such that it is the identity element (for that operation).

We associate a monoid to a given set in the following way.

Definition 4.1.4 (Inverse monoid)**.** Given a set S the inverse monoid $\mathcal{I}_S$ is the set of all the partial bijections of S.

The braid inverse monoid $\mathcal{IB}_n$ introduced in (Easdown and Lavers, 2004) is the braid analogous of the symmetric inverse monoid $\mathcal{I}_n$ of partial permutations on n symbols. The monoid $\mathcal{IB}_n$ is the monoid of all homotopy classes of partial braids on n strands. A partial braid can be thought as a *full* braid $b \in \mathcal{B}_n$, with some strands removed. A representative of a partial braid in $\mathcal{IB}_5$ is depicted in Figure 4.3. There exists an epimorphism $\mathcal{IB}_n \to \mathcal{I}_n$, extending the projection $\pi : \mathcal{B}_n \to \Sigma_n$ defined above. The epimorphism

$$\pi^* : \mathcal{IB}_n \to \mathcal{I}_n$$
$$\beta \mapsto \overline{\beta}$$



(a) Two elements β and β_1 of $\mathcal{IB}_5$.

(b) Concatenation $\beta * \beta_1$.

(c) Removal of the unlinked strands.

Figure 4.4: Concatenation of partial braids in $\mathcal{IB}_5$.

is defined by construction. The partial braid depicted in Figure 4.3 is naturally associated to the partial permutation in $\mathcal{I}_5$, such that

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ \diamond & 1 & 4 & \diamond & 2 \end{pmatrix}.$$

The operation defined on $\mathcal{IB}_n$ is the multiplication of partial braids. Given two elements of $\{ \beta, \beta_1 \} \subset \mathcal{IB}_n$ (see Figure 4.4a), the multiplication of the two partial braids is depicted in Figure 4.4b and in Figure 4.4c. Observe that the string fragments which do not connect the upper plane to the lower plane are removed. See (East, 2007) for further details.

4.1.3 Singular braids

Singular braids are a generalisation of the standard braids, in which strands can intersect generating at most a finite number of singularities. Singular points have no inverses (Fenn and Keyman, 2000), then the set $\mathcal{S}B_n$ of singular braids on n strands is not endowed with a group structure, it is a monoid and it is possible to define it as follows.

Definition 4.1.5. $\mathcal{S}B_n$ is generated by $s_1, \ldots, s_{n-1}, s_1^{-1}, \ldots, s_{n-1}^{-1}, t_1, \ldots, t_{n-1}$, due to the relations

1. $\forall i < n,\ s_i s_i^{-1} = e$.

2. For $|i - j| > 1$ the compositions $s_i s_j$, $t_i t_j$ and $s_i t_j$ commute;

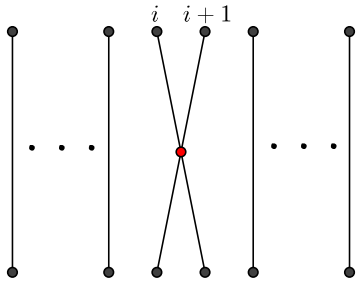


Figure 4.5: Singular generator of $\mathcal{S}B_n$.

3. $\forall i < n-1$

$$s_i s_{i+1} s_i = s_{i+1} s_i s_{i+1}$$
$$t_i s_{i+1} s_i = s_{i+1} s_i t_{i+1}$$
$$t_{i+1} s_i s_{i+1} = s_i s_{i+1} t_{i+1}.$$

Geometrically the singular generator has to be included, see Figure 4.5.

Musically, we shall only admit singularities representing unisons (multiple strands starting or ending at the same point).

4.1.4 Partial singular braids

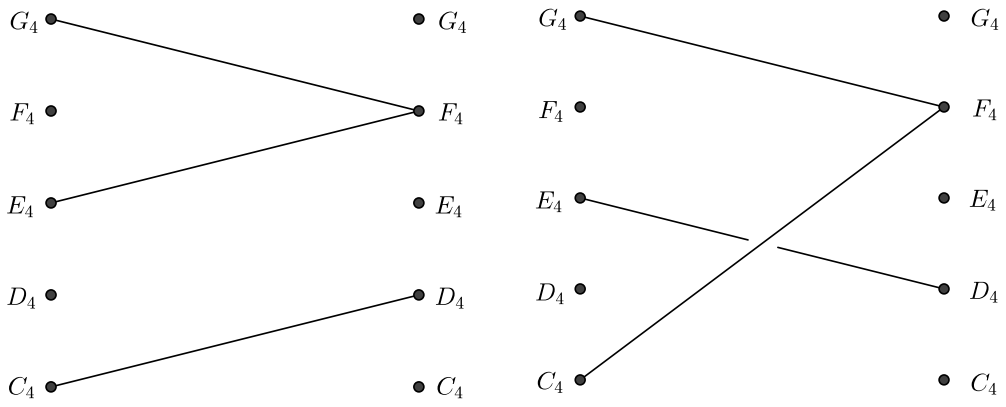
To model all of the possible simultaneous motions of voice leadings including rests and unisons, we need to consider the monoid of partial singular braids $\mathcal{PS}B_n$ containing both the partial and the singular braid monoid defined above. The theory about $\mathcal{PS}B_n$ will be sketched in this section, however we refer to (East, 2010) for a detailed analysis.

Consider the set $\{ 1, \ldots, n \}$ with $n \in \mathbb{N}$ and a singular braid $b \in \mathcal{S}B_n$. A partial singular braid β is obtainable by removing some strands of b. The removal of the whole set of strands is allowed in $\mathcal{PS}B_n$. In this particular situation β is said to be a sub-braid of b. A partial singular braid β induces a partial permutation $\bar{\beta} \in \mathcal{I}_n$ exactly as a partial braid does. Let $\{ \beta_1, \beta_2 \} \in \mathcal{PS}B_n$, the two partial singular braids are said to be equivalent if the partial permutations they induce are the same and if $\beta_1 \subset \gamma_1$ and $\beta_2 \subset \gamma_2$ with γ_1 and γ_2 singular braids equivalent in the sense of rigid-vertex-isotopy (Birman, 1993).

The multiplication of two partial singular braids follows the same rule we introduced for partial braids (see Figure 4.4). In particular denoting as $|\beta|$ the number of strings of β and as $N(\beta)$ the number of its singular points, the two submonoids of partial braids and singular braids of $\mathcal{PS}B_n$ can be described as $\mathcal{I}B_n = \{ \beta \in \mathcal{PS}B_n \mid N(\beta) = 0 \}$ and $\mathcal{S}B_n = \{ \beta \in \mathcal{PS}B_n \mid |\beta| = n \}$.

4.2 Modelling voice leading in $\mathcal{PS}B_n$

The first approach we describe is a mere translation in the braid formalism of the model described in Section 3.2. We define a voice leading of at most n voices as a



(a) Unison. (b) Unison and crossing.

Figure 4.6: Partial singular braid representation of voice leadings.

partial singular braid $\beta \in \mathcal{PS}B_m$, where m is the cardinality of the underlying set of the multiset $M \cup L$ as it has been introduced in Section 3.1.

These braids are better visualisable as *piecewise linear* braid diagrams, i. e. 2-dimensional projections of the partial singular braids, in which strands are depicted as line segments in $\mathbb{R}^2$. See Figure 4.6 for an exemple. This particular visualisation is suitable for representing simultaneous motions of voices, since the slope of each (projected) strand encodes the information concerning the movement (upward, downward) of each voice.

The introduction of singularities in correspondence of the starting and ending points of the braid diagram allows to simplify the model described in Section 3.2. Given a voice leading $v : M \to L$, where M and L are two multisets, we can represent the elements of $M \cup L$ labeled with the same symbol as a single point of the domain of the braid and encode their multiplicity as in the example of Figure 4.6a, where the voice leading

$$(C_4, E_4, G_4) \to (D_4, F_4, F_4)$$

is represented as a singular braid.

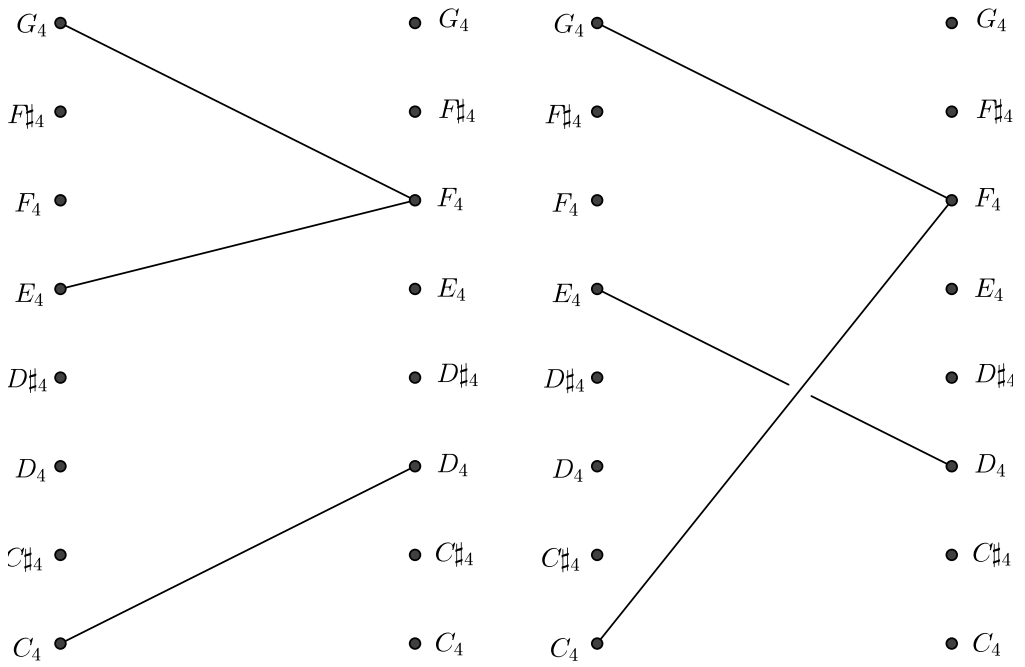
A crossing of the voices in $\mathcal{PS}B_n$ is represented as a crossing of the strands of the partial singular braid diagram. The voice leading

$$(C_4, E_4, G_4) \to (F_4, D_4, F_4)$$

represented in Figure 4.6b has a surprisingly clear musical interpretation: it contains a voice crossing corresponding to the crossing of the projected strands and it is *singular*, since two voices collapse on their target chord in a unison.

Let $s(\beta^\alpha)$ be the slope of the line segment β^α, and b^+, b^-, b^0 the number of strands of the piecewise braid with positive, negative and zero slope respectively; cr the number of crossings; $r = n - N(\beta)$, where n is the number of voices involved in the voice leading including rested ones; and finally $s = |\beta|$ the number of singularities. We can now rewrite the complexity vector associated to voice leadings in braid notation as

$$c = \left(b^+, b^-, b^0, \mathrm{cr}, r, s\right).$$



(a) Voices' leaps. (b) Voices' leaps and crossing.

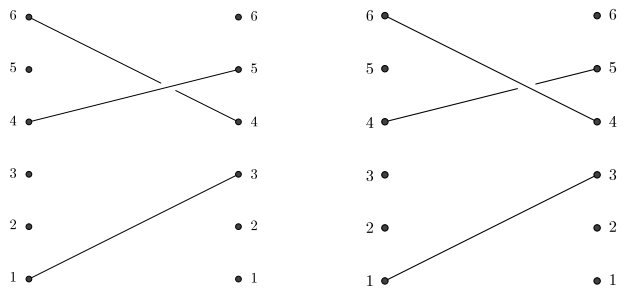
Figure 4.7: Partial singular braid representation of voices leaps.

4.2.1 Leaps

In Section 1.2, leaps were pointed out as an important feature to determine the *quality* of a voice leading. The partial permutation matrix model described in Section 3.2 does not take them into account. To do so, we should define a partial permutation whose domain has minimal cardinality equal to $n \times m$, where n is the number of voices involved in the voice leading and m is the number of half-steps from the lower to the higher pitch involved in the voice leading. For example, the representation of a voice leading between two 4-notes chords, ranging on 2 octaves, will require a partial permutation matrix of dimension 96.This is why in the partial permutation representation of voice leadings, we established a convention in order to manage repeated pitches and we build the model aiming at minimising the dimension of the matrix representing the voice leading[1].

Partial singular braids allow to represent leaps defining a dominion of cardinality m. The dominion of the braid consists in the pitches ranging from the minimum to the maximal pitch involved in the voice leading. Singularities can be used to represent repeated voices in a chord and the the slope of each strand corresponds univocally to a musical interval. In Figure 4.7, two piecewise linear singular partial braids describing the leap information are depicted. It is possible to store this information in the complexity vector either by writing explicitly the slope of each strand, or for instance, by splitting them into two classes of *consonant* and *dissonant*

[1]Considering a concatenation of voice leadings, to represent the whole counterpoint with matrices and to take into account intervallic leaps, it would be necessary to use matrices of maximal dimension, generating in the case of an orchestra a high-dimensional sparse representation.



(a) β_1 (b) β_2

Figure 4.8: Partial braids inducing the same partial permutation.

intervals (referring to the definitions given in Section 1.2).

Voice leading, partial singular braids and partial permutations

As we stated in the mathematical introduction of this section, there exists a natural projection $\pi^* : \mathcal{PS}B_n \to \mathcal{I}_n$. Thus, a class of partial singular braids β, in the sense of braid's homotopy (Definition 4.1.2) describes a particular partial permutation $\bar{\beta}$ on the elements of the dominion of β in an obvious sense. As it is shown in Figure 4.8 the braids β_1 and β_2 induce the same partial permutation represented by the cycle

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & \diamond & \diamond & 5 & \diamond & 4 \end{pmatrix}.$$

A crossing of two strands, both oriented from left to right is said *positive* if it corresponds to a positive braid generator. The particular choice of dealing with piecewise linear, positive, partial singular braid diagrams allows to associate a partial permutation to this particular class of braid diagrams and vice versa.

4.2.2 Partial singular braids on pitch classes

It is possible to model voice leadings modulo octave by considering the pitch-class space $\left(\mathbb{T}^1, \bar{d}\right)$, introduced in Section 2.2. In this case, the partial singular braids domain is given by the chromatic set of pitch classes

$$\{\,[C]\,,[C\sharp]\,,\ldots,[B]\,\} \cong \{\,[0]\,,[1]\,,\ldots,[11]\,\}$$

and the strands are defined on a cylinder $C = \mathbb{R}/12\mathbb{Z} \times [0, 2\pi]$. Following the piecewise linear approach that we used in the previous paragraphs, we assume the braid's strands to be lines wrapped around a cylinder i. e. geodesics on a cylinder, corresponding to helix segments parametrised as

$$\gamma : [0, 2\pi] \to \mathbb{R}^3$$
$$\gamma\,(t) = (\cos\,(at)\,,\sin\,(at)\,,t)\,.$$

Consider the voice leading

$$\mathcal{C}_1 = (C, E, E) \to \mathcal{C}_2 = (F\sharp, C\sharp, E)\,,$$

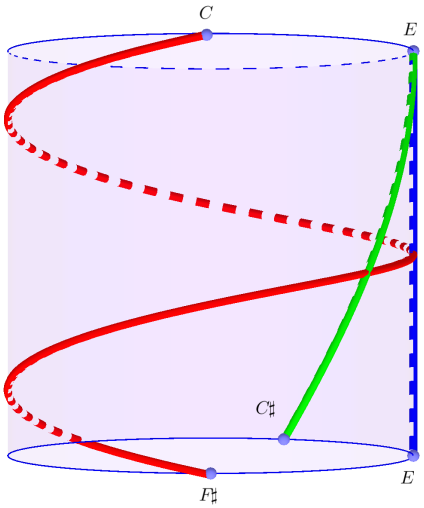


Figure 4.9: The partial singular braid representation of a voice leading defined in $\mathbb{R}/12\mathbb{Z}$.

depicted in Figure 4.9. Although the information concerning the octave is neglected, we can read in the image the measure of the leaps relative to each voice. The path connecting C to $F\sharp$ makes a complete round along the cylinder, meaning that the two notes are more than one octave apart. The singularity at E in the top face of the cylinder represents the unison or the doubling of the second and third voices of the chord $\mathcal{C}_1$. These doubled voices lead to $C\sharp$ and E respectively without octave leaps.

Simultaneous voice motions have an interesting representation in this context: taking into account the orientation of the wrapping (clockwise or counterclockwise/right-handed or left-handed) of the helix segments on the cylinder. In our example we can deduce that C and one of the E move downward to $F\sharp$ and $C\sharp$ respectively, while the last E is fixed since the trajectory is a straight line. Topologically, this information is encoded in the fundamental group of the cylinder C: $\pi_1(C) = \mathbb{Z}$ meaning that m positive or negative turns around the cylinder encode the octave leaps information.

Remark 8. Considering the class of partial singular braids with geodesics strands and positive crossings on pitch-classes, we cannot associate a unique braid to a partial permutation, in fact in this context geodesics are not unique, however it is always possible to consider minimal geodesics to represent strands among pitch classes, respecting the direction of the voice leading, if it is known *a priori*.

True and false crossings

The pitch class representation does not allow to distinguish among *true* and *false* voice crossings, unless the distance among the voices of the first chord is known *a priori*: a trajectory representing a leap of more than an octave and less than two, makes one complete turn around the cylinder, and hence crosses all the other strands involved in the voice leading, even if voices do not truly cross in musical sense.

In Figure 4.10 four different voice leadings among the 2-pitch class chords $\mathcal{C}_1 = (C, E)$ and $\mathcal{C}_2 = (D, F)$ are depicted[2]. As we stated few lines above, the lack on information given by the identification of the octaves does not allow to distinguish among true or false voice crossings, as it can be shown by analysing the four voice leading represented in Figure 4.10:

a) In Figure 4.10a the two strands do not make a complete tour of the cylinder and do not cross, meaning that the target pitch-classes lie in the same octave of the pitch-classes of the first chord and that there is no *topological* and musical crossing among the voices.

b) Figure 4.10b shows the *crossed* alternative of the previous voice leading $\mathcal{C}_1 \to \sigma_{12}\mathcal{C}_2$, where $\sigma_{12} \in \mathcal{S}_2$. Since the helix segments are left-handed and right-handed respectively and the strands of the braid do not complete the tour of the cylinder, what we can deduce from this configuration is that F lies in the same octave as C and symmetrically D lies in the same as E. To establish if the voices cross in a musical sense mirroring the strands' crossing, it is necessary to know the distance among the voices of the first chord: assuming C and E to belong to the same octave, the voices actually cross. However, if the two notes belong to two different octaves, for instance C_4 and E_5, no crossing occurs among them.

c) In Figure 4.10c, C and E moves downward to reach F and D respectively and both voices move in contrary motion of less than one octave. No crossing can occur among these voices as it is mirrored by the trajectories of the braid.

d) In the last figure, voices move in contrary motion, downward and upward respectively always targeting pitch-classes less than one octave distant from them. If the pitch classes of the first chord lie in the same octave, then no music crossing occurs, despite the topological configuration of the braid's strands, however it suffices to choose the representative of C and E to be C_4 and E_3 to have an actual crossing corresponding to the one depicted on the figure.

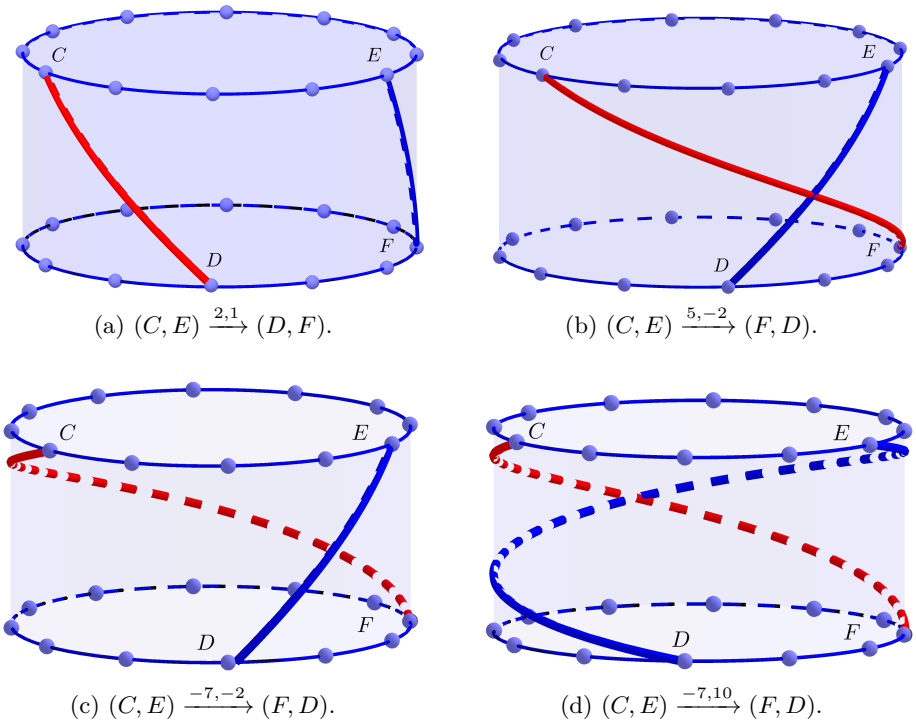
In conclusion, the analysis of voice leadings between pitch-class sets gives a representation of the no-crossing voice leading as the collection of shortest paths among multisets of pitch-classes[3] and maximize the number of crossings in the other cases. Thus, the pitch-class braid-oriented visualisation of voice leadings collects the information concerning both octave leaps and voice crossings as they are described in Hughes (2015), where a model of voice leading built on the fundamental groupoid of the chord space $\mathbb{A}_n$, is discussed.

4.2.3 Concatenation of voice leadings in $\mathcal{PS}B_n$

As we point out in Section 3.3, representing several ordered voice leadings, it is not desirable to compose the braids representing each of them, but to concatenate them one after the other: the composition $\mathcal{PS}B_n$ inherits from $\mathcal{P}B_n$ imposes to delete

[2] see (Tymoczko, 2011, p. 76) for a representation of the same voice leadings in $\mathbb{A}_2$.

[3] Hence, neglecting the order in which voices are associated, we can always retrieve a no-crossing voice leading connecting the voices of the first chord to the ones of the second through minimal geodesics on the cylinder, as in Figure 4.10a.



(a) $(C, E) \xrightarrow{2,1} (D, F)$. (b) $(C, E) \xrightarrow{5,-2} (F, D)$.

(c) $(C, E) \xrightarrow{-7,-2} (F, D)$. (d) $(C, E) \xrightarrow{-7,10} (F, D)$.

Figure 4.10: Simultaneous motions of two voices. The movement of each voice is written above the arrow in half-steps, the sign distinguish among upward and downward movements.

strand fragments not connecting the first braid to the second, see Figure 4.4. Thus, using the multiplication defined on $\mathcal{PS}B_n$ to compose braids, one would delete the strands representing any melody containing a rest, losing the information concerning the whole piece of music.

The idea is to represent a succession of voice leadings as a time series $\{ \beta_i \}_{i \in \{ 1,\dots,n \}}$, such that $\beta_i \in \mathcal{PS}B_n$ for each i, corresponding to a concatenation of braids as it is shown in Figure 4.11 where both the pitches and pitch classes braids for the first seven voice leadings (corresponding to eight melodic states) of *Alleulia: Angelus Domini* are depicted. The fragment we analyse is given by the superposition of the two voices

$$\begin{aligned} v_1 &= (F_4, G_4, A_4, G_4, F_4, G_4, B\flat_4, A_4) \\ v_2 &= (C_4, D_4, E_4, F_4, G_4, G_4, F_4, E_4)\,, \end{aligned}$$

represented by the blue and red trajectory respectively.

In this case, being the pitches involved in the segment of the composition we represented contained in a octave, the pitch and the pitch class diagram are equivalent. It is possible to observe how this kind of representation gives a friendly access to the information describing the simultaneous motion of voices. It retrieves the special case of parallel motion, voice crossings and unisons are represented by strands crossings and singularities, respectively. Observe, that in this case, the concatenation of partial

singular braids corresponds to the multiplication defined in $\mathcal{PS}B_n$, since no rests are included in the passage we examined.

Given a sequence of voice leadings $\{\beta_1, \ldots, \beta_n\}$, a motif is a subsequence $\{\beta_p, \ldots, \beta_q\}$ with $1 \leqslant p < q \leqslant n$. This last representation allows to compare voice leading *motifs* at first sight (consider, for instance, the crossing pattern in Figure 4.11). It provides a possible solution for the evaluation of their features, according to both their geometry and concatenation in time. The advantage of this braid representation, is the possibility to encode the whole information concerning the voice leading in a three dimensional braid, despite the number of voices composing it.

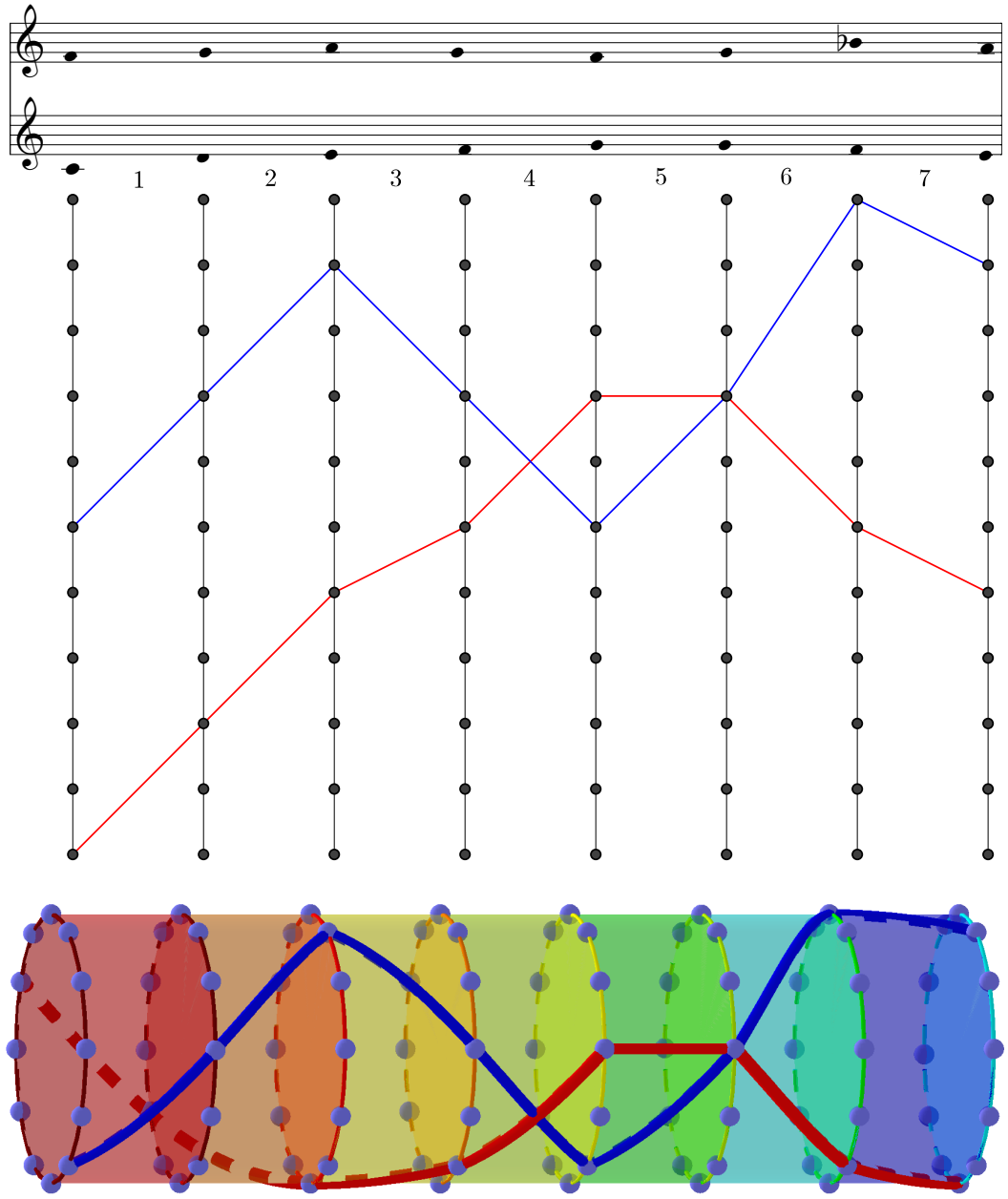
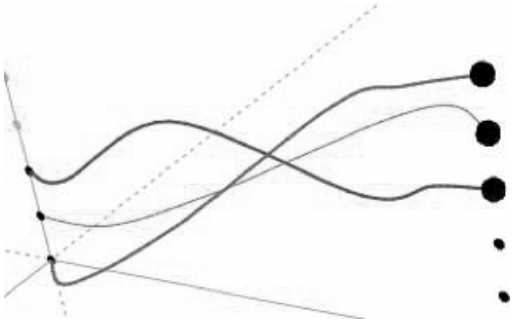


Figure 4.11: Concatenation of pitch and pitch-class partial singular braids.The observation of a single strand, or of the whole voice leading (regions $1, \ldots, 7$) provide an intuitive representation of both the motions of pairs of voices (similar, parallel, oblique, contrary) and of the behaviour of each voice (downward, upward and fixed). The length of a crossing is simply measurable, as more complicated phenomena such as the overlap (see Section 1.2).

Five

Discussion and future works



Thanks to the mathematical formalisation of the concept of voice leading, we deduced a model to describe voices' motions as low dimensional partial permutation matrices. The geodesics-oriented interpretation of voice leadings and the analysis of their concatenation, provided a relationship between the algebraic and the geometric context. Thereafter, the information carried by the partial permutation matrix associated to a voice leading has been rewritten under the form of complexity vector. Sequences of such vectors have been used to characterise the counterpoint of first specie as a multiset of 4-dimensional points.

We proposed a generalisation of the model to the other contrapuntal species and the sequence of complexity vectors has been interpreted as a multi-dimensional time series, describing how different kind of voice leadings has been concatenated in time. Dynamic time warping provides a measure of the distance between the complexities of pairs of compositions, and gives a quantitative description of the dissimilarity of their time-series representation.

In order to visualise voice leadings in a 3-dimensional space, a braid-like representation has been introduced allowing to extend the first model, measuring the intervallic leap of each voice in the passage from a chord to the next one. In addition, a pitch-class version of the braid representation provides an environment to visualise voice leadings among n-chords, mirroring the properties of trajectories in the space of chords.

A straightforward development offered by the model we describe is the possibility to classify the collection of possible voice leadings among two chords in terms of the length of the geodesic strands of the braid representing it. Connecting the notes of two chords with minimal paths corresponds to a crossings-free voice leading. The variations of this configuration could be classified considering the length of each strand of the braid associated to the voice leading.

In addition, the model we introduced as a visualisation tool has topological properties that could be investigated, for instance in terms of knot theory (Alexander, 1923, 1928). To do that one should weaken our assumption on the crossings among the strands. A possible definition could involve, for piecewise braids, the slope of the line segment describing the voices, forcing a strand associated to a bigger leap to pass above the others.

Part III

The vertical dynamics of music: persistent musical features

Table of Contents

Abstract

In contraposition to the analysis of voice leadings as superposition of horizontal melodies, this part is devoted to the study of music as composed by vertical structures. Albeit triads, seventh chords and their altered forms represent the basic vertical objects in western music, every melodic interval (*arpeggio*) can be notated and thought *vertically* as the harmonic superposition of its note. In Figure 5.1 both the melodic and harmonic representations of a major third interval, a major seven chord, an altered chord and a whole scale are depicted in consecutive bars. This double vision of scales and chords as *arpeggi* and clusters inspired several techniques such as *broken chords* (Pass, 1987, p. 3) to enrich the harmonic and melodic playing of guitar with uncommon phrases built breaking a chord in smaller clusters. Neglecting the notion of voices, but taking into account only pitches (or pitch-classes) and durations of the notes, it is possible to produce an efficient and simple musical representation, able to grasp a compositional idea, that is repeated in a composition and hence, represents its core.

In Chapter 6 an early approach aiming at describing music (concepts) through the deformation of a topological space is described. The vertices of the *Tonnetz* are endowed with a variable height, respect to particular choices of pitch classes and durations of a sequence of notes. This representation inspired a second approach, whose mathematical bases are introduced in Chapter 7. The topological theory described in this chapter provides the tools we shall employ to interpret the musical information, once it is represented in the geometrical and topological structure of the deformed *Tonnetz*. The applications to music analysis and classification and the results obtained through these strategies are described in Chapters 8 and 9. The former takes place in the symbolical domain and the latter is positioned at the crossroad between signal and symbols (this last part has been the subject of the talk (Bergomi, 2015)).

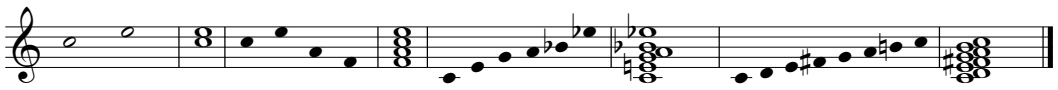


Figure 5.1: Melodic and harmonic intervals. Pairs of consecutive bars represent different musical entities from a melodic and a harmonic viewpoint, respectively.

Six

Music analysis through deformations of the *Tonnetz*

One of the most common features of geometric spaces for music analysis is their isotropic nature. Indeed, the pitch-class space $(\mathbb{R}/12\mathbb{Z}, \overline{d})$ is usually represented as a cyclic graph where the 12 vertices representing the pitch classes are evenly spaced (we depicted this representation in Figure 2.4b). This feature reflects the equality of pitches in equal tuning, as in the representation of the pitch classes as an abstract graph a weight equal to 1 is naturally associated to the graph's vertices. Here, we seek a strategy to associate to these vertices a collection of musically relevant weights, in order to produce an intuitive and analysable geometrical representation of a music piece.

When dealing with Western Music and in particular with modern Music it is natural to reduce the possible temperaments to the equal tuning and hence to develop models based on these evenly spaced representations and unweighted graphs. This homogeneity is unavoidably inherited by spaces generated through the identification of notes modulo octave. Here, the main idea is to develop a strategy to introduce *preferred* directions in these spaces. By preferred directions we mean a change in the geometry of the space, encoding relevant musical information. For instance, we shall represent a relevant pitch-class set on the realisation of the *Tonnetz* as a mountain ridge, highlighting the relevance of that particular choice of pitch classes respect to the others.

In musical terms, these preferred directions can be thought as the core concepts representing the main musical ideas of a composition. Often, a concept sprouts in the mind of a composer as a small cell generally based on a precise rhythmical idea, a sequence of pitches, a harmonic pattern, or their combination. The original idea is then varied, merged with new ones, stretched or condensed and maybe left for a completely new one.

Often, these musical variations are conceived to accompany the listener and are well codified, as it is shown by the vast literature concerning the study of melodic variations and the analysis of motifs. See for instance (Piston, 1947; Dudeque, 2005; Johnson, 2009) for a musical theoretical point of view on this subject and (Lewin, 2007; Buteau and Mazzola, 2000) and many others for a mathematical-oriented viewpoint. In (Dowling, 1972) the perception of the inversion and the retrogradation of a melody has been investigated, showing that it is grasped and understood by the listener. Being such variations relevant to the humans' perception of music, our claim is that their representation as deformations of a metric space would grab this

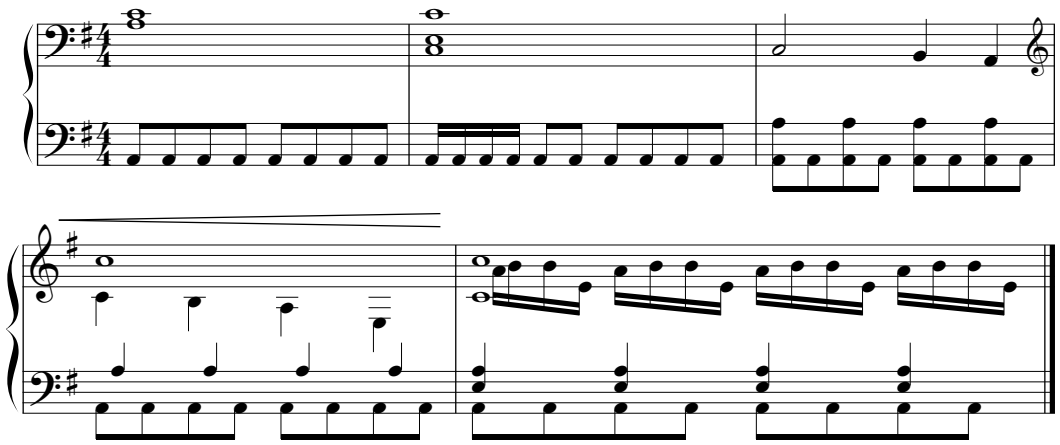


Figure 6.1: The musical concept evolution in *Time* by Hans Zimmer. The first bar represents the musical idea that opens the composition. The following bars depicts consecutive evolutions of the first concept.

fundamental information. Once constructed, such a space shall give an immediate visual feedback and represent this information in geometrical and topological terms.

Figure 6.1 shows how the musical concept described in the first bar of *Time* by *Hans Zimmer* evolves during the piece. Each bar describes one of its variations (see Appendix D for the whole partition, as it has been arranged for piano by Sebastian Wolff). The A minor triad suggested by a minor third interval in the first bar, appears as a whole in the second one. On the rhythmical side, always in the second bar, the bass' *clave* changes doubling the rhythmical figure. Then a new melodic idea is introduced: the second of the chord is added to enlarge the sonority of the triad. In the fourth variation, an indication concerning the dynamics of the phrase is introduced. Finally, a new melodic concept is added in the fifth bar. Observe how these variations of the first idea allow to declare an initial context, describe it and later enrich it with details and dynamics. The musical concept that is implied in these variations shall represent our *preferred directions* in music.

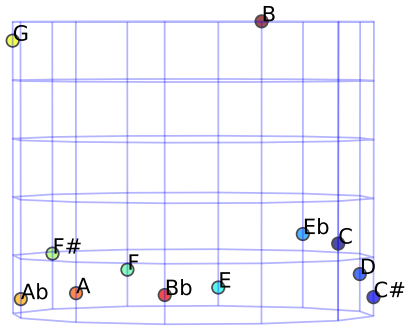


Figure 6.2: A displacement of the vertices of $|\mathbb{Z}/12\mathbb{Z}|$ according to the occurrences of their labels in the second movement of *Shönberg* Op. 19.

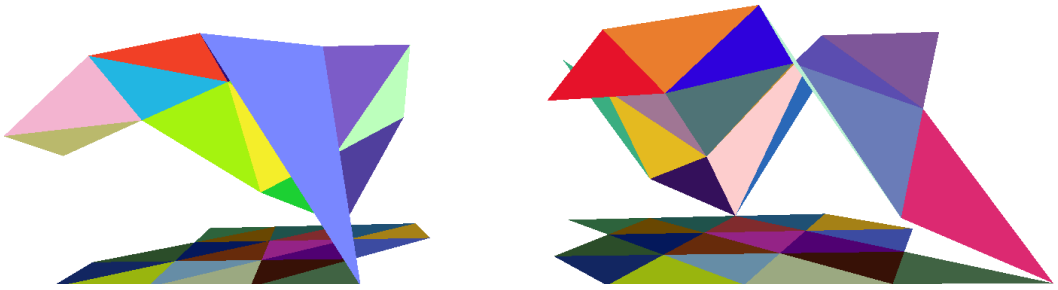


Figure 6.3: Deformed geometries generated from the *Tonnetz*. A portion of the planar *Tonnetz* is represented on the plane $z = 0$.

6.1 An anisotropic *Tonnetz* for music analysis

Consider the graph describing pitch classes and assume to weight the vertices counting the occurrences of each pitch class in a piece of music. Equivalently, it is possible to define a function on the set of vertices of the space $(\mathbb{R}/12\mathbb{Z}, \bar{d})$, say $h : V \to \mathbb{R}$ associating to each vertex a height corresponding to the number of occurrences of its label in the piece. Hence, the function h is defined as the composition

$$h : V \xrightarrow{\mathrm{l}} L \xrightarrow{\mathrm{c}} \mathbb{R},$$

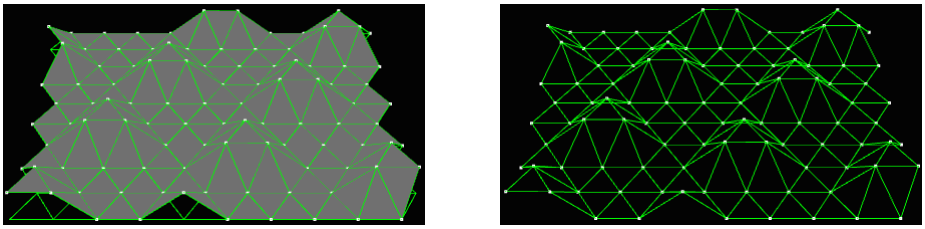
where l associates vertices to pitch classes and $c : L \to \mathbb{R}$ is the function counting the pitch classes' occurrences.

Consider the cylinder $\mathbb{R}/12\mathbb{Z} \times [0, 1]$. It suffices to redefine the position of the vertices as $(x_v, y_v, h(v)) \in \mathbb{R}^3$ to obtain the space represented in Figure 6.2. Such a space does not differ from a common pitch-class histogram (Six and Cornelis, 2012), the only additional information retrieved by this representation is due to the structure of the graph. Pitch classes a half-step apart are connected by an edge. This consideration suggests the possibility to take advantage of the structure of the graphs that already proved their efficacy in Music analysis.

Dealing with a more structured graph as the *Tonnetz* (see Section 2.3), one has the possibility to take advantage of the symbolic and acoustical properties it is endowed with. In particular, we recall that interpreting the *Tonnetz* as a simplicial complex, its edges are associated to precise intervals and triangles represent triads. Moreover, it was conceived to represent the acoustical relationships among pitch classes. This whole structure is preserved, when updating the height of the vertices of its geometric realisation.

The *Tonnetz* has already been used to classify *genres* in (Bigo et al., 2013) analyzing the compactness of the simplicial structures representing the trace of a piece of music on different planar *Tonnetze*. As it has been shown in Section 2.3.2 the subcomplexes generated as a planar trace on the *Tonnetz* do not distinguish scales or sonorities in a geometrical and topological sense.

In order to capture the *temporal* and harmonic information, the vertices shall be displaced depending on the pitch which is played and on its duration. The reason why we use only three dimensions to encode these two features is to have the possibility to visualise the surface generated through these displacements, and provide a direct visual feedback, as it is depicted in Figure 6.3.



(a) $\mathcal{T}$ (b) 1-skeleton

Figure 6.4: The *Tonnetz* deformed with a major triad and its 1-skeleton. The triad appears as a maximal triangle with respect to the height function.

6.1.1 A variable geometry, 3-dimensional *Tonnetz*

Let T be the infinite planar simplicial *Tonnetz* and $|T| \subset \mathbb{R}^3$ its geometrical realisation (see Sections 2.1 and 2.3). Given a note n, let p be its pitch and d its duration. The vertices of T labelled with the pitch class $[p]$ are translated in $\mathbb{R}^3$ along the z-axis direction of a distance d.

In symbols, let V be the 0-skeleton of $|T|$ and $l : V \to L$ the function associating vertices to labels. Let

$$V_{[p]} = \{\, v \in V \;\mid\; l\,(v) = [p] \,\}$$

be the set of vertices of $|T|$ labelled with $[p]$. The updating of the height of the vertices corresponding to a certain pitch-class is provided by a family of functions $\{h_{[p]}\}_{[p]\in L}$ defined as

$$\begin{aligned} h_{[p]} : V_{[p]} &\to \mathbb{R}^3 \\ (x_v, y_v, z_v) &\mapsto (x_v, y_v, z_v + d) \end{aligned} \tag{6.1.1}$$

for every $[p] \in \mathbb{Z}/12\mathbb{Z}$. Considering a collection of notes

$$\{n_1, \ldots, n_m\} = \{(p_1, d_1), \ldots, (p_n, d_n)\},$$

the vertices of the *Tonnetz* labelled with the pitch class $[p]$ will be translated vertically of the value corresponding to the sum of the durations of the notes n_i such that $p_i = [p] \mod 12$, for $1 \leqslant i \leqslant n$. We refer to the geometric realisation of the *Tonnetz* in a deformed state, i. e. when at least one of its vertices does not lie in the plane $z = 0$, denoting it as $\mathcal{T}$. In Figure 6.4 deformation $\mathcal{T}$ induced by a major triad played for 8 seconds is depicted.

A 3-dimensional interactive animation showing how the *Tonnetz* is deformed by a musical phrase and allowing the user to play with its own keyboard to generate specific deformations is available at `http://nami-lab.com/tonnetz/examples/deformed_tonnetz_int_sound_pers.html`. The Javascript and html code are also available on the web. See Appendix C for a commented version of the code generating the animation and a brief tutorial concerning its functions. The translation of the vertices is rendered as a continuous displacement in time, at a constant speed. The resulting shape after the deformation is equivalent to the one generated through the collection of functions defined in Equation (6.1.1).

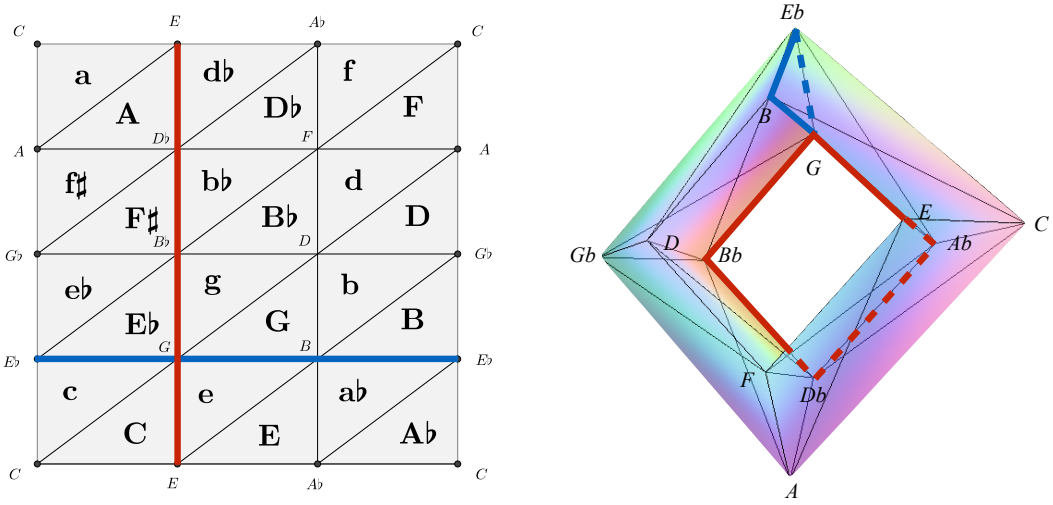


Figure 6.5: A vertex map from the fundamental domain of the *Tonnetz* to the *Tonnetz* torus. The red and blue lines corresponds to the two generators of the torus, given by the translation (transposition) of 3 and 4 half-steps, respectively.

6.1.2 Preferred directions in music: a naïve approach

Before using persistent homology to classify different configurations of the *Tonnetz*, we describe the first approach we followed and consequently a first simple music representation strategy. This trial is based on the definition of a *preferred subcomplex* of T induced by the values of the height function on the vertices of its deformed geometrical realisation.

Consider the simplicial complex $\mathcal{T}$ derived by deforming the heights of the vertices of $|T|$ with a sequence of notes. Let

$$f_V : V \subset \mathcal{T} \to \mathbb{R}$$
$$(x, y, z) \to z.$$

be the height function, V the 0-skeleton of $\mathcal{T}$ and $m = \max\{f(v) \mid v \in V\}$. The label of the vertex of maximal height is given by $\{l(v) \mid f_V(v) = m\}$ for $v \in V$. Let $V_t \subset V$ be

$$V_t = \{\, v \in V \;\mid\; f_V(v) > m/t \,\}$$

the collection of vertices whose height is greater than the threshold $m/t \in \mathbb{R}$. The *preferred pitch class set* $\mathcal{P}$ is constituted by the pitch classes labelling the vertices of V_t.

This set has infinite cardinality since any label is associated with infinite vertices of T. The labelling of the Tonnetz is double periodic with respect to both the translations of major and minor third intervals. We can restrict our analysis to the fundamental domain $F \subset T$ generated by the translation corresponding to these intervals. See Figure 6.5.

To map the square F in the *Tonnetz* torus $\mathbb{T}$ depicted in Figure 6.5, it is necessary to introduce some definitions. Let K be a simplicial complex.

Definition 6.1.1. A point $x \in |K|$ belongs to the interior of exactly one simplex of $\sigma \in K$. Assume $\sigma = [v_0, \ldots, v_n]$. Then

$$x = \sum_{i=0}^{n} \lambda_i v_i,$$

where $\lambda_i > 0$ for each i and $\sum_{i=0}^{n} \lambda_i = 1$. Let v be an arbitrary vertex of K, the *barycentric coordinates* $b_v(x)$ of x with respect to v are defined by setting

$$b_v(x) = \begin{cases} \lambda_i & \text{if } v = v_i \text{ for } 0 \leqslant i \leqslant n \\ 0 & \text{otherwise} \end{cases}.$$

The following lemma (Munkres, 1984, Ch. 2, Lemma 2.7) allows to extend a map defined on the 0-skeletons of two simplicial complexes to their entire structures.

Lemma 6.1.1. *Let $\varphi : K_V \to L_V$ a map between the 0-skeletons of the complexes K and L. Suppose that whenever the vertices $v_0, \ldots, v_n$ of K span a simplex of K, their images $f(v_0), \ldots, f(v_n)$ are vertices of a simplex of L. Then φ can be extended to a continuous map $\Phi : |K| \to |L|$, such that*

$$x = \sum_{i=0}^{n} \lambda_i v_i \Rightarrow \Phi(x) = \sum_{i=0}^{n} \lambda_i \varphi(v_i).$$

The map Φ is called the linear simplicial map *induced by the vertex map φ.*

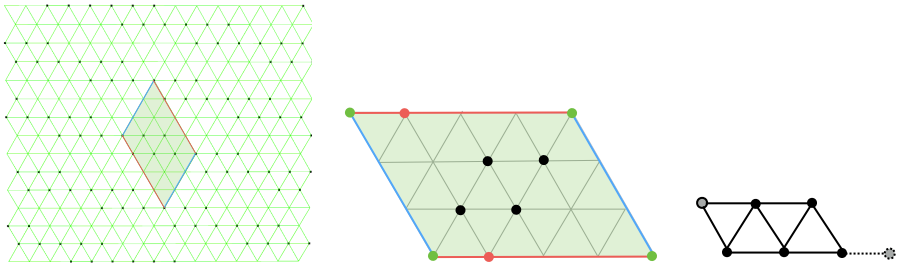
In our case, the enharmonic labelling of the *Tonnetz* allows to define a vertex map and its extension to a simplicial map $\Phi : F \to \mathbb{T}$. Consider the subcomplex $S \subset F$ given by the simplices of F whose vertices are elements of V_P. We obtain a subcomplex of $\Phi(S) \subset \mathbb{T}$, by identifying the simplices of S lying on opposite edges of F. In Figure 6.6 we show how this subcomplex can be computed in the case of Ravel's *Jeux d'Eau* and setting $t = 2$ in three steps:

1. Figure 6.6a represents the projection of the preferred set of vertices on $|T|$. The darker parallelogram is the fundamental domain F visualised as a region of $|T|$.

2. The same region of the space is depicted in Figure 6.6b, where different colours highlight edges and points to be identified.

3. The last figure represents the subcomplex $S \subset \mathbb{T}$ generated by the vertices labelled with preferred pitch classes.

In geometrical terms, to consider the threshold t corresponds to slice $\mathcal{T}$, in order to retrieve the vertices belonging to level-set $f^{-1}([m/2, m])$. In musical terms, this operation allows to segregate the pitch classes that we heard at least m/t seconds during the piece. In addition, relevant pitch-class sets endowed with the structures of triads and consonant intervals (perfect fifths, major and minor thirds and their inversions), corresponds to simplices of S. It is also possible to retrieve the absence of preferred musical entities if every pitch is a preferred one, and hence $S = \mathbb{T}$.



(a) Fundamental domain. (b) Preferred pitch class set (c) Preferred pitch classes subgraph

Figure 6.6: Deriving a preferred pitch class set for *Jeau d'Eau*.

Sometimes we shall refer to the subcomplex S, or its equivalent that will be defined in the remainder of this work, as the subcomplex of *preferred directions* associated to a musical phrase or to a composition. This expression refers to the musical preferred directions (pitch-class sets corresponding to intervals, triads, or the whole chromatic scale) highlighted by the height function.

Remark 9. In our analyses we consider only the Euler *Tonnetz*. However, this approach and in general the whole set of strategies we shall describe in this part are suitable for the analysis of other types of *Tonnetze* (Cohn, 2011; Bigo et al., 2013).

Application

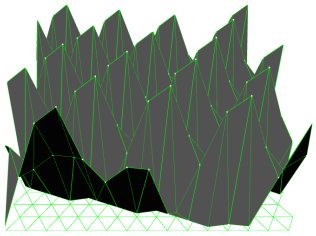
The structure of the subcomplex of preferred directions associated to a composition can be used in order to classify them. It is important to notice that in order to do that, no labelling is needed in order to perform the analysis.

In Figure 6.7 the shapes associated to several music pieces are depicted along with their preferred subcomplexes. The threshold we considered for the examples presented in this paragraph is $t = 2$. It is possible to see how tonal pieces are associated to subcomplexes representing a selection of major and minor triads along the fifth axis of the *Tonnetz* (horizontal edges). *Jeux d'Eau* is described by a suite of three pitch classes a perfect fifth a part, or equivalently three triads composed by six distinct notes corresponding to a diatonic scale minus its fourth degree (see Figure 6.6c). The first movement of Mozart's Sonata no. 8, is represented by two consecutive triads a perfect fifth apart. The subcomplex associated to the first movement of Beethoven's Sonata no. 13 is also developed on the fifth's axis of the *Tonnetz*, where a minor triad represented by the only 2-simplex in Figure 6.7d is enriched with a perfect fourth, evoking a pentatonic sonority.

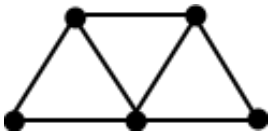
Figures 6.7h and 6.7j represent *Klavierstük I* and a sequence of random pitches, respectively. Note that the first subcomplex includes 11 of the 12 vertices (modulo identifications) of F, while the collection of randomised pitches includes the whole set of pitch classes. The analysis of the third piece of the Schönberg's work reveals exactly the same preferred pitch-class subcomplex, we identified for the first one. On the contrary, the second piece has a different structure. The subcomplex depicted in Figure 6.7e differs both from the representations we found for random sequences of pitches and from the tonal ones. In this case, a minor triad is preferred together

with its major sixth and minor second, which could be considered a modal, rather then a tonal or a chromatic choice.

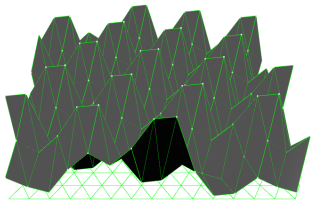
Remark 10. The results presented in this section can be replicated on the web application. The set of preferred vertices can be visualised on $|T|$ by clicking on the button *dips_pref*, after the *Tonnetz* has been deformed. The information concerning the labels of the height of the vertices is displayed in the JavaScript console.



(a) Sonata n. 8, mov. 1- Mozart.

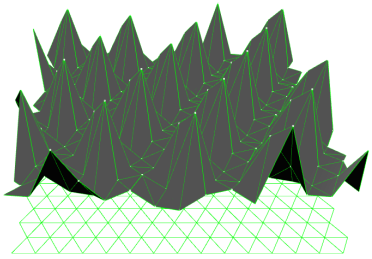


(b) Preferred pitch class subcomplex.

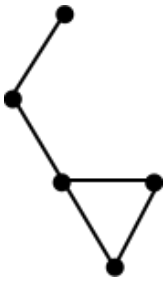


(c) Sonata in C major, mov. 1 - Beethoven.

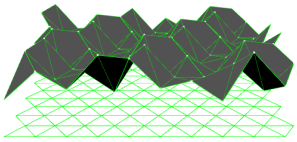
(d) Preferred pitch class subcomplex.



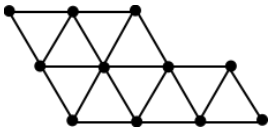
(e) Klavierstück I - Schönberg.



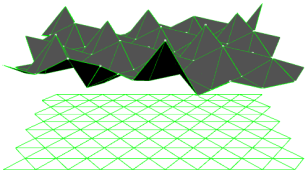
(f) Subcomplex.



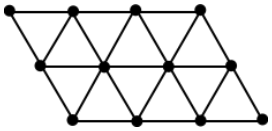
(g) Klavierstück II - Schönberg.



(h) Preferred pitch class subcomplex.

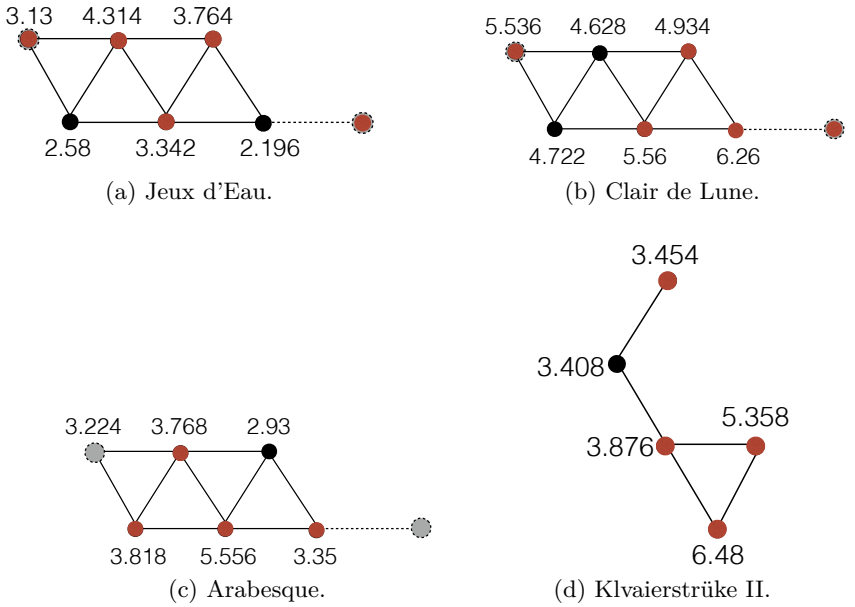


(i) Random Pitches.



(j) Preferred pitch class subcomplex.

Figure 6.7: Deformed *Tonnetze* and their preferred subcomplexes. Identifications are omitted for clarity.



(a) Jeux d'Eau. (b) Clair de Lune.

(c) Arabesque. (d) Klvaierstrüke II.

Figure 6.8: Weighted preferred subcomplexes of T.

6.2 Towards a topological classification of music

Although the coincidence of the subcomplexes associated to the first and last piece of *Drei Klavierstücke* is encouraging given their common atonal nature, we retrieve identical preferred subcomplexes even by considering Debussy's *Clair de Lune* and *Arabesque* and Ravel's *Jeux d'Eau*. Again, this result is interesting, being Ravel and Debussy the two most relevant exponents of the *Impressionistic music*. However, it points out one of the limitations of this approach: forgetting the weight associated to the vertices, it is not possible to distinguish between identical *preferred subcomplexes* of the *Tonnetz*.

Let $\bar{f} : \mathcal{F} \to \mathbb{R}$ be the restriction of the height function to the fundamental domain of the deformed *Tonnetz*. Such a function allows to distinguish the three isomorphic subcomplexes associated to *Jeux d'Eau*, *Arabesque* and *Clair de Lune*. Moreover, it induces an ordering on the vertices of F given by $\bar{f} \circ h : |F| \to \mathbb{R}$, where $h : |F| \to \mathcal{F}$ is a simplicial homeomorphism.

The vertices of the preferred subcomplexes in Figure 6.8 are labelled with the values induced by the height function. The subcomplex associated to *Jeux d'Eau* is based on a minor triad and includes its perfect fourth, that is to say, this representation suggests a minor pentatonic sonority to be a characteristic of the piece. In the case of *Claire de Lune* a major triad enriched with its major second and its perfect fourth is highlighted. By ordering the notes according to their heights and assuming for simplicity the root of the major triad to be C, we obtain the sequence of note (G, C, D, E, F, A). This sequence reveals the diatonic inspiration of this piece, the height of the notes points out a mixolydian[1] structure. The subcomplex

[1] the mode built on the fifth degree of the diatonic scale and associated to dominant chords and pentatonic sonorities.

associated to the third movement of *Drei Klavierstücke* suggests either a sonority based on one of the modes deduced from the melodic minor scale as the dorian ♭2, or a chromatic construction of the composition.

The height function retrieves relevant information even neglecting the temporal ordering in which notes are presented. The natural advance from this point is to avoid the definition of an arbitrarily threshold and to extend the function defined on the vertices to the other simplices (edges and triangles) of the complex. Morse theory (Milnor, 1963) revealed the close relationship between the topology of a manifold and the critical points of a smooth function defined on it. In particular, discrete Morse theory (Forman, 1998, 2002) is an adaptation of this formalism to simplicial complexes. Unfortunately, in order to produce a discrete Morse function from a real function f defined on a point cloud (King et al., 2005), it has to be injective on the set of points where it is defined. Even after the restriction to the fundamental domain of the *Tonnetz*, where the labelling function is bijective, this hypothesis is too strong for music analysis: two or more pitch classes can be played exactly for the same amount of time in a composition, or never be played, and this information cannot be neglected. However, it is possible to describe the topology of a simplicial complex by taking into account an ordering induced on its simplices by a continuous real-valued function under milder assumptions. The solution is provided by the theory of persistent homology, we introduce in the next chapter.

Seven

Topological persistence

Topological persistence was introduced by Patrizio Frosini and collaborators under the name of Size Theory (Frosini, 1992), addressing the problem of shape recognition from a rigorous mathematical point of view. This theory is comparable (although more general) to the 0-dimensional persistent homology described in (Edelsbrunner et al., 2002).

As we shall see in the remainder of this chapter, the hypothesis required by the formalism of persistent homology are weak enough to make it suitable for a plethora of applications, including the analysis of music. It has been applied to the classification of shapes (Chazal et al., 2009; Di Fabio and Landi, 2011), of hepatic and melanocytic lesions (Adcock et al., 2014; d'Amico et al., 2004), the analysis of cortical data (Chung et al., 2009), covering of sensor networks (De Silva and Ghrist, 2007; Munch et al., 2012), group behaviour analysis (Topaz et al., 2015) and many other fields.

In order to safely introduce persistent homology, the homology of a topological space has to be defined.

7.1 Simplicial homology

The homology theory is a standard subject in Algebraic Topology. It is extensively described in a general setting in (Hatcher, 2002) and (Munkres, 1984). In this context we will describe homology as it is defined for simplicial complexes.

7.1.1 n-chains

Let K be a simplicial complex and $n \in \mathbb{Z}$. A simplicial *n-chain* is a formal sum $\sum_i \alpha_i \sigma_i$ where $\alpha_i \in \mathbb{Z}/2\mathbb{Z}$ and σ_i are n-simplices of K. Let $c = \sum_i \alpha_i \sigma_i$ and $d = \sum_i \beta_i \sigma_i$ be two n-chains and define their sum as

$$c + d = \sum_i (\alpha_i + \beta_i)\sigma_i. \tag{7.1.1}$$

The set of n-chains, equipped with the operation defined in Equation (7.1.1) form the *group of n-chains* $(C_n, +)$ or simply C_n, if the context allows to simplify the notation. The neutral element is 0 and the associativity of $+$ is inherited by the sum in $\mathbb{Z}/2\mathbb{Z}$. The inverse of an element $c \in C_n$ is $-c = c$, since $c + c = 0$. Moreover, C_n is abelian since the addition modulo 2 is commutative. The group of n-chains is defined for every $n \in \mathbb{Z}$. In particular, if n is less than 0 or greater than the dimension of K,

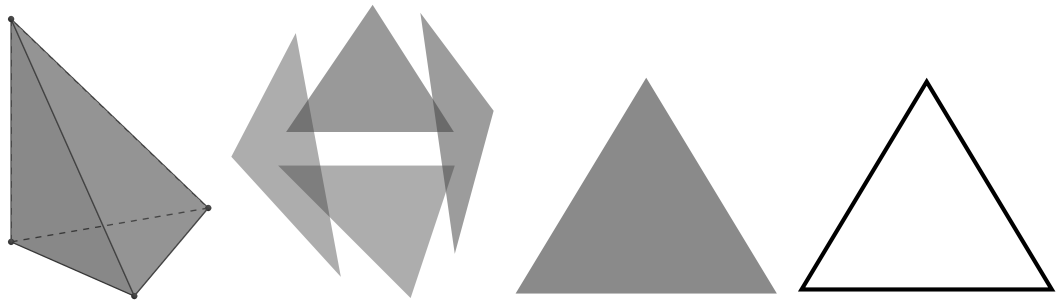


Figure 7.1: A 3 and a 2-simplex and their respective boundaries.

C_n is trivial. It is possible to link chain groups of different dimension by defining the *boundary homomorphisms*.

7.1.2 Boundary homomorphisms and homology groups

The boundary of the n-simplex $\sigma = [v_0, \ldots, v_n]$, denoted as $\partial_n(\sigma)$, is the sum of its $(n-1)$-faces. In symbols we have

$$\partial_n(\sigma) = \sum_{i=0}^{n} [v_0, \ldots, \hat{v_i}, \ldots, v_n].$$

where the hat indicates the vertex to be dropped in order to consider the ith face of σ. In Figure 7.1 a tetrahedron and a triangle are depicted along with their respective boundaries. When considering coefficients in fields different from $\mathbb{Z}/2\mathbb{Z}$ it is necessary to orient each simplex by taking into account the indices of its vertices. Thus, it is necessary to define the boundary homomorphism as an alternate sum, rather than a simple one (Hatcher, 2002, Ch. 2).

The boundary of a n-chain is the sum of its simplices' boundaries. Let c and d be two n-chains, then $\partial_n(c+d) = \partial_n(c) + \partial_n(d)$. Hence, $\partial_n : C_n \to C_{n-1}$ is a group homomorphism.
The sequence

$$\cdots \xrightarrow{\partial_{n+2}} \mathrm{C}_{n+1} \xrightarrow{\partial_{n+1}} \mathrm{C}_n \xrightarrow{\partial_n} \cdots \xrightarrow{\partial_2} \mathrm{C}_1 \xrightarrow{\partial_1} \mathrm{C}_0 \xrightarrow{\partial_0} 0,$$

of the abelian groups $\{\mathrm{C}_n\}_n$ equipped with the boundary homomorphisms, where $\partial_0 = 0$, is called a *chain complex*. A n-chain c with zero boundary, i. e. such that $\partial_n(c) = 0$, is called a *n-cycle*. The collection of n-cycles denoted by Z_n is the kernel of the boundary homomorphism ∂_n and consequently a subgroup of C_n.

A *n-boundary* is a n-chain c such that $\partial_{n+1}(d) = c$, for some $d \in C_{n+1}$. In other words, a n-boundary is a n-chain which is the boundary of a $(n+1)$-chain. In particular, the collection of n-boundaries $B_n = \operatorname{Im} \partial_{n+1}$ and hence $B_n \subseteq C_n$ is also a subgroup.

Lemma 7.1.1. *$\partial_{n+1} \circ \partial_n = 0$ for every $n \in \mathbb{Z}$.*

Proof. Let σ be a $(n+1)$-simplex. Its boundary consists of the sum of its n-faces and each $(n-1)$-face belongs exactly to two n-faces, hence $\partial_{n+1}\partial_n(\sigma) = 0$. □

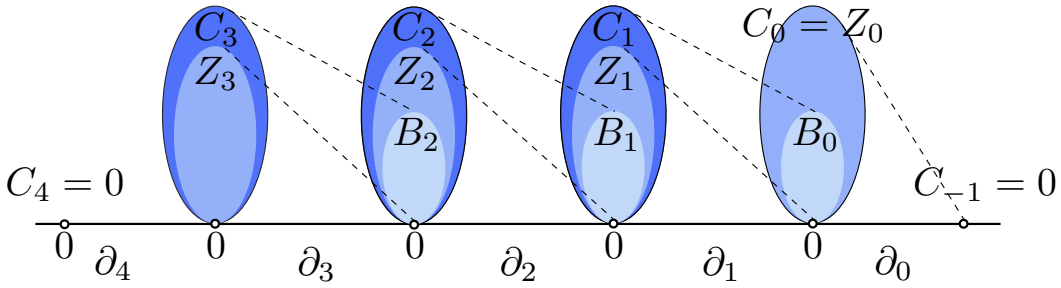


Figure 7.2: Representation of a chain complex associated to a 3-dimensional simplicial complex.

It is simple to verify this property in low dimension, for instance consider the boundary of the boundary of a 3-simplex:

$$\partial_2\partial_3([v_0, v_1, v_2]) = \partial_2([v_0, v_1] + [v_1, v_2] + [v_2, v_0]),$$

as we said above the boundary homomorphisms commute with the sum, hence

$$\partial_2([v_0, v_1] + [v_1, v_2] + [v_2, v_0]) = 2(v_0 + v_1 + v_2) = 0 \mod 2.$$

From the Lemma 7.1.1 follows that every n-boundary is a n-cycle. Hence, B_n is a subgroup of Z_n. We can consider the quotient Z_n/B_n, whose elements are the *cosets* of the form $c + B_n$ where $c \in Z_n$. This quotient is an abelian group, since Z_n is so. In Figure 7.2 a chain complex associated to a simplicial complex of dimension 3 is depicted. Observe how the 4th and 0th chain groups are both trivial. In particular, the triviality of the latter implies $C_0 = Z_0$. The dashed lines between consecutive chain groups denote the effect of the boundary homomorphism. The n-cycles vanish when mapped to the lower chain group; the n-boundaries are represented as a subset of the subgroup of n-cycles and correspond to the image of ∂_{n+1}.

Definition 7.1.1. The nth *homology group* of a chain complex is the quotient $H_n = Z_n/B_n$, for $n \in \mathbb{Z}$. Two cycles are said to be *homologous* if they are in the same coset.

The chain group is also a vector space for every $n \in \mathbb{Z}$. Furthermore, the group $C_n \simeq (\mathbb{Z}/2\mathbb{Z})^{s_n}$, where $s_n \in \mathbb{N} \cup \{ 0 \}$ is the number of n-simplices in K. Hence, C_n is generated by s_n elements. These elements can be thought as the vectors having only one non-zero component, corresponding to the ith n-simplex of K, for $i \in \{ 1, \ldots, s_n \}$. The same structure is inherited by the cycles and the boundaries of C_n. We define the nth *Betti number* of K as

$$\beta_n = \dim H_n = \dim Z_n - \dim B_n.$$

7.1.3 An algorithm for computing homology

To compute the Betti numbers of a simplicial complex K, it is necessary to introduce its matrix representation. The information provided by the boundary homomorphisms is stored in a collection of matrices called *boundary matrices*. The nth

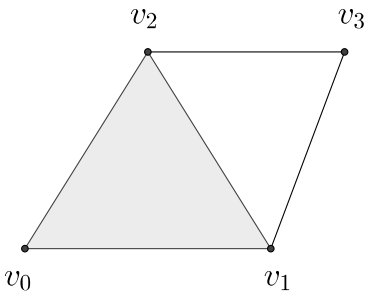


Figure 7.3: 2-dimensional simplicial complex.

boundary matrix $\mathcal{B}_n$ is defined as $\mathcal{B}_n(i,j) = 1$ if the i-th $(n-1)$-simplex is the boundary of the jth n-simplex and 0 otherwise. The ordering used to store the simplices in the boundary matrix is the one induced by the vertices of the simplicial complex.

Example 7.1.1 (Boundary Matrices)**.** Consider the simplicial complex K of Figure 7.3. Its 0-boundary matrix is

$$\mathcal{B}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}.$$

Since, the 0-simplices (columns of the matrix) have no faces. Ordering the 1-simplices of K as $[v_0, v_1], [v_0, v_2], [v_1, v_2], [v_1, v_3], [v_2, v_3]$ on the columns and the vertices following the subscripts indices on the rows, we have

$$\mathcal{B}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The last boundary matrix has one column corresponding to $[v_0, v_1, v_2]$ and six rows corresponding tot he 1-simplices of K ordered as before, we have

$$\mathcal{B}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Let $\mathbf{v}$ be the column vector of the coefficients of a n-chain c. Its boundary is computed as $\mathcal{B}_n \cdot \mathbf{v}$, where $\cdot$ is the standard matrix product. The vector $\mathbf{v}$ is a n-cycle if and only if it exists a vector $\mathbf{u} \in \mathrm{C}_{n+1}$ such that $\mathcal{B}_{n+1} \cdot \mathbf{u} = \mathbf{v}$.

The rank of the n-chain group C_n is the number of n-simplices in the simplicial complex K let denote it as s_n, hence the n-th boundary matrix $\mathcal{B}_n \in \mathrm{Mat}(s_{n-1}, s_n)$. To represent the sizes of B_n and Z_n and consequently H_n, the matrix $\mathcal{B}_n$ is reduced to normal form, as it is shown in Figure 7.4. The operations required to achieve the normal form reduction of the matrix are equivalent to the ones used in Gaussian reduction to solve linear systems of equations. See Algorithm 7.1 for the pseudocode.

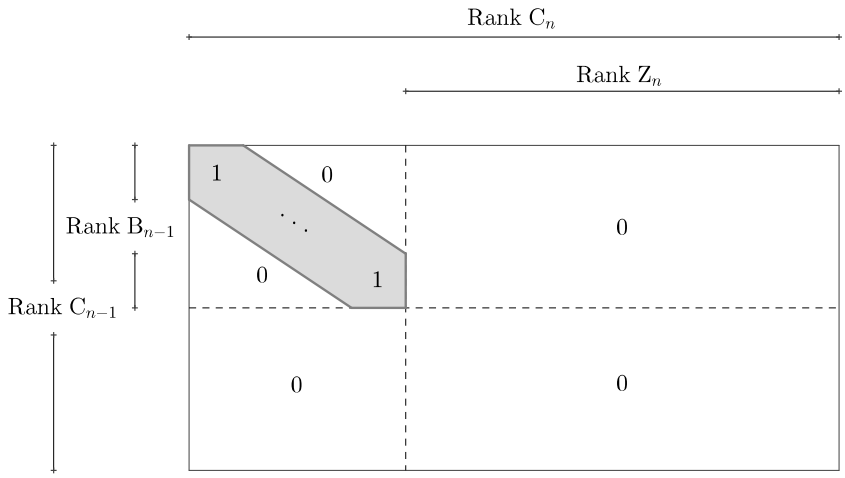


Figure 7.4: Reduced n-th boundary matrix.

Algorithm 7.1 Boundary Matrix Reduction.

Input:
$\mathcal{B}_n$ ▷ a boundary matrix
Output:
R ▷ the reduced boundary matrix

while $m \in \{ 1, \ldots, s_n \}$ **do**
  if $\mathcal{B}_n(i, j) == 1$ and $i \geqslant m$ and $l \geqslant m$ **then**
    Exchange the rows m and i and the columns m and j;
    for $k \in \{ m+1 \ldots s_{n-1} \}$ **do**
      if $\mathcal{B}_n(k, m) == 1$ **then**
        add row k to row m;
      end if
    end for
    for $l \in \{ m+1 \ldots s_n \}$ **do**
      if $\mathcal{B}_n(m, l) == 1$ **then**
        add column j to column m;
      end if
    end for
  end if
end while

On every iteration on m at most a linear number of rows and columns operations is performed. Hence the total running time is at most $O\left(N^3\right)$, where N is the number of simplices of K.

Example 7.1.2 (Boundary matrix reduction)**.** Consider the boundary matrices of Example 7.1.1, in their normal form they are

$$\mathcal{R}_0 = \mathcal{B}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathcal{R}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and

$$\mathcal{R}_2 = \mathcal{B}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Setting $z_n = \text{Rank}Z_n$ and $b_n = \text{Rank}B_n$, we have $z_0 = 4$ from $\mathcal{B}_0$ and $b_0 = 3$ from $\mathcal{B}_1$, hence the 0-th Betti number is $\beta_0 = 1$, which is exactly the expected value, since the simplicial complex of Figure 7.3 has one connected component. In dimension 1 we have $z_1 = 2$ and $b_1 = 1$, thus $\beta_1 = 1$ corresponding to the 1-dimensional hole of the simplicial complex. Finally $z_2 = 0$ and hence $\beta_2 = 0$.

7.2 From homology to persistent homology

7.2.1 An intuition

Observing a shape for the first time, one tries to identify its persistent properties by neglecting details that can be easily lost by changing the position of the shape or hidden by small occlusions.

The idea behind persistent homology is to measure these properties by rebuilding a shape as a monotonic sequence of nested spaces called a *filtration*. In Figure 7.5, we considered a point cloud subsampled from the image of a manuscript note and we associated to each point a circle of radius $r \in \mathbb{R}$. The blob formed by the union of the circles assumes the shape of a musical note, while increasing the radius of the circles. Disconnected regions of the blob, as well as the small holes generated by partial intersections of circles do not impede the perception of the whole shape. Moreover, it is necessary to largely increase the radius of the circles to hide the *persistent* shape of the note.

As an example, consider the classification of manuscript notes, in order to recognise their author. The filtration we produced above, is sensitive to the variations of the relative positions of the points, but invariant under uniform translation or rotations of the whole point cloud. Hence, this particular choice is suitable for the discrimination of the author who wrote the notes, that can be rotated according to

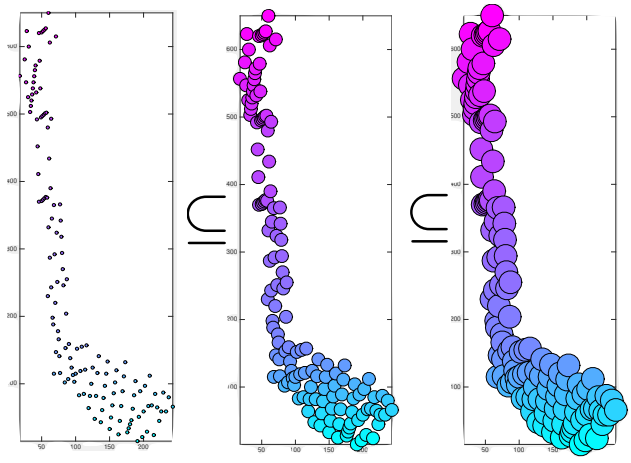


Figure 7.5: Three topological spaces corresponding to a part of the filtration associated to a cloud of points derived form an image representing a manuscript note.

their position on the score, but that are generally characterised by similar shapes of the head, or thickness of the stem.

Shortly, we shall develop the necessary theory to define a filtration function by considering either a continuous function on a topological space, or as a sequence of nested subcomplexes of a given simplicial complex. Furthermore, the noisy and persistent properties of the shape will be represented by a multiset of 2-dimensional points. This representation provides a surprisingly suitable framework for the comparison of shapes and in our case, of musical compositions.

7.2.2 Persistent homology for topological spaces

Here and for the remainder of the dissertation, we assume $\mathbb{X}$ to be a triangulable topological space and $f : \mathbb{X} \to \mathbb{R}$ a continuous function. We recall that homology is computed with coefficients in $\mathbb{Z}/2\mathbb{Z}$.

Homological critical values and tame functions

Let $f : \mathbb{X} \to \mathbb{R}$ be a continuous function on $\mathbb{X}$. We denote by $\mathbb{X}_u = f^{-1}((-\infty, u])$ the *sub-level set* of the function f, for every $u \in \mathbb{R}$. Consider the topological sphere TS depicted in Figure 7.6 on the following page. The real numbers

$$a_1 \leqslant a_2 \leqslant \ldots \leqslant a_7$$

generate seven nested sub-level sets $TS_{a_i} = f^{-1}((-\infty, a_i])$ of the height function $f : TS \to \mathbb{R}$, for $i \in \{1, \ldots, 7\}$.
To safely introduce persistence homology, two fundamental ingredients have to be defined: homological critical values (Govc, 2013) and tame functions.

Definition 7.2.1. Let $\mathbb{X}$ a topological space and $f : \mathbb{X} \to \mathbb{R}$ a continuous function. A real number a is called a *homological regular value* of f if there exists $\varepsilon > 0$, such

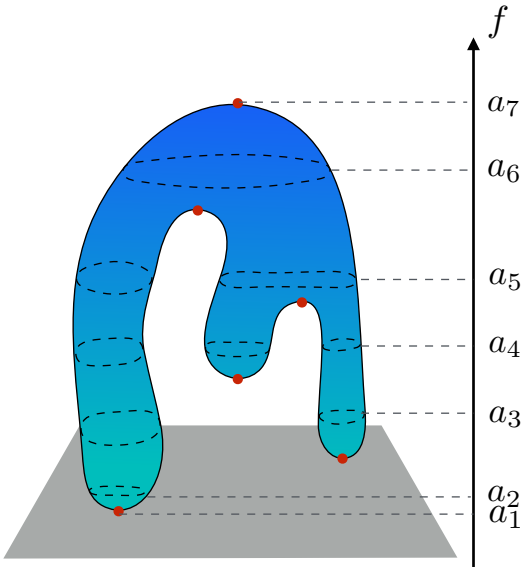


Figure 7.6: Sub-level sets of the height function on a topological sphere. The six critical points of the height function are depicted as red dots.

that for every couple of real numbers $x < y$ on the interval $(a-\varepsilon, a+\varepsilon)$, the inclusion

$$f^{-1}((-\infty, x]) \hookrightarrow f^{-1}((-\infty, y]),$$

induces isomorphisms on all homology groups. Otherwise, a is called a *homological critical value* of f.

Definition 7.2.2. Let $\mathbb{X}$ be a triangulable topological space. A continuous function $f : \mathbb{X} \to \mathbb{R}$ is *tame* if it has a finite number of homological critical values and the homology groups $H_k(\mathbb{X}_u)$ are finite-dimensional for every $u \in \mathbb{R}$ and $k \in \mathbb{Z}$.

Note that in general, the definition of a tame function asks the homology groups to have finite dimension. The fact the f has a finite number of homological critical values assures that changes in the homology groups occur a finite number of times along the filtration and in correspondence of these critical values. Examples of tame functions are Morse functions on manifold and piecewise linear functions on triangulable topological spaces.

Persistence modules and persistent Betti numbers

Definition 7.2.3. Let $\mathbb{X}$ be a triangulable topological space, $f : \mathbb{X} \to \mathbb{R}$ a tame function and $u, v \in \mathbb{R}$, such that $u < v$. The kth *persistence module* $H_k^{u,v}$ is the image of the homomorphism

$$\iota_k^{u,v} : H_k(\mathbb{X}_u) \to H_k(\mathbb{X}_v),$$

induced by the inclusion $\mathbb{X}_u \hookrightarrow \mathbb{X}_v$.

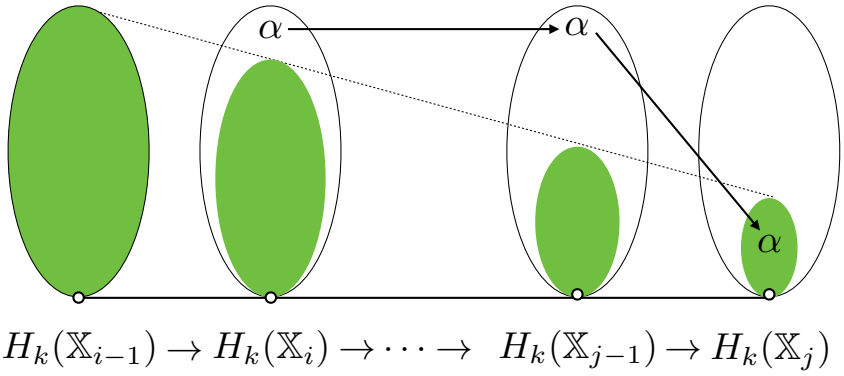


Figure 7.7: The class α is born in $\mathbb{X}_i$ since it is not in the image of $\iota_k^{i-1,i}$ depicted in green. It dies in $\mathbb{X}_j$, since it merges in $\mathrm{Im}\iota_k^{i-1,j}$.

The *k-persistent Betti number* is defined as $\beta_{f,k}^{u,v} = \dim \mathrm{Im}(\iota_k^{u,v})$, for every $k \in \mathbb{Z}$. It counts the homology classes of dimension k *surviving* in the passage from $\mathbb{X}_u$ to $\mathbb{X}_v$. Here, it is possible to speak about the *dimension* of the image of the function induced by the inclusion $\mathbb{X}_u \hookrightarrow \mathbb{X}_v$, since considering coefficients in a field, $H_k(\mathbb{X}_u)$ and $H_k(\mathbb{X}_v)$ are vector spaces and $\iota_k^{u,v}$ is a linear function.

It is now possible to define the filtration of the topological space $\mathbb{X}$ induced by the sub-level sets of a tame function $f : \mathbb{X} \to \mathbb{R}$. By Definition 7.2.3 f has a finite number of homological critical values, say $\{c_1, \ldots, c_n\}$, where $n \in \mathbb{N}$. Let $\{r_0, \ldots, r_n\}$ be regular homological values of f, such that $r_{i-1} \leqslant c_i \leqslant r_i$ for every $i \in \{1, \ldots, n\}$. It is possible to define a *filtration* of $\mathbb{X}$ as the collection of nested subspaces $\{\, \mathbb{X}_{r_i} \,\}_{i \in \{0,\ldots,n\}}$, such that

$$\mathbb{X}_{r_0} \subseteq \mathbb{X}_{r_1} \subseteq \cdots \subseteq \mathbb{X}_{r_n}.$$

In addition, set $r_{-1} = c_0 = -\infty$ and $r_{n+1} = c_{n+1} = +\infty$, in order to include in the filtration that empty set and the whole space. If the context is clear we denote the subspaces of this filtration simply as $\mathbb{X}_i$.

Finally, by traversing the filtration, we assist to a finite number of changes (since f is tame) of the homology groups associated to each subspace $\mathbb{X}_i$ for $i \in \{0, \ldots, n+1\}$. In Figure 7.7, we say that the homology class α *is born* entering in $\mathbb{X}_i$, since it does not came from a class in $\mathbb{X}_{i-1}$. Symmetrically, we say that α dies entering $\mathbb{X}_j$ if the image of the map induced by the inclusion $\mathbb{X}_{j-1} \subset \mathbb{X}_j$ contains α and the image of the map induced by $\mathbb{X}_{i-1} \subseteq \mathbb{X}_{j-1}$ does not.

Persistence barcodes and persistence diagrams

The information retrieved by the analysis of the lifespan of homology classes along the filtration can be represented as a diagram called *persistence diagram*. In such a diagram birth and death-levels of each homology class are represented as points lying in the open half-plane above the diagonal and endowed with a multiplicity (Frosini and Landi, 2001; Ferri et al., 2011).

To describe how the persistence diagram is built, it is necessary to introduce another fundamental ingredient of persistent homology: the *pairing*. Under the

hypotheses of triangulability and tameness of $\mathbb{X}$ and f respectively, let c_i be a critical homological value of f. If c_i is responsible of the birth of a homological class α, it is *paired* with the homological critical value c_j responsible of its death (if it exists). The lifespan of α corresponds to the open interval $[c_i, c_j)$, with $i < j$. The homology with infinite lifespan are paired with ∞. The collection of intervals retrieved running the whole filtration is called a *persistence barcode* (Carlsson et al., 2005; Ghrist, 2008).

The *persistence diagram* is built considering the pairing (c_i, c_j) as point in $\mathbb{R}^2$, or to be more precise, in the closure of the Euclidean plane including the points at infinity.

Let $\Delta = \{ (u, v) \in \mathbb{R}^2 \mid u = v \}$ be the diagonal of the Euclidean plane, $\Delta^+ = \{ (u, v) \in \mathbb{R}^2 \mid u < v \}$ be the open-half plane above the diagonal and $\Delta^* = \Delta^+ \cup \{ (u, \infty) \mid u \in \mathbb{R} \}$ the extension of Δ^+ including the points at infinity. Observe that the definition of Δ^* is necessary, in order to describe cycles with infinite lifespan.

Definition 7.2.4. Let (u, v) be a point of Δ^+. The number $\mu(u, v) \in \mathbb{R}$ realising the minimum over the real numbers $\varepsilon > 0$, with $u + \varepsilon < v - \varepsilon$, of

$$\beta_{f,k}(u + \varepsilon, v - \varepsilon) - \beta_{f,k}(u - \varepsilon, v - \varepsilon) - \beta_{f,k}(u + \varepsilon, v + \varepsilon) + \beta_{f,k}(u - \varepsilon, v + \varepsilon),$$

is called the multiplicity of (u, v) for the persistent Betti number $\beta_{f,k}$ and $k \in \mathbb{Z}$. A point (u, v) is called a *proper cornerpoint* for $\beta_{f,k}$ if its multiplicity is strictly positive.

Definition 7.2.5. Let $r : u = \bar{u}$ be a vertical line in $\mathbb{R}^2$. We identify it with its point at infinity $(\bar{u}, \infty) \in \Delta^*$. The multiplicity $\mu(\bar{u}, \infty)$ is the minimum over the positive real numbers ε, with $\bar{u} + \varepsilon < 1/\varepsilon$, of

$$\beta_{f,k}\left(\bar{u} + \varepsilon, \frac{1}{\varepsilon}\right) - \beta_{f,k}\left(\bar{u} - \varepsilon, \frac{1}{\varepsilon}\right).$$

A point at infinity endowed with a strictly positive multiplicity is called a *cornerpoint at infinity* for $\beta_{f,k}$.

Finally, we can introduce the definition of persistence diagram.

Definition 7.2.6. The *k-persistence diagram* $D_k(f)$ is the multiset[1] of all cornerpoints for $\beta_{f,k}$, union the points of the diagonal Δ counted with infinite multiplicity.

The persistence barcode and the persistence diagram associated to the filtration of a topological sphere are depicted in Figure 7.8. Green intervals describe the information concerning the 0-dimensional persistence of the shape, while the blue ones describe the contribution of the 2-persistence module. Finite intervals correspond to proper cornerpoints, while the cornerpoints at infinity are represented as vertical half-lines, also called *cornerline*.

In conclusion, persistence diagrams describe the topological and geometrical properties of a shape $\mathbb{X}$. These properties are retrieved by the analysis of the life and death-levels of the homological classes of the nested spaces determined by the

[1] Each cornerpoint is equipped with a multiplicity.

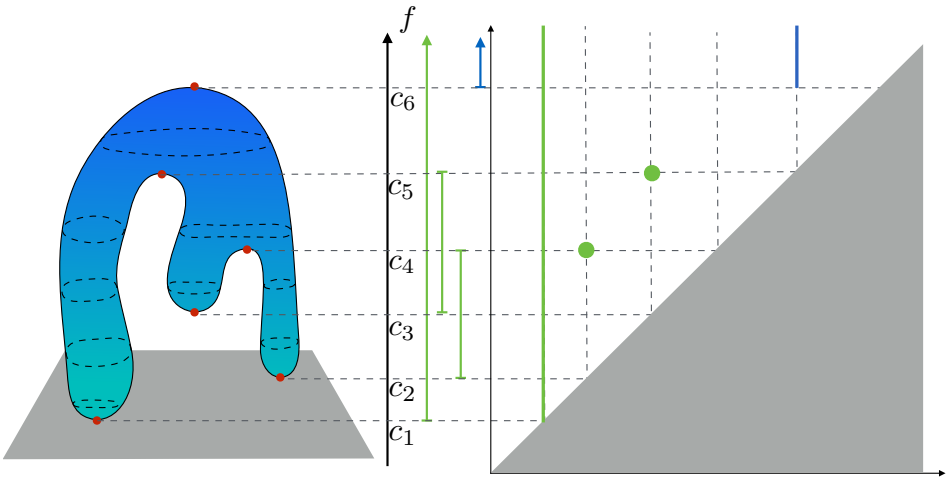


Figure 7.8: An example of persistence barcode and persistence diagram. Noisy classes are represented as short bars in the barcodes and as points near the diagonal in the diagram representation. The critical points of the height function are denoted by red circles. According to their labels, the pairing is given by (c_1, ∞), (c_2, c_4), (c_3, c_5) and (c_6, ∞).

filtration induced by the sub-level sets of a tame functions f. Moreover, the lifespan of the homology classes represented by a cornerpoint corresponds to its distance from the diagonal. Thus, noisy and persistent homological classes are represented by cornerpoints lying near to or far from the diagonal, respectively.

Bottleneck distance

Persistence diagrams are simpler than the shape they represents and describe its topological and geometrical properties, as they are highlighted by the homological critical values of the function used to build the filtration. The bottleneck distance allows to compare such diagrams.

Definition 7.2.7. Let $\mathbb{X}$ be a triangulable topological space and $f, g : \mathbb{X} \to \mathbb{R}$ two tame functions. The *bottleneck distance* between $D_k(f)$ and $D_k(g)$ is

$$d_B\left(D_k(f), D_k(g)\right) = \inf_{\gamma} \sup_{p \in D_k(f)} \|p - \gamma(p)\|_\infty ,$$

where $\gamma : D_k(f) \to D_k(g)$ is a bijection and $\|p - \gamma(p)\|_\infty = \max_{p \in D_k(f)} |p - \gamma(p)|$.

In Figure 7.9 a bijection between two k-persistence diagrams is depicted. Corner points belonging to the two diagrams are depicted in orange and yellow, respectively. Observe how the inclusions of the points of Δ allows the comparison of multisets of points whose underlying set has different cardinality (see Section 3.1 for a definition of multiset) by associating one of the purple points to one of the points lying on the diagonal.

An important property of persistence diagrams is their stability. A small perturbation of the tame function f produces small variations in the persistence diagram with respect to the bottleneck distance.

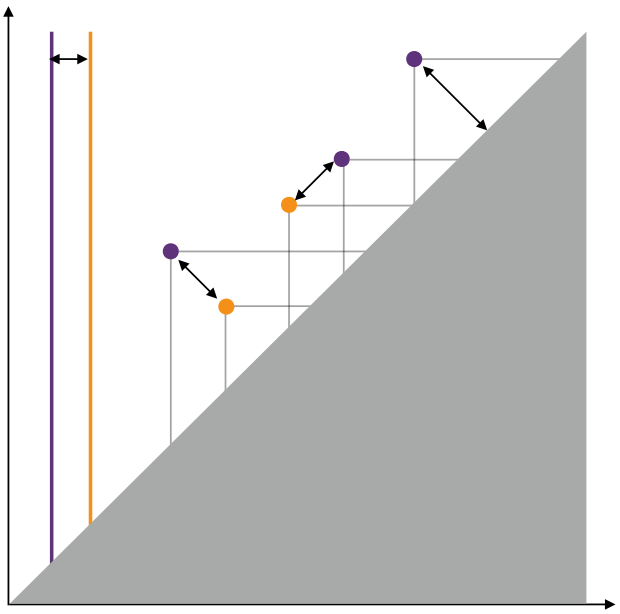


Figure 7.9: A matching between two k-persistence diagrams. The bijections between elements of the diagrams is denoted using left-right arrows.

Theorem 7.2.1. *Let $\mathbb{X}$ be a triangulable topological space and $f, g : \mathbb{X} \to \mathbb{R}$ two tame functions. For every integer k, the inequality*

$$d_B\left(D_k(f), D_k(g)\right) \leqslant \|f - g\|_\infty ,$$

where $\|f - g\|_\infty = \sup_x |f(x) - g(x)|$, holds.

7.2.3 An algorithm for computing persistence

Persistence is computed through an algorithm mirroring the one we described in Algorithm 7.1. Let K be a triangulation of $\mathbb{X}$, and $\tilde{f} : K \to \mathbb{X}$ a monotone function such that $\tilde{f}(\tau) \leqslant \tilde{f}(\sigma)$ if τ is a face of σ. Consider an ordering of the simplices of K, such that each simplex is preceded by its faces and $\tilde{f}$ is non-decreasing.

This ordering allows to store the simplicial complex in a *boundary matrix* B, whose entries are defined as

$$B(i,j) = \begin{cases} 1 & \text{if } \sigma_i < \sigma_j \\ 0 & \text{otherwise} \end{cases} . \tag{7.2.1}$$

The algorithm receives in input a boundary matrix B and reduces it to a new $0-1$ matrix R via elementary column operations. Let $J = \{ 1, \ldots, n \}$ be the indices of the columns of B and

$$\begin{aligned} \mathrm{low}_R : J &\to \mathbb{N} \\ j &\mapsto l, \end{aligned}$$

where l is the lower row index of the last 1 entry of the jth column. If a column has only 0 entries $\mathrm{low}_R(j)$ is undefined. A matrix R is reduced if for every couple of

non-zero columns of indices $j \neq j'$, $\text{low}_R(j) \neq \text{low}_R(j')$. The reduction process is described recursively in Algorithm 7.2. The corresponding Python code is available in Appendix C.

Algorithm 7.2 Persistence Algorithm.

Input:
B ▷ a boundary matrix
Output:
R ▷ the reduced boundary matrix

```
1: function R = REDUCE(B)
2:     R = B
3:     for j ∈ { 1, ..., m } do                    ▷ the number of simplices in K
4:         for j_0 ∈ { 1, ..., j − 1 }: do
5:             if low_R(j) == low_R(j_0): then
6:                 R_j = R_j + R_{j_0} mod 2       ▷ Add the j-th and the j_0-th columns
7:                 R = reduce(R)
8:                 return R
9:             end if
10:        end for
11:    end for
12:    return R                                    ▷ if B is already reduced, return B
13: end function
```

Thus, the algorithm compute a upper-triangular invertible matrix $V = v(i,j)$ whose entries are elements of $\mathbb{Z}_2$, such that $R = BV$. From the reduced matrix R it is possible to deduce the pairing of critical simplices and thus the k-persistence diagram for every $k \in \mathbb{Z}$. Consider the couple of simplices (σ_i, σ_j) such that $i = \text{low}_R(j)$. We call σ_i *positive* and σ_j *negative*, since the homology class created by σ_i dies when σ_j is introduced. By construction of B in Equation (7.2.1), $\dim\sigma_i = \dim\sigma_j - 1 = n$, thus the coordinates $\left(\tilde{f}(\sigma_i), \tilde{f}(\sigma_j)\right)$ has to be added to the n-dimensional persistence diagram.

Observe that the reduced matrix is not unique, for instance it can be computed as the complete Smith normal form of B. However the points $\left(\tilde{f}(\sigma_i), \tilde{f}(\sigma_j)\right)$ do not depend on the choice of R. Let denote with M_i^j the minor of the matrix $M \in \text{Mat}_{\mathbb{R}}(k,l)$ obtained by deleting the first $i-1$ rows and $l-j$ columns, and define

$$r_B(i,j) = \text{rank}B_i^j - \text{rank}B_{i+1}^j + \text{rank}B_{i+1}^{j-1} - \text{rank}B_i^{j-1}.$$

Lemma 7.2.2. *Let $B = RU$, being $U = V^{-1}$, a decomposition of B. Then $\text{low}_R(j) = i$ if and only if $r_B(i,j) = 1$. In particular, the pairing function is not dependent on the choice of R.*

Proof. A proof of the Pairing Uniqueness Lemma can be found in (Cohen-Steiner et al., 2006, Sec. 3). □

If the number of simplices in the complex is m, then the algorithm runs in a $O(m^3)$ time in the worst-case. For instance, the Vietoris-Rips complex of a n-points

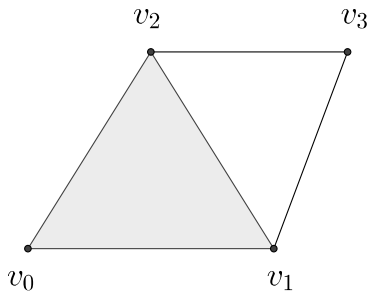


Figure 7.10: 2-dimensional simplicial complex.

cloud has at most $\binom{n}{k}$ simplices for each dimension k, making the computation time prohibitive. It is possible to fix this issue by computing persistence in low dimensions, or limiting the length of the radius associated to each point.

We give an explicit example of the construction of the persistent boundary matrix and the deduction of the pairing in the following examples.

Example 7.2.1. Consider the simplicial complex K depicted in Figure 7.10. It consists of 10 simplices: 4 vertices, 5 edges and 1 triangle. To get a filtration we add them following the order induced by their dimension. Since each simplex has to be associated to a column and a row of the boundary matrix, we number them from 1 to 10 following the ordering defined by the filtration. Hence, the simplices will be placed in the order $(v_0, v_1, v_2, v_3, e_{01}, e_{12}, e_{20}, e_{13}, e_{23}, t_{012})$ on the rows and columns of the matrix:

$$B = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \left[\begin{array}{cccccccccc} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array}$$

Following the algorithm we look for columns whose last 1 entry has the same index, i. e. the column $B(j)$ and $B(j_0)$ with $\mathrm{low}_B(j) = \mathrm{low}_B(j_0)$ with $j \in \{1, \ldots, 10\}$ and $j_0 < j$. In this case we have $\mathrm{low}_B(7) = \mathrm{low}_B(6)$ and $\mathrm{low}_B(9) = \mathrm{low}_B(8)$. We sum

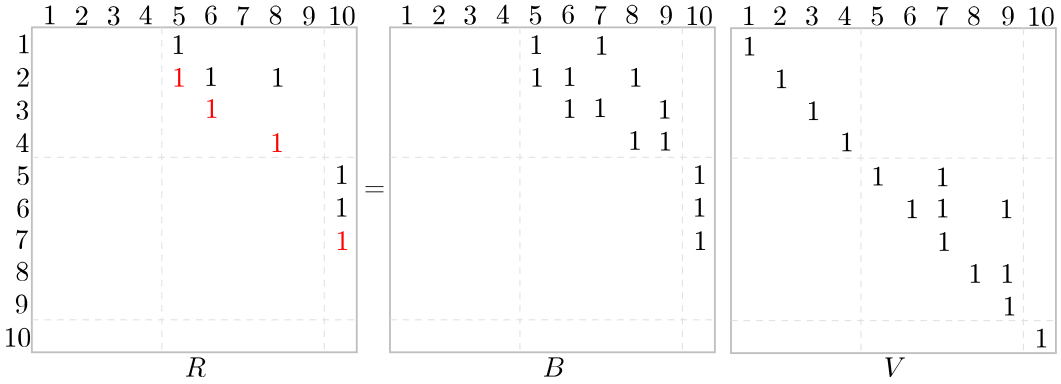


Figure 7.11: Reduction of the persistent boundary matrix to normal form.

the two columns obtaining

$$R_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \left[\begin{array}{cccccccccc} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array}$$

Iterating this process once we obtain the reduced boundary matrix $R_2 = R$, depicted in its decomposition $R = DV$ in Figure 7.11.

The iteration of the algorithm stops at its second step, since there are no more $\mathrm{low}_R(j) = \mathrm{low}_R(j_0)$ with $j_0 < j$, while j ranges on the column's indices of the matrix. Consider the reduced matrix R of Figure 7.11, the red 1 of the 5-th column, corresponds to the row index 2, that is to say the vertex 2 creates a 0-cycle, killed by the edge 5. The same argument holds for the 1 in position $(3, 6)$ and $(8, 4)$. When the edges 7 and 9 appear nothing change in terms death of cycles, since the columns $R(7)$ and $R(9)$ has only 0 entries. These two zero columns corresponds to two 1-cycles generated by the edges $(5, 6, 7)$ and $(6, 8, 9)$ as it is shown by the 7-th and 9-th column of the matrix V in Figure 7.11. While the triangle kills the first 1-cycle when the edge 7 is added to the complex in position $(7, 10)$, the other one survive along the whole filtration.

Eight

A topological fingerprint for music analysis

A persistence diagram is a fingerprint of a shape that represents its geometrical and topological properties as a multiset of 2-dimensional points. The deformations of the *Tonnetz* we discussed in Chapter 6 can be analysed using the height function defined on $\mathcal{T}$ to induce a filtration on the fundamental domain F. Moreover, the fundamental domain of the *Tonnetz* is completely rebuilt by this filtration, allowing to remove the threshold we defined in Section 6.1.1 and to take into account pitch classes that are less used in the piece, but that could reveal interesting properties of musical phrases, or whole compositions.

In the first part of this chapter we set up all the machineries needed to safely compute persistent homology when considering the deformation of the *Tonnetz* and analyse the persistence diagrams associated to several music pieces. In the second part we will utilise the bottleneck distance (see Definition 7.2.7) to provide a distance between musical pieces and to classify them, according to their *persistent properties*.

Remark 11. In the following applications, the persistence diagrams and the bottleneck distance will be computed by using *Dionysus*[1].

8.1 Persistent homology classification of deformed *Tonnetze*

The main aim of this section is to compute the persistent homology of the deformed *Tonnetz* we described in Section 6.1. In the previous chapter, we shown how a filtration can be defined considering a tame function f on a topological space $\mathbb{X}$. A filtration of a finite simplicial complex K can be provided as a sequence of nested subcomplexes $\{ K_0, \dots, K_n \}$ containing as its first and last elements (considering the ordering induced by the index of the subcomplexes of the filtration) the empty set and K, respectively.

In our case, it is necessary to define a filtration on a simplicial complex K, equipped with a function $f : V \to \mathbb{R}$ defined on its 0-skeleton.

[1]Dionysus is a C++ library for persistent homology available at `http://www.mrzv.org/software/dionysus/`.

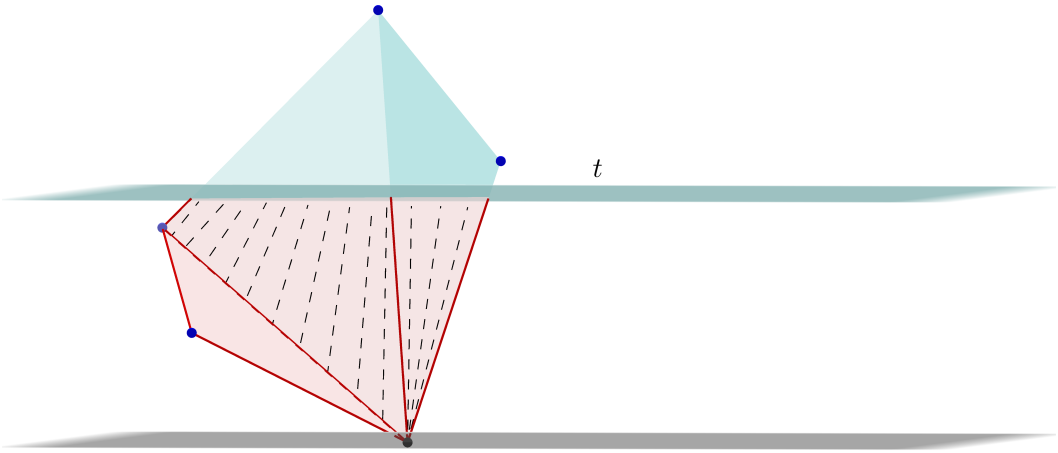


Figure 8.1: Lower star filtration of a simplicial complex.

8.1.1 The lower star filtration

Let $f_V : V \to \mathbb{R}$ be a real-valued function defined on the vertices of K. Linearly extending f_V we obtain a piecewise linear function $f : |K| \to \mathbb{R}$, such that $f(x) = \sum_i b_i(x) f(v_i)$. This continuous, piecewise extension of f allows to build a filtration of K. Moreover, since K is finite, f is tame.

Remark 12. It is possible to show that under some hypotheses, every filtration of a simplicial complex K is induced by a continuous function $f : K \to \mathbb{R}$ (Di Fabio and Frosini, 2013).

Assume that f is injective on the vertices of K. Then it is possible to order them as

$$f(v_0) < f(v_1) < \cdots < f(v_n). \tag{8.1.1}$$

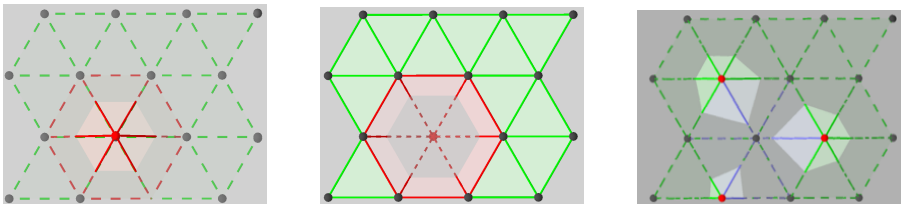
For every $0 \leqslant i \leqslant n$ define K_i as the subcomplex defined by the first i vertices, i. e. a simplex σ belongs to K_i if and only if its vertices are smaller or equal to v_i with respect to the ordering of Equation (8.1.1). The lower star of v_i is defined as

$$St\_v_i = \{\, \sigma \in St\, v_i \;\mid\; x \in \sigma \Rightarrow f(x) \leqslant f(v_i) \,\},$$

where $St\, v_i$ is the star of v_i, as it has been defined in Section 2.1.2 (see also Figure 2.3 for an intuition). Each simplex has a unique maximum vertex, since we assumed f to be injective on V, such that σ belongs to a unique lower star. Moreover, $K_i = \cup_{j \leqslant i} St\_v_j$, and $K_n = K$. The *lower star filtration* of K is given by

$$\emptyset = K_{-1} \subset K_0 \subset K_1 \subset \cdots \subset K_n = K.$$

Each element of the filtration corresponds to a sub-level set of the function f. Furthermore, for $t \in [f(v_i), f(v_{i+1})) \subset \mathbb{R}$, $|K|_t = f^{-1}((-\infty, t])$ has the same homotopy type of K_i. Let σ be a simplex that is cut by the plane defined by $z = t$. Assume that $\sigma = [v_0, \ldots, v_k] \in K$, then there exists at least a couple (v_l, v_k) of distinct vertices of σ, such that $v_l \in K_i$ and $v_k \in K - K_i$. Consider σ as a union of line segments connecting the points of its maximal face in K_i to its the maximal face in $K - K_i$. By construction the collection of such line segments lies only partly in



(a) Maximum (b) Minimum. (c) Monkey saddle.

Figure 8.2: Critical points of the height function defined on the vertices of a portion of the deformed *Tonnetz*.

$f^{-1}((-\infty, t])$. Refer to Figure 8.1 for an intuitive representation of this construction. Define the fractions of the line segments contained in K_i as

$$s : [0,1] \to \sigma$$
$$s(\lambda) = \lambda x + (1-\lambda)y,$$

where $f(s(0)) = f(y) = t$ and $s(1) = x$ is the endpoint of the line segment. By considering the deformation retraction, obtained going from time $\lambda = 0$ to $\lambda = 1$, we have that $|K|_t$ and K_i have the same homotopy type.

8.1.2 A filtration of the deformed *Tonnetz*

The deformed geometrical realisation of the infinite planar *Tonnetz* is not a suitable for the computation of persistent homology. Following a symmetrical approach with respect to the one we used in Section 6.1.2, we consider the fundamental domain F of the *Tonnetz* and its warping $\mathbb{T}$. The idea is to use the values of the height function to induce a filtration on $\mathbb{T}$.

However, even after this restriction, the height function can assume the same value on several vertices of $\mathbb{T}$. Indeed, more than one pitch class could be silenced in a phrase, having height 0, or two or more pitch classes could be played exactly for the same amount of time.

Maintaining the notation introduced in the previous paragraph, assume that the function $f_V : V \to \mathbb{R}$ has the same value on some (or even all) the vertices of K. Consider the set of the unique values of f_V, ordered as $a_1 < \cdots < a_n$, with $n \in \mathbb{N}$. Define $V_i = \{v \in V \mid f(v) = a_i\}$, the collection of vertices whose value with respect to the function f_V is a_i. We define $K_i = \cup_{l=1}^{i} St\_V_l$. The sequence

$$\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K$$

is a filtration of the complex K. Hence, it is possible to induce a filtration on the finite simplicial complex $\mathbb{T}$, by considering the sub-level sets of the linear extension of the height function defined on the vertices of $\mathcal{F}$.

Remark 13. It is possible to approximate the linear extension of f with the constant linear function $\bar{f}(\sigma) = \max_{x \in \sigma} f(x)$, in order to obtain a function that is monotone in the sense of simplicial complexes (if $\sigma \subset \tau$, then $f(\sigma) \leqslant f(\tau)$).

The homological critical values, whose pairing determine the lifespan of the homological classes of $\mathbb{T}$ correspond to the critical points of the height function on the deformed *Tonnetz*. In Figure 8.2 the simplest configurations of a maximum, a minimum and a saddle on the geometrical realisation of a portion of the deformed *Tonnetz* are depicted. Observe that, in our case a maximum or minimum can be a whole subcomplex of connected pitch classes, whose vertices share the same height. As an example, the configuration depicted in Figure 6.4 on page 84 is obtained by playing the pitch classes corresponding to a major triad for the same amount of time.

Remark 14. For an intuition concerning the sub-level sets of the height function, it is possible visualise them using the web application. The button *disp_filtr* generates the plane $z = 0$. The slider *filt_height* allows to change the height of the plane and visualise the sub-level sets of the height function.

8.2 Musical interpretation and persistent clustering

Introducing the Topological Persistence, we claimed that it is capable of describing persistent features of a shape, mirroring the process we use to identify them. In a music analysis context, it is necessary to endow our filter with a relative pitch rather then an absolute one. This is why we chose the height function. On one hand, it assures the invariance under uniform transposition of a phrase; on the other hand, it takes into account the structure of the *Tonnetz*, when extending the function to the whole simplicial complex.

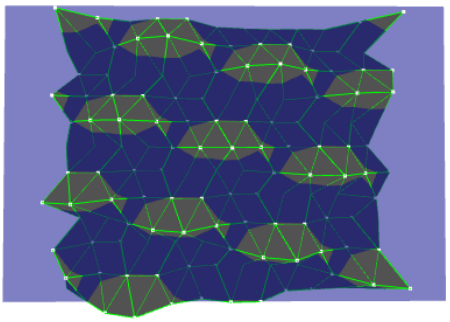
In Figure 8.3 a particular sub-level set of the height function of several deformed *Tonnetze* is depicted. The differences between the tonal and atonal approaches are highlighted by the geometry of the sub-level set associated to each composition. By considering a single sub-level set at a fixed height t, we obtain the specular approach to the one we considered in Section 6.1.2. The filtration induced by the height function rebuilds $\mathbb{T}$ entirely, hence it is not necessarily to fix a threshold. The information retrieved evaluating the birth and death-levels of the 0 and 1 homological classes traversing the filtration is encoded in two persistence diagrams.

8.2.1 Musical Interpretation

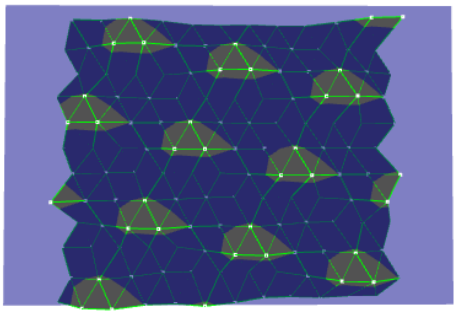
The 0 and 1-persistence diagrams associated to three compositions are represented in the first and second row of Figure 8.4, respectively. The two diagrams gives two representation of the each shape, in our case the different configurations of cornerpoints and cornerlines can be interpreted as *descriptors* of the compositional styles characterising the compositions we analysed.

0-persistent homology

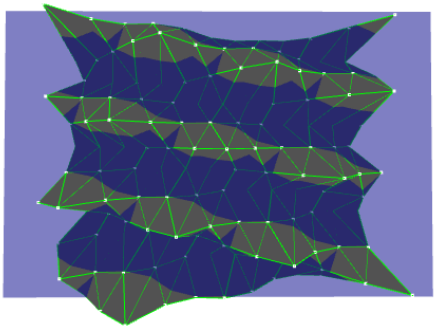
Consider the first row of Figure 8.4. Being F connected it is not surprising to observe the presence of one cornerpoint at infinity in each diagram. This cornerline retrieves the connectedness of F. Moreover, its birth-level gives an insight on the use of the pitch classes in the composition, by representing the height of the minimal subcomplex of the deformed *Tonnetz*. More information is retrieved considering



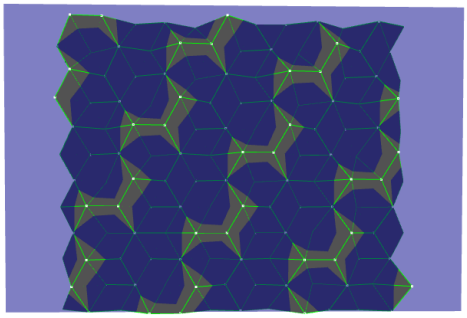
(a) KV 311, mov. 1 - Mozart.



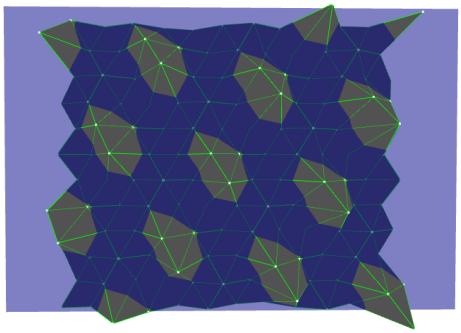
(b) KV 311, mov. 2 - Mozart.



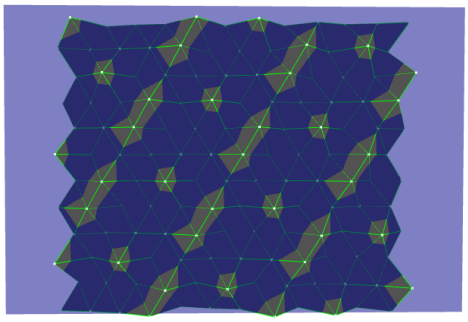
(c) KV 311, mov. 3 - Mozart.



(d) Klavierstück, I - Schönberg.



(e) Klavierstück, II - Schönberg.

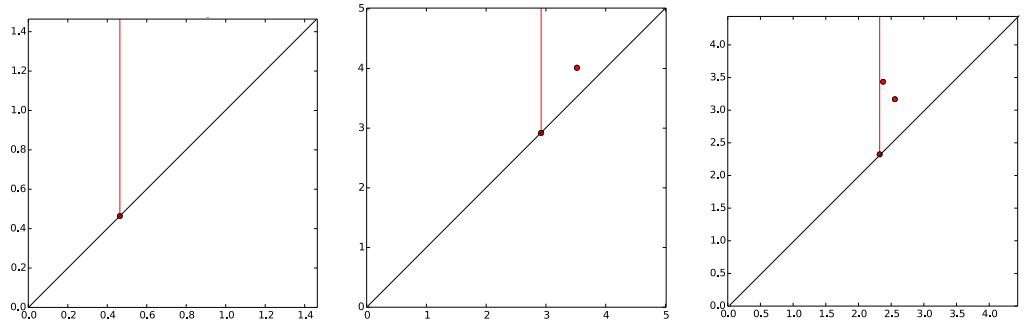


(f) Klavierstück, III - Schönberg

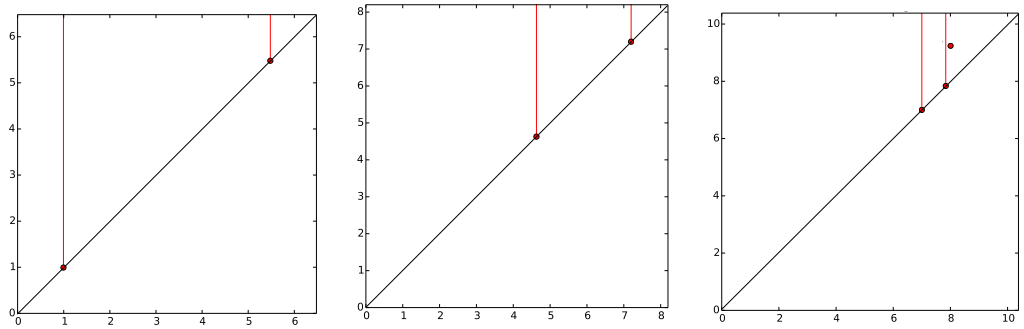
Figure 8.3: Sub-level sets of the height function (in blue) on several deformed *Tonnetze*. Different compositional styles are characterised by particular choices of pitch classes and durations.

the proper cornerpoints, that describe the lifespan of the connected components along the filtration. The three examples of Figure 8.4 present as many different configurations. In particular, *Arabesque* and *Jeux d'Eau* that were topologically equivalent for our first *naïf* classifier are now neatly distinguished by their persistence diagrams.

It is possible to give a musical interpretation of the 0-persistence diagram considering the birth-level of the cornerline, say $x = b$, and the proper cornerpoints $p \in C$. In our representation, b corresponds to the height of the minimal subcomplex



(a) Arabesque, Debussy. (b) Jeux d'Eau, Ravel. (c) Klavierstück I, Schönberg.



(d) Arabesque, Debussy. (e) Jeux d'Eau, Ravel. (f) Klavierstück I, Schönberg.

Figure 8.4: The 0 and 1-persistence diagrams representing the topological fingerprints associated to three different compositions.

of the deformed *Tonnetz*. If $b \approx 0$, there exist a pitch-class set that has not a relevant role in the composition, suggesting that it is based on a stable tonal or modal choice. On the contrary, if $b >> 0$ every pitch class has been used in the composition for a relevant time. This configuration corresponds to a more atonal or chromatic style.

The presence of more than one connected component has to be interpreted as the presence of two minimal subcomplexes respect to the height function. Hence, these subcomplexes are not connected by an edge in the 1-skeleton of F. Furthermore, the structure of the fundamental domain generated by major and minor third allows to retrieve a maximum of three connected components. To create this particular configuration it is necessary to play a chromatic cluster, for instance $C, C\sharp, D$. The same argument holds for the maxima of the height function, that will be discussed in the next paragraph.

Coming back to the three examples of Figure 8.4, we can interpret the low birth-level and the absence of proper cornerpoints in the diagram associated to *Arabesque* as an evidence of its pentatonic and diatonic modal inspiration (Trezise, 2003), we retrieve also the fact that the whole chromatic scale has been used during the composition. The cornerline in the persistence diagram associated to $Jeux d'Eau$ is characterised has been a higher birth-level compared to the one we analysed previously. Moreover, it is not surprising in this case to retrieve a proper cornerpoint, whose presence is justified by the *ante-litteram* use of the *Petrushka chord*, a superposition of a major triad and its tritone substitute, for instance

$G = (G, B, D) + C\sharp = (C\sharp, E\sharp, G\sharp)$. Finally, the diagram associated to *Klavierstück I* has a cornerline with a birth-level comparable with the one associated to *Jeux d'Eau*. In this case, two proper cornerpoints (that corresponds to two minima, forming a chromatic cluster with the one retrieved by the cornerline) point out the atonal nature of the composition.

1-persistent homology

Now, consider the second row of Figure 8.4. The common denominator of the three diagrams is the presence of two cornerpoints at infinity. These infinitely persistent homological classes retrieve the two generators of the torus $F \cong \mathcal{S}_1 \times \mathcal{S}_1$, see Figure 6.5. In musical terms, the value of birth-levels of the cornerlines and their distance are relevant and give a first characterisation of the style of the composition. We discuss four configurations of the cornerlines, in order to provide an intuition concerning the stylistic information they retrieve. Let $b_1 < b_2$ the birth-levels of the two generators, their distance is given by $d = |b_1 - b_2|$.

(a) $b_1 \approx 0$ and $d >> 0$ (Figure 8.4d). This configuration points out a tonal choice. A cycle representing one of the generator is born suddenly, this means that there exist a pitch-class set that has not been used in the composition. Hence, this feature suggest a precise choice in terms of tonality (or modulations among near tonal centres), or modality. The high distance between the cornerlines points out that a pitch class set is less used than the others in the composition, generating two maxima, as it is depicted in a representative surface of Figure 8.5a.

(b) $b_1 >> 0$ and $d >> 0$ (Figure 8.4e). An extensive use of the whole chromatic scale (both in terms of pitches and durations) is retrieved by the high birth-level of the first cycle. However a particular modal or tonal choice is highlighted by the presence of two distinguishable maxima (see Figure 8.5b).

(c) $b_1 >> 0$ and $d \approx 0$ (Figure 8.4f). This configuration represents an atonal compositional choice. The whole chromatic scale has been equally relevant during the composition and the average height of the vertices of the *Tonnetz* does not allow to distinguish any *preferred direction* generating a *compressed* surface as the one of Figure 8.5c. In the applications we present in the next section, we shall see how a tonal piece modulating on several tonal centres, and hence, using extensively the whole set of available pitch classes, is equipped with a structure, that allows to distinguish it from a serial or a ultrachromatic composition.

(d) $b_1 \approx 0$ and $d \approx 0$. This case that is not represented in the persistence diagrams we chose. In this case, there exists a pitch-class set that less relevant for the composition and two distinct maxima with a low minimum. In this case the composition can be classified as modal or tonal and based on a small set of precise musical ideas. See Figure 8.5d for a representative surface of this configuration.

More information is retrieved considering the proper cornerpoints of the persistence diagrams. These points retrieve the lifespan of other maxima, arising in different configurations by considering chromatic, dodecaphonic or serial compositions.

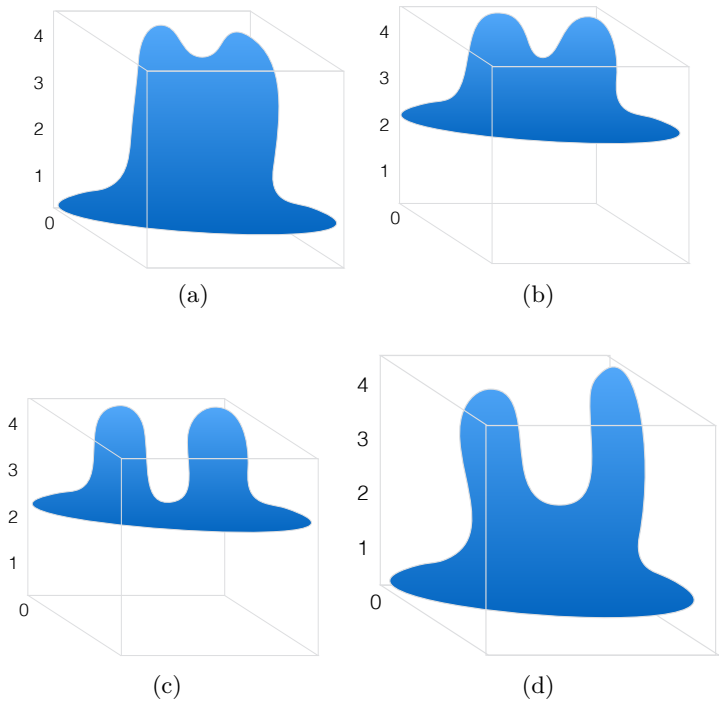


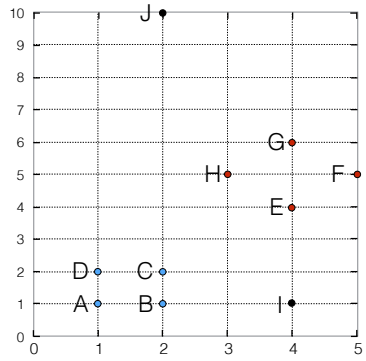
Figure 8.5: Smooth surfaces representative of four different configurations of the deformed *Tonnetz*.

8.2.2 Hierarchical persistent music clustering

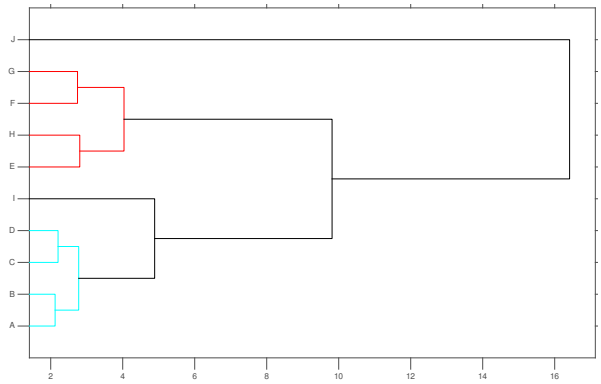
Let (D_∞, d_B) be the space of the persistence diagrams equipped with the bottleneck distance. We recall that a piecewise linear functions defined on finite simplicial complexes are tame. Thus the bottleneck distance is stable under small variation of f. A collection of k-persistence diagrams represents a point cloud in $P \subset D_\infty$. In particular, considering the collections of k-persistence diagrams associated to a set of pieces of music, it is possible to compute their pairwise bottleneck distance. Then, we shall describe the configuration of such point cloud through a hierarchical clustering analysis (Ott, 2009). This analysis gives a simple representation of all the possible clusterings between points, visualisable as a dendrogram.

Representation of data through dendrograms

Dendrograms provide an intuitive representation of the hierarchical clustering of data. We refer to (Langfelder et al., 2008; Martinez et al., 2010) for a complete description of these subjects. Consider the 2-dimensional data represented as points in $\mathbb{R}^2$ in Figure 8.6a. The data form two clusters and have two singletons labelled as I and J. The horizontal axis of the dendrogram represents the distance or the dissimilarity between clusters, while each object is represented by its label on the vertical axis. The information carried by the dendrogram concerns similarity and clustering of data. Each joining is represented by the splitting of a horizontal line into two horizontal line. The position of the split allows two retrieve the distance



(a) 2-dimensional data collection.



(b) Dendrogram.

Figure 8.6: Dendrogram representation of data dissimilarity. The structure of the 2-dimensional point cloud consists of two distinct groups and two outliers. The dendrogram reflects such a structure representing the two groups as separate clusters and joining the outliers to the clusters respecting their relative position respect to the configuration of the point cloud.

among two clusters. Observing the dendrogram in Figure 8.6b, one can see how the two main clusters are represented as branches occurring at about the same distance. The outliers are fused at much higher distances.

Computation. Consider a collection of n objects and let $D = d_{ij}$ be the matrix representing the distance among the clusters i and j, composed by n_i and n_j objects respectively. The dendrogram is computed as follows:

i) Find the clusters $\bar{\imath}$ and $\bar{\jmath}$ such that $d_{\bar{\imath}\bar{\jmath}}$ is minimum in D.

ii) Merge $\bar{\imath}$ and $\bar{\jmath}$ in a new cluster k with $n_k = n_i + n_j$ objects.

iii) Compute a new clusters distance matrix as

$$d_{kl} = a_{\bar{\imath}} d_{\bar{\imath}l} + a_{\bar{\jmath}} d_{\bar{\jmath}l} + b d_{\bar{\imath}\bar{\jmath}} + c|d_{\bar{\imath}l} - d_{\bar{\jmath}l}|.$$

Particular choices of the parameters distinguish among different algorithms. We shall utilize the *complete linkage* where $a_i = a_j = 1/2$, $b = 0$ and $c = 1/2$.

iv) Iterate the previous steps.

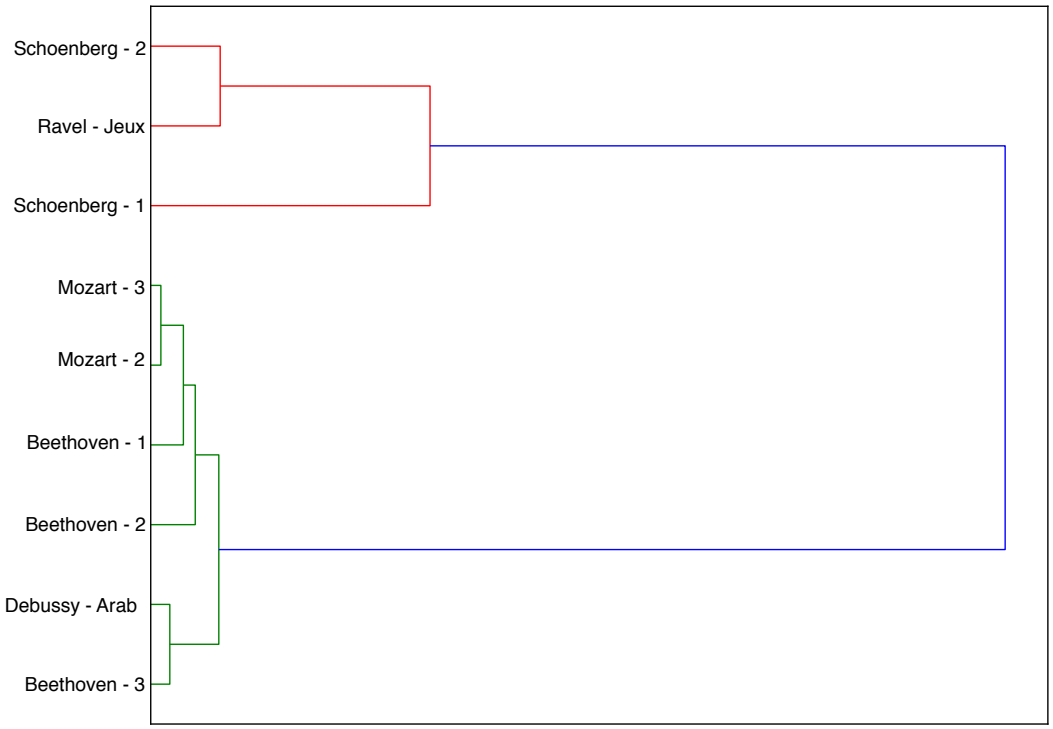


Figure 8.7: Persistence-based clustering of nine classical and contemporary pieces.

0-dimensional persistence

In the following examples 0-dimensional persistence has been used to classify different collections of music pieces. The examples are organised in order to show some possible applications of the topological characterisation of music: classification of tonal and atonal pieces, evaluation of different versions of the same jazz standard played by different improvisers and the discrimination of different styles in a pop music context.

Remark 15. To safely compare different pieces, the height function has been normalised. The collection of MIDI files used to generate the examples described in the following section is available at `http://nami-lab/experiments/midi-collection.com`.

Example 1. Tonal and Atonal Music. The first example we present is the hierarchical clustering of some of the pieces included in the web application for the visualisation of deformed *Tonnetze*. The nine pieces we analyse have been selected among the compositions by Beethoven, Debussy, Mozart, Ravel and Schönberg, in order to provide a heterogeneous dataset in terms of compositional style. The similarity among these pieces, computed considering the pairwise bottleneck distances of the persistence diagrams associated to the selected pieces, is depicted in Figure 8.7 as a dendrogram.

It is possible to observe how data are organised in two main clusters, segregating the two first pieces of Schönberg's *Drei Klavierstücke* and Ravel's *Jeux d'Eau*, from the ones by Mozart, Beethoven and Debussy. The association between the second

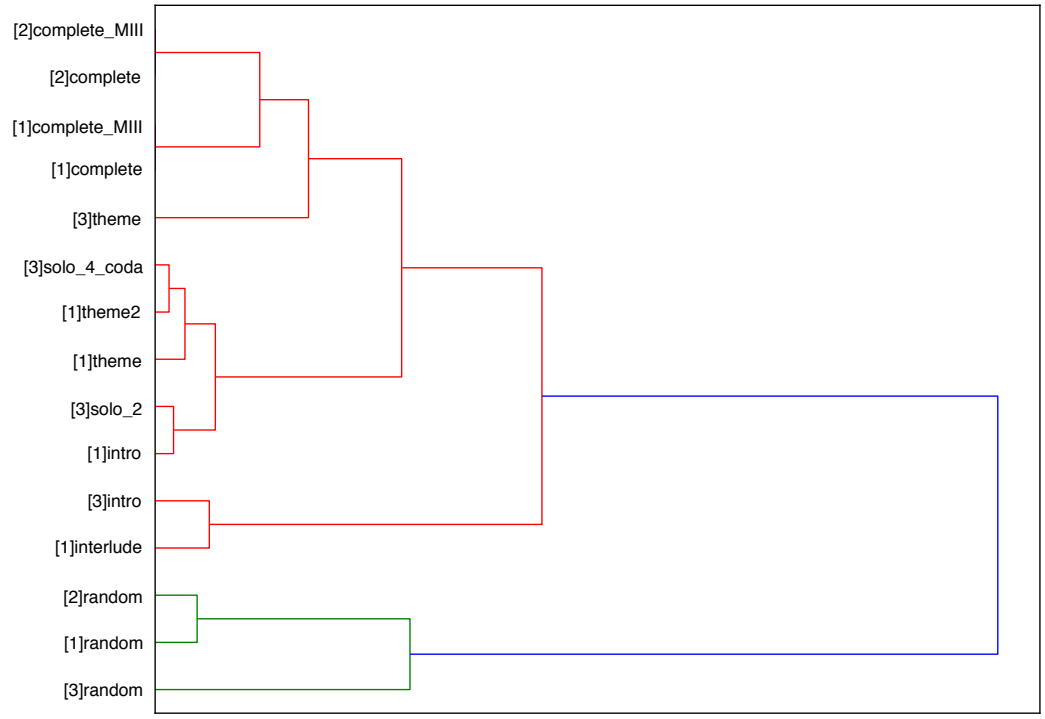


Figure 8.8: Comparing three different version of All the Things You Are.

piece of the *Drei Klavierstücke* with *Jeux d'Eau* respects what we found examining the weighted subcomplex of the *Tonnetz* in Section 6.1.2. Some tonal traces are hidden in this piece, albeit they are not evident to a human interpretation, as it is proven by the disparate tonal interpretations of these three pieces, see for instance (Brinkmann, 1969; William, 1984; Ogdon, 1981). Concerning the Ravel's composition, the *ante litteram* utilisation of the *Petrushka chord* highlight the atonal nature of the piece.

The two movements from Mozart's KV311 form suddenly a cluster reached at an increasing distance by the two first movement of the Sonata in C major by Beethoven, while the third movement is grouped with *Arabesque*, which is characterised by a generous use of the pentatonic scale, before joining the others.

Example 2. Comparing three versions of *All the Things You Are*. The aim of this test is firstly to investigate the distance between the 0-dimensional topological fingerprint of three versions of the same jazz standard; secondly to show the invariance of such fingerprint under (musical) transposition; and finally, to show the relationship between the fingerprint extracted by the whole piece of music and its segments. In addition, a randomized-pitches version of each song has been introduced in the dataset, to test the ability of persistent homology to distinguish between a piece modulating in several distinct tonalities[2] and enriched with chromatic solos, and a suite of random pitches without any apparent structure.

[2] A tonal harmonic analysis of the standard reveals it modulates in five different keys: $A\flat$ major, C major, $E\flat$ major, G major and E major.

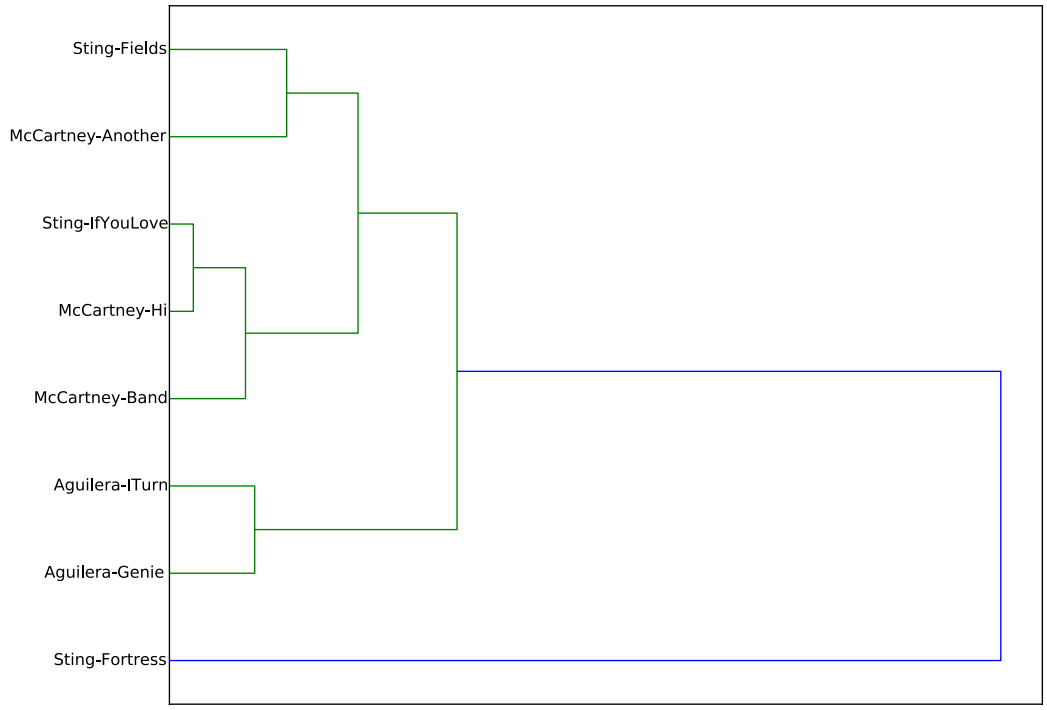


Figure 8.9: Pop clustering.

The three versions are labelled in the dendrogram of Figure 8.8 as [1], [2] and [3]. The number is followed by an attribute such as $[i]complete$ meaning the persistence diagram has been computed on the whole length of version i, or on its transposition $[i]complete\_interval$, or a segment as $[i]intro$. The three versions we considered are structured as follows:

a) version [1] is played by four instruments in a pretty standard way, with doubled harmony. The main section are a 3/4 introduction, a first exposition of the theme $[1]theme$, a 12 bars interlude labelled as $[1]interlude$, introducing the last theme containing short improvisations and embellishments.

b) [2] is performed by a piano solo, and it is characterized by a rich chromatic playing style of both hands in which the main theme is executed twice.

c) the last version we examined is performed by a trio version (piano, bass, percussion). Its structure consists of an introduction, an exposition of the theme, and a piano solo. We included two improvisations dividing them according to the structure of the standard, the introduction and the main theme.

It is not surprising to see that the transposed versions of the pieces have distance zero from the original ones. What is interesting to observe is that the randomized versions of the songs are well segregated from the rest of the dataset, as it is shown by the green cluster at the bottom of the figure.

Proceeding from bottom to top, we find a small cluster containing the interlude of the first and the introduction of the third version, which share a very similar

structures in terms of leaps and rhythm, see the partitions *[1]interlude* and *[3]intro* in appendix Appendix D.

Finally, it is possible to observe how in the top cluster the two complete songs are linked to the fragment of the third version containing the theme. Hence the 0-persistence homology retrieves the fragments containing the whole structure of the standard. This feature is surprising, taking into account the several modulations of the piece and that we are considering only the complex created by the whole segment. The first and second themes of the first version all clustered together with the last improvisation of the the third version of the standard, which is the one more respectful of the original theme.

Example 3: Pop clustering. The dendrogram obtained considering two songs by Christina Aguilera, and three songs by Sting and Paul McCartney respectively is depicted in Figure 8.9. Sting's *Fortress Around Your Heart* is represented as an outlier. A cluster contains the two songs by Christina Aguilera, that result well separated from the other cluster grouping Sting's and Paul McCartney's tunes. The position of the outlier is due to the hard modulations of the piece. For its harmonic transcription refer to `http://yalp.io/app/sting-fortress-around-your-heart-701`.

8.2.3 1-dimensional persistence

It is possible to explore dendrograms built utilizing the pairwise distances of persistence diagrams representing the behavior of the first homology group. In layman terms, measuring the persistence of 1-dimensional holes generated by the filtration, as its 0th counterpart discussed above, measures the lifespan of the connected components.

Example 1: Tonal and Atonal Music. We propose a new clustering of the nine pieces we analysed above. As we expected, the persistence of 1-cycles gives different results than its 0 counterpart. Figure 8.10 shows a dendrogram composed by two green clusters grouping the two movements of Mozart's Sonata no. 9 and Debussy's Arabesque and the first and third movements of Beethoven's Sonata no. 13 with *Klavierstück II* as an outlier. A red cluster is formed by the second movement of the Sonata no. 13 and *Jeux d'Eau*. Even in this case, the first of the three Schöberg's piano pieces is represented as an outlier of this last cluster. We still retrieve a classification of the different compositional styles. In this analysis, Beethoven and Mozart are represented by two different clusters and the atonality of the *Drei Klavierstücke* is expressed by the high dissimilarity of its two pieces. Such a dissimilarity is nuanced for the two compositions, as in the 0-persistence analysis, the first piece is the farther from the rest of the dataset.

Example 2: All the Things You Are. In Figure 8.11 the hierarchical clustering between the three versions of All the Things You Are we analysed above, is depicted in its 1-dimensional version. The invariance under transposition still holds, being the transposed version of [1] and [2] at distance 0 from the original ones. The random-pitch versions are still grouped together. The introductions of the first and third versions are represented as outliers. A homogeneous cluster recollect the first

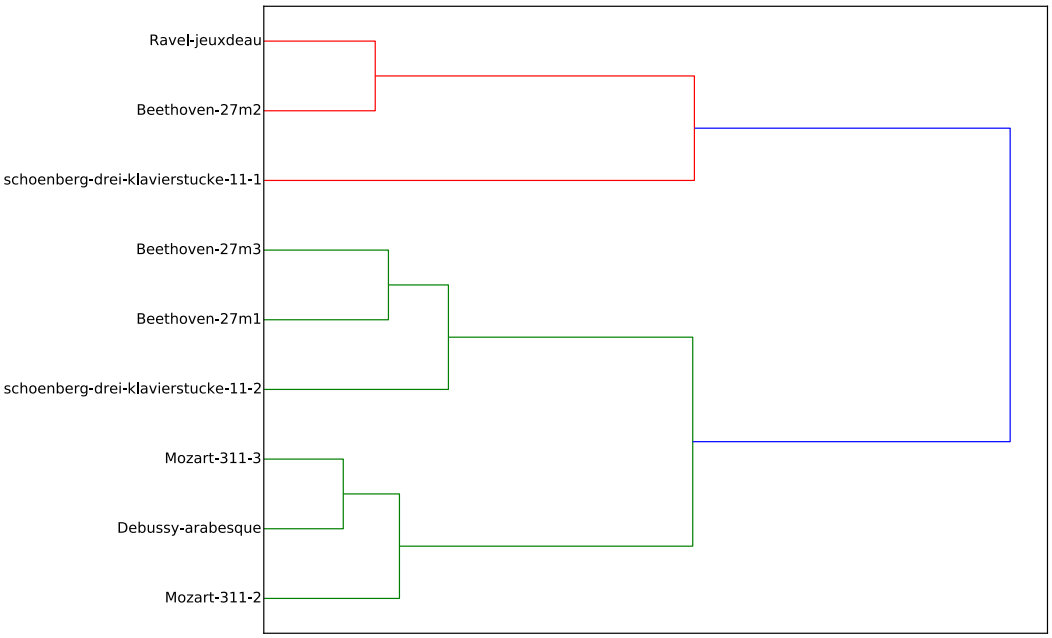


Figure 8.10: H_1 persistence-based clustering of nine classical and contemporary pieces.

version and its segments, denoting a sort of *robustness to occlusions* of the musical fingerprint. The similarity between the first and the second version is retrieved in this cluster. Finally, a solo of version [3] is grouped with its theme.

Example 3: Big Pop Clustering. Figure 8.12 shows a simplification of the clustering resulting by the comparison of 58 pop songs performed by 28 artists, spacing from Ray Charles to Lady Gaga. In order to give a simplified representation of this clustering we considered only the three big groups the algorithm found and listed on the left of each cluster the artists whose songs belong to that group. In particular names written in black bold characters are artists whose song are entirely grouped in the cluster at their right, while the artists' names written in red bold characters identify the three artists whose songs are spread along the three groups.

It is interesting to observe how the whole collection of songs by *Ringo Starr*, *Paul McCartney* and *Simon and Garfunkel* are grouped together in the blue cluster, with *Ray Charles*, *Stevie Wonder* and *George Benson*. At the same time it is admissible that the diversity characterizing Sting's compositions is mirrored by the positions of his songs in the dendrogram. The seconds and third clusters are less homogeneous, but promising, taking into account that so far songs are identified by a single persistence diagram.

Discussion

In this chapter we suggested a model describing music taking into account the contribution of each paring (pitch class, duration) associated to the notes of a composition. A filtration has been defined on the fundamental domain of the *Tonnetz*.

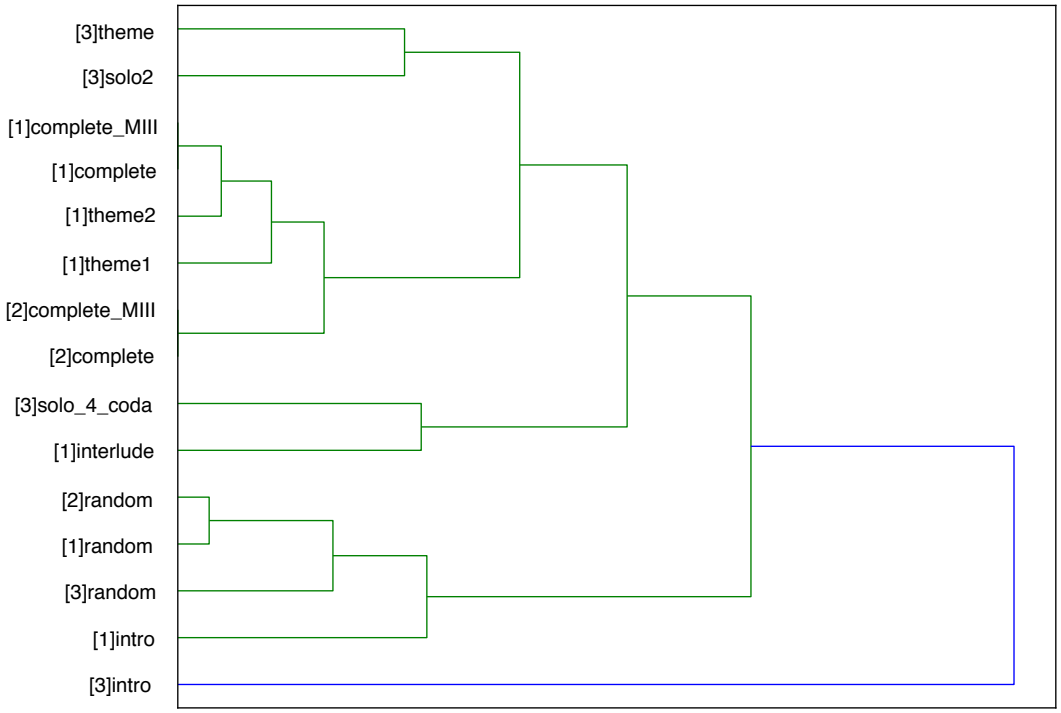


Figure 8.11: Comparing three different version of All the Things You Are using 1-dimensional persistence.

Such a filtration is induced by the height function defined on the vertices of $\mathcal{T}$. The k-persistence diagrams associated to different music pieces have been considered as point of a space equipped with the bottleneck distance. The possible clusterings of the points belonging to such dataset have been discussed and represented as dendrograms, showing that 0 and 1-persistence can be used to analyse and classify music. In particular, the stability of the bottleneck distance allows to generalise this construction from MIDI files to audio, as we shall discuss in the conclusions of the whole part.

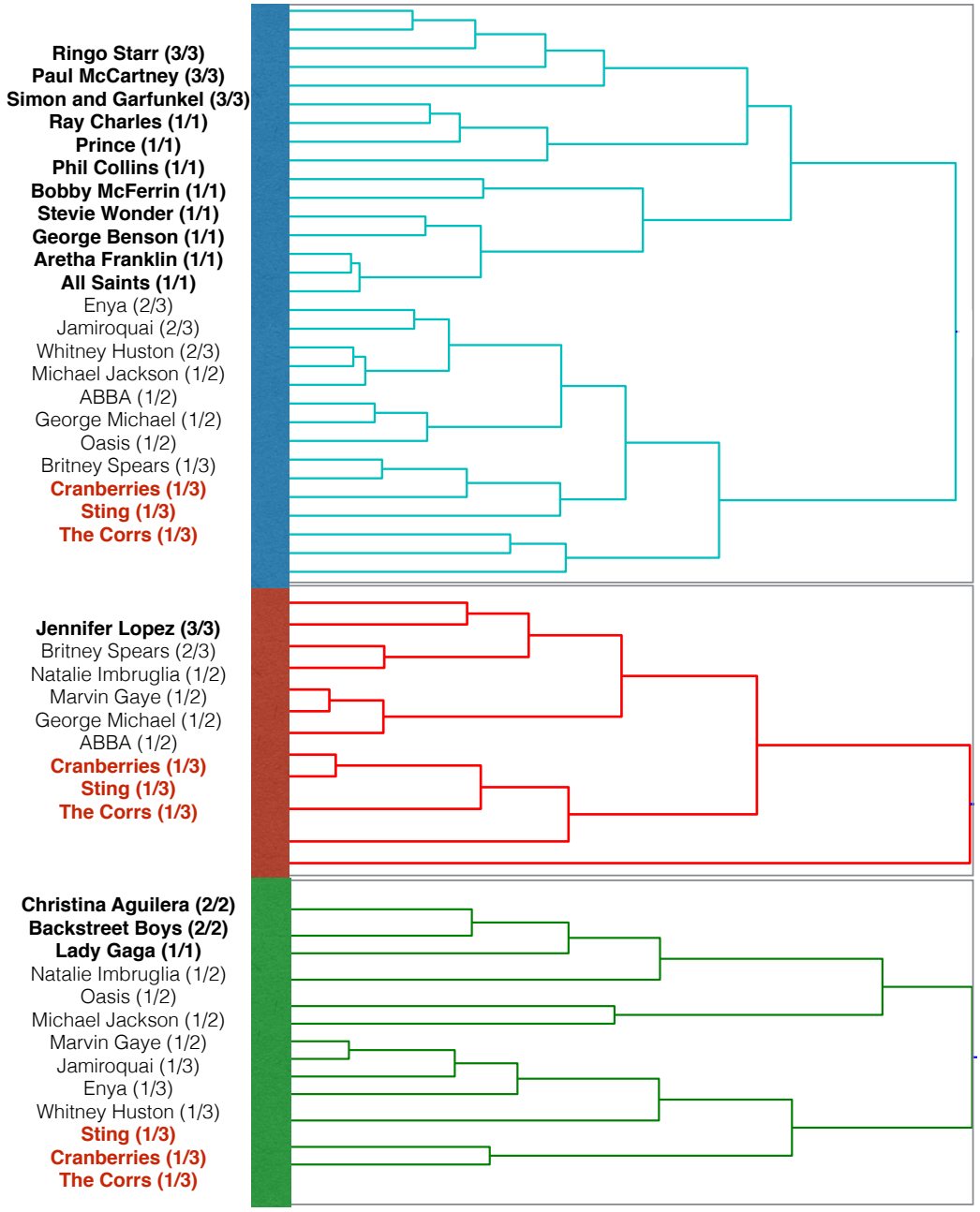


Figure 8.12: A simplified version of the clustering of 58 pop songs generated from their 1-persistence diagrams.

Nine

Audio feature deformation of the *Tonnetz*

The analysis of music can be considered from two different sides: a horizontal one naturally suggested by counterpoint and voice leading theory, and a vertical one given by the superposition of notes.

Nevertheless, a significant piece of information is carried by the signal. For instance, the *timbre* of the instrument we are listening to affects the perception of a whole piece, as the same phrase played on an acoustic piano or on a Rhodes will surely evolve in different ways in a composition, albeit keys and hammers are used in the same way by both instruments (Barona, 2014; Lee et al., 2009).

In this chapter, we suggest two applications based on the deformed *Tonnetz*, in which the function used to displace its vertices is derived form the signal domain. Hence, the height of each vertex is computed by considering an audio feature. In particular, we shall use the consonance function as it has been introduced by Plomp and Levelt. In a first part, it will be used to compute the displacement of the vertices of the *Tonnetz* labelled with the pitches belonging to a single octave and compared to a fixed pitch. This space shall be used to classify the 21 modal scales

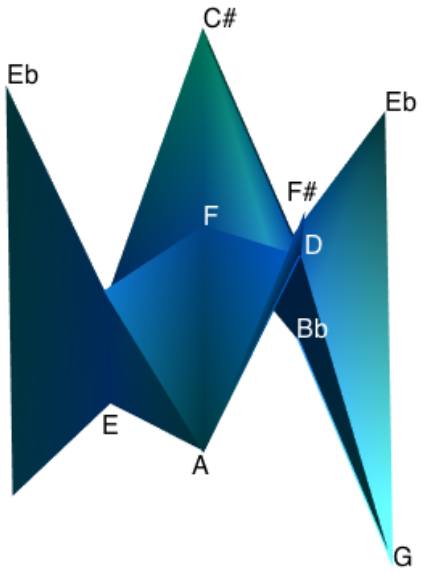


Figure 9.1: A *Tonnetz* deformed through a signal-based height function.

derived from the diatonic, melodic minor and harmonic minor scale. The octave dependency of the consonance function is highlighted as a fundamental feature, that musicians exploit in their compositions. As a second application, we will show how a trivial extension of the consonance function to chords can give interesting results, once interpreted in a geometrical metric context and in the formalism of topological persistence.

9.1 The calculation of consonance values according to Plomp and Levelt

The notion of consonance has a long history, dating back to the time of Greek philosophers. The notion itself is complex and has been given multiple meanings and explanations throughout the history of music theory and acoustics. For a detailed review of the history of consonance theory, the reader is invited to refer to (Sethares, 2004) and (Tenney, 1988).

The original idea of the Greeks, and of various philosophers and scientists such as Galileo (Galilei, 1638), Euler (Euler, 1766, 1739a), or Diderot, was that consonant intervals are those based on small frequency ratios as consonants, an idea which originated from Pythagoras. In the nineteenth century, departing from consonance theories based only on the musical objects at hands, Helmholtz introduced in his book *On the Sensation of Tone* (Helmholtz, 1877) a theory of *sensory dissonance* based on the processes at work in the auditory system. It is a well-known fact that two pure tones of close frequencies produce beats, the frequency of which is equal to the difference of frequency between the original pure tones. When the frequencies of the original signals are close, the beat frequency is low, and the slowly evolving resulting signal is not perceived as dissonant. Helmholtz observed that when the beat frequency increases, the roughness of the resulting signal increases, peaking at a maximum for a reported beat frequency of 32 Hz. Thus, the dissonance of a tone is directly linked to the presence or absence of beats. By considering the interaction of all the partials of two harmonic sounds, and by assuming that the total dissonance is obtained additively from the dissonance between two partials, Helmholtz was able to calculate the dissonance value of any interval.

In the mid 1960s, Plomp and Levelt (Plomp and Steeneken, 1968; Plomp and Levelt, 1965) published an influential experimental work on the sensation of consonance and dissonance for pure tones. A number of listeners were asked to rate the consonance of various pairs of pure tones sounded at different frequencies. This resulted in the determination of a dissonance function, which gives a dissonance value as a function of the frequency ratio of the two pure tones, expressed in units of the *critical bandwidth*. The typical plot of this function is presented in Figure 9.2. The notion of critical bandwidth derives from the mechanism of the auditory system itself (Fletcher, 1940). In the cochlea, pure sinusoidal tones excite different places of the basilar membrane. The place theory of pitch perception links the excitation place with the perceived pitch of the tone. When two tones of similar frequencies are sounded together, they excite similar places of the basilar membrane: in other terms, they occupy the same *critical band*. The width of this critical band is therefore linked to the ability to perceive two simultaneous tones of different frequencies as a

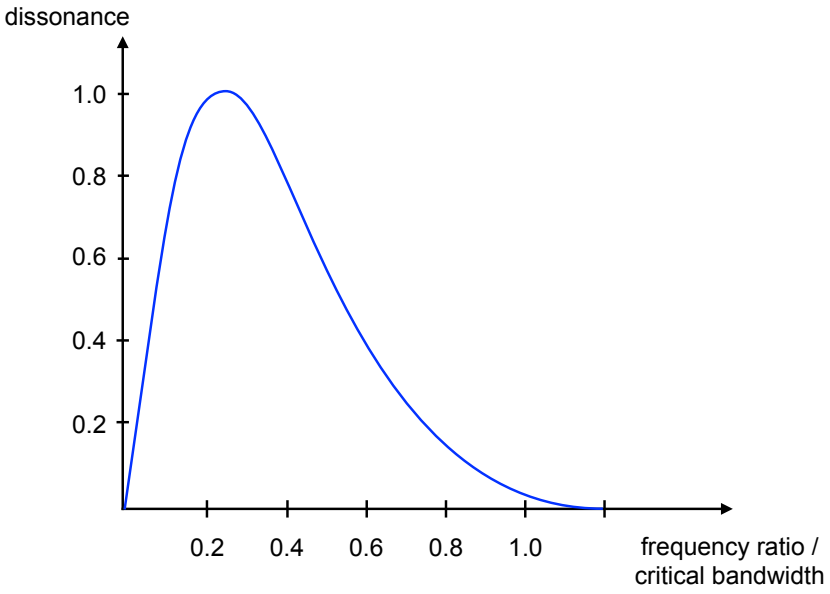


Figure 9.2: Plot of the consonance function between two sinusoidal tones, whose frequency ratio is expressed as a ratio of the corresponding critical bandwidth.

unique tone or not. The critical bandwidth is roughly constant and equal to 100 Hz in the range 100-1000 Hz, and then increases proportionally with frequency (Zwicker, 1961; Zwicker and Terhardt, 1980).

The results of Plomp and Levelt provide an experimental justification of the work of Helmholtz, with the addition that the maximum of dissonance occurs at roughly one quarter of a critical bandwidth. It is therefore dependent on the frequency of the tones and not fixed to 32 Hz, as was the case for Helmholtz (which is incidentally the value of one quarter of a critical bandwidth for a frequency of roughly 600 Hz).

Multiple parametrizations of the Plomp and Levelt curve have been given by various authors. We use here the parametrization used by Sethares in (Sethares, 2004), wherein the consonance function between two pure tones of frequencies f_1 and $f_2 > f_1$ is given by

$$d(f_1, f_2) = \exp(-3.5 \cdot s \cdot (f_2 - f_1)) - \exp(-5.75 \cdot s \cdot (f_2 - f_1)),$$

where s is defined as

$$s = \frac{0.24}{0.021 \cdot f_1 + 19},$$

and is introduced to account for the variation of the critical bandwidth with frequency.

Based on their work on pure tones, Plomp and Levelt then studied the consonance of complex tones. Since a complex tone has a spectrum consisting of multiple partials, they assumed that the total consonance results from the addition of the consonance values between all pairs of distinct partials (under the hypothesis that all partials have the same intensity). In other terms, given a complex tone whose spectrum is a set $\{f_1, f_2, \ldots\}$ of partials at frequencies f_i, the total consonance is given by

$$D = \sum_{f_i} \sum_{f_j > f_i} d(f_i, f_j).$$

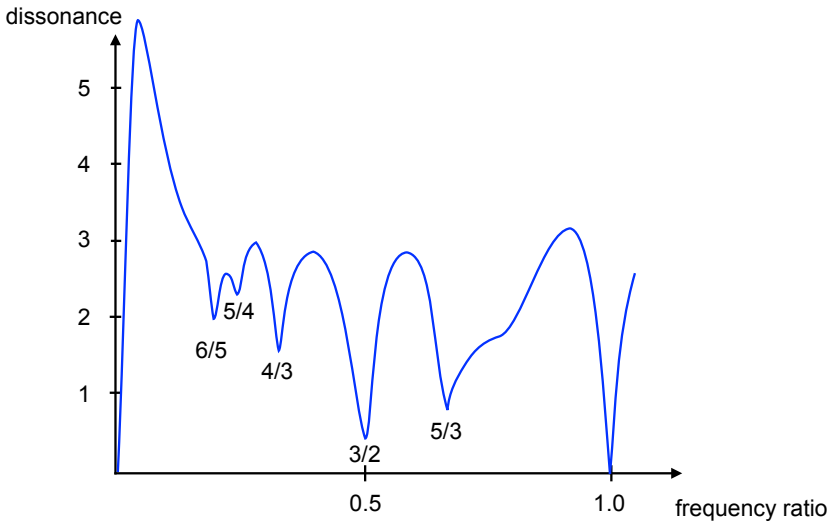


Figure 9.3: Plot of the consonance value of a complex tone consisting of the superposition of two tones with six harmonic partials of identical intensities, as a function of the frequency ratio of the two tones, the first one having a fixed frequency.

Using this definition, they calculated the consonance value of a complex tone consisting of the superposition of two tones as a function of their frequency ratio, the first tone being fixed. Each tone had a spectrum consisting in six harmonic partials of identical intensities. The plot of this consonance value is given on Figure 9.3. As can be seen on this graph, minima of the dissonance function are obtained for pure intervals such as the unison $(1:1)$, the octave $(2:1)$, the fifth $(3:2)$, the fourth $(4:3)$, the major third $(5:4)$, the minor third $(6:5)$, and the major sixth $(5:3)$.

Of course, the calculation of the consonance value of a complex tone is not limited to harmonic sounds. Sethares (Sethares, 2004) has investigated complex tones whose spectrum is inharmonic, and has deduced the corresponding consonant intervals between such sounds. In the following sections, we explore the use of consonance calculations to determine the hierarchical organization of various musical entities.

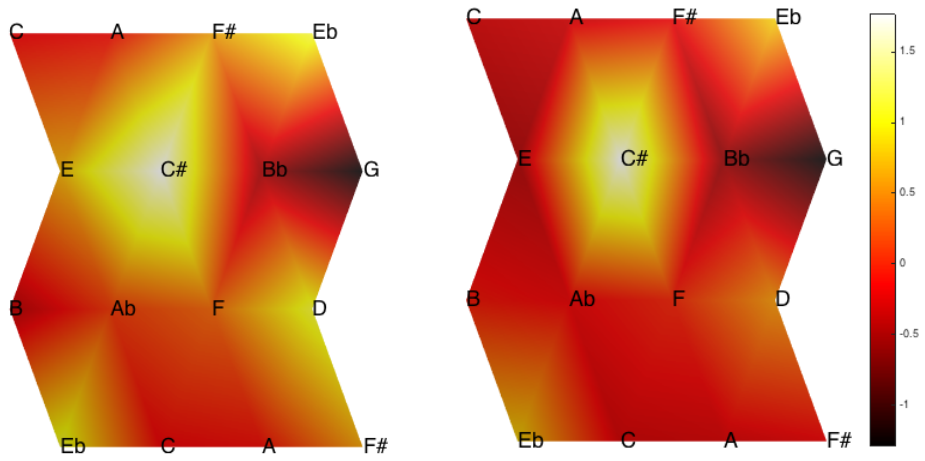
A consonance-based height function

Maintaining the notation of the previous section, let $p \in \mathbb{R}$ be a pitch. Define $h_p : V \to \{p_i\} \times L_i \to \mathbb{R}$ as

$$h_p(v) = d\left(p, l\left(v\right)\right),$$

where $d : \mathbb{R}^2 \to \mathbb{R}$ is the consonance function[1] and $l : V \to L_i$ is the labelling function associating to the vertices of the *Tonnetz* the chromatic scale built on the ith octave of the piano, with root r, such that $[r] = [p]$. In the remainder of this section we shall refer to p as the *reference pitch* used to compute the displacement

[1]Here we identifies a pitch with its fundamental frequency, see Equation (2.2.1) for the formula associating fundamental frequencies to pitches.



(a) $\mathfrak{T}_3^4$. (b) $\mathfrak{T}_3^5$.

Figure 9.4: Deformation of a portion of the *Tonnetz*. The reference note used to displace its vertices is C_3. The labels associated to the *Tonnetz*'s vertices correspond to the chromatic scale built on the fourth and the fifth octave of the piano.

of the vertices of the *Tonnetz*. Let

$$W : V \subseteq \mathbb{R}^3 \to \mathbb{R}^3$$
$$v \mapsto (x_v, y_v, h_p(v)),$$

be the function that defines the height of every $v \in V$. Such a height corresponds to the consonance value of the interval $(p, l(v))$. It should be noted that, for a transposition of the reference pitch, the consonance value decreases when the frequency of its reference pitch increases. In order to compensate for this effect, we renormalise the frequencies to the reference tone.

Variable geometry

The space we defined above is endowed with two interesting properties. First, the evenness of the equal temperament assures that the computation of the consonance is robust modulo uniform transposition of the reference pitch and the chromatic scale. That is to say, the intervals $(C_3, C\sharp_3)$ and $(D_3, D\sharp_3)$ share the same consonance value. Second, the consonance function is not invariant modulo octave. Respecting the common sensory experience, an interval of minor second $(C_4, D\flat_4)$, is less consonant than a minor ninth $(C_4, D\flat_5)$[2]. Hence, the geometry of the *Tonnetz* varies depending on the choice of both the chromatic set of pitches associated to its vertices and the choice of the reference pitch p.

The surfaces resulting from the deformation of the planar *Tonnetz* computed with C_3 as reference pitch and the pitches of the chromatic scale of the third and fourth octave of the piano are depicted in Figures 9.4a and 9.4b respectively. We will denote these surfaces as $\mathfrak{T}_3^3, \mathfrak{T}_3^4$. In the figure, a height function highlights

[2]Behaviour of the dissonance function respect to octave changes: $d(C_2, C\sharp_2) = 1.7$, $d(C_2, C\sharp_3) = 0.9$ and $d(C_2, C\sharp_4) = 0.4$.

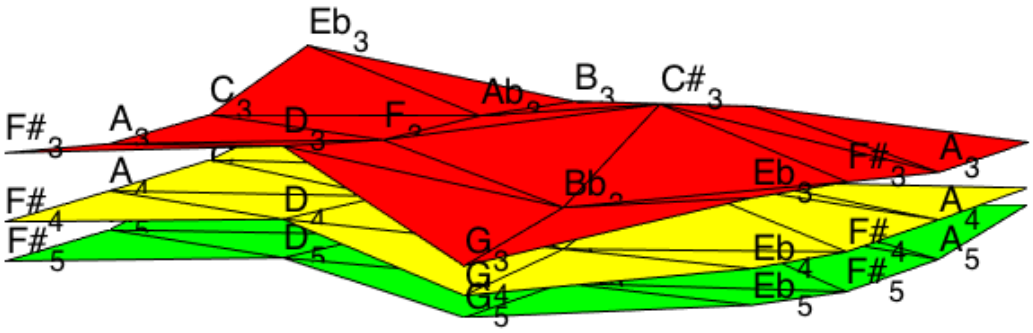


Figure 9.5: Variations of the *Tonnetz*'s geometry on three octaves.

maxima and minima of each surface. As in the application examined in the previous chapter, what we gain is the metric nature of this representation and the information given by the interaction between the structure of the *Tonnetz* and the deformation. In Figure 9.5 three different states of the geometry of $\mathcal{T}$ are depicted.

9.2 Persistent homology and audio feature deformed *Tonnetze*

While listening to a bass line or a whole harmonic sequence, it is a common experience to imagine a melody matching with the progression of notes or chords we are listening to. This melodic choice can be represented by a set of tensions and a set resolutions having the bass (or the chord) as a reference. One possibility is to build from such a choice a 7-notes, octaviant scale[3], that superposed to the reference bass creates a recognisable sonority, called a *mode*. The tensions and the resolutions of a mode give to a trained listener the whole information he or she needs to recognise it. Thus, the space we propose is particularly suitable for the representation of modes, since tensions and resolutions are nuanced by the height function defined on the vertices and the whole scale is described as an extended shapes of the *Tonnetz*.

Exploiting this characteristic, we classify the 21 modes derived form the diatonic, melodic and harmonic minor scales (see Table A.1) by considering the point cloud of vertices of the consonance-*Tonnetz* and computing their persistent homology. In the next section we introduce the construction that we will use to associate a filtration to these point clouds.

9.2.1 Persistence for point clouds

Let $P \subset \mathbb{R}^n$ be a point cloud. There are two main constructions used to associate a simplicial complex to P, called the Čech and the Rips complex. The former is defined as an abstract simplicial complex denoted by $\mathcal{C}_P(r)$, such that its 0-simplices are exactly the vertices of P and its simplices are generated whenever the balls of radius r centred on its vertices have non-empty intersections. In symbols, we have

[3]The pitches composing the scale are contained in a single octave.

$$\mathcal{C}_P(r) = \left\{ \sigma = [p_0, \ldots, p_k] \;\middle|\; \bigcap_{i=0}^{k} B_r(p_i) \neq \emptyset, \right\},$$

where $p_i \in P$ for every $i \in \{1, \ldots, k\}$.

Given a point cloud P, we have that $\mathcal{C}_P(r) \subseteq \mathcal{C}_P(q)$ if $r < q$. Hence, it is possible to build a filtration of the Čech complex choosing a sequence $\{r_1, \ldots, r_n\}$ of increasing *radii* and setting $\mathbb{X}_i = \mathcal{C}_P(r_i)$.

An important feature of this construction is that the homology of the Čech complex is exactly the one given by the union of r-balls centred on the points of P. This is a straight-forward consequence of the nerve lemma (the interested reader is referred to (Kozlov, 2007)).

Definition 9.2.1. Let $\mathbb{X}$ be a topological space and $\mathcal{U} = \{X_i\}_{i \in I}$ be a covering. The *nerve* of $\mathcal{U}$ is the (abstract) simplicial complex $\mathcal{N}(\mathcal{U})$, whose set of vertices is given by I and such that a finite subset $S \subset I$ is a simplex of $\mathcal{N}(\mathcal{U})$ if and only if $\cap_{i \in S} X_i$ is nonempty.

Lemma 9.2.1. *Let $F = \{C_1, \ldots, C_n\}$ be a finite family of closed set, such that every intersection between its members is either contractible or empty. Then, the nerve of F and the union of sets in F have the same homotopy type.*

Balls in $\mathbb{R}^d$ satisfy the hypothesis of the Nerve lemma, and as an immediate consequence we have that for every point cloud $P \subset \mathbb{R}^n$,

$$H_k\left(\mathcal{C}_P(r)\right) \cong H_k\left(\bigcup_{p \in P} B_r(p)\right),$$

for $k \in \mathbb{Z}$.

The Čech complex is an object of difficult computation, thus it is often substituted by the Vietoris-Rips (or simply Rips) complex $\mathcal{R}_P(r)$. A simplex is added to the Rips complex, when all pairs of points representing its vertices are less than $2r$ distant.

$$\mathcal{R}_P(r) = \{\, \sigma = [p_0, \ldots, p_k] \mid \|p_i - p_j\| \leqslant 2r, \, \forall\, i, j \,\}.$$

Although the Vietoris-Rips complex of a point cloud does not share the homotopy type of the union of the balls built on its vertices, it is largely used for its computational ease. The error in the approximation of $\mathcal{C}_P(r)$ with $\mathcal{R}_P(r)$ is bounded by

$$\mathcal{R}_P(r) \subseteq \mathcal{C}_P\left(\sqrt{2}r\right) \subseteq \mathcal{R}_P\left(\sqrt{2}r\right).$$

9.2.2 Deformed *Tonnetze* for modern modes classification

Let the mode $M = (p, \{\, r, m_2 \ldots, m_7 \,\})$ be the couple composed by the reference pitch p and the set of pitches corresponding to a modal scale. This section's aim is to provide a characterisation of the modal scales, considering the 3-dimensional point cloud generated by the vertices labelled as $\{r, m_2, \ldots, m_7\}$ on the fundamental domain $\mathcal{F} \subset \mathcal{T}$.

Methodoloy

The procedure we use to classify modes is similar to the one that allowed us to deal with musical compositions in the previous chapter:

(i) The *Tonnetz* is deformed according to the consonance values induced by the reference pitch p and the chromatic scale built on the same octave than p.

(ii) The point cloud $\mathcal{M}_0$ is extracted from the the 0-skeleton of $\mathcal{F}$.

(iii) According to the definition given in Section 9.2.1, we compute the 0-persistent homology of the point cloud, considering the filtration induced by the Rips complex. Such a filtration is sensitive to the relative distance between the points composing the cloud, whose configuration depends both on the dissonance function and on the structure of the *Tonnetz*.

(iv) Finally, the similarity between the point clouds is represented as persistence diagrams and visualised in a dendrogram.

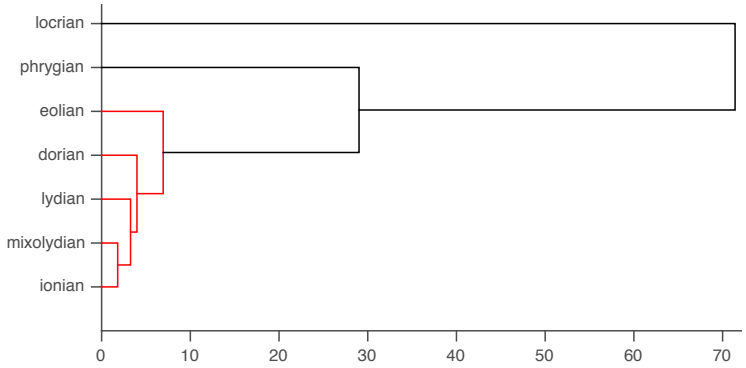
9.2.3 Applications

Scale-wise classification of modes

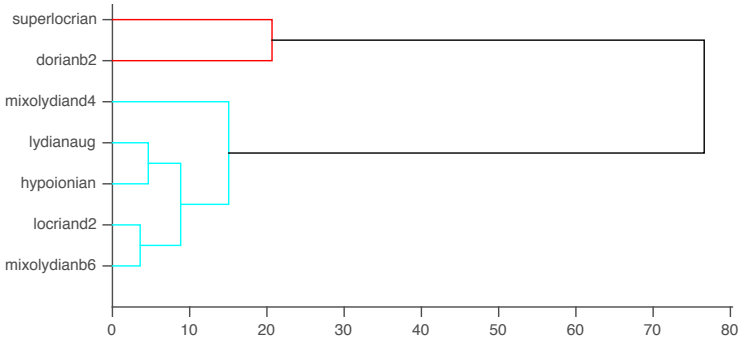
As we stated above, 21 modes can be deduced considering the degrees of the major, melodic minor and harmonic minor scales. In Figure 9.6 the dendrograms representing the hierarchical organisation of modes, induced by the distances of their 0-persistence diagrams is depicted *scale-wise*. As a first remark, observe how no sonorities have 0 distance. Thus, this representation grasp the different tension/resolution sets of each mode.

In Figure 9.6a the modes deduced from the major scale are considered. The only mode associated to a half-diminished (its root note forms a half-diminished chord with the third, fifth and seventh degree of the modal scale) chord is segregated from the others. The two more similar point clouds are the ones associated to the ionian and the mixolydian modes. The lydian scale, characterised by the presence of an augmented fourth and a major seventh, is separated from the ionian and mixolydian modes. However, these three major modes are represented as a homogenous cluster. Further from this three modes, we find the dorian and the eolian sonorities. The phrygian and the locrian modes are a minor and a diminished mode respectively and are represented as outliers. Both contain a minor second and are the most tense and recognisable sonorities of this dataset.

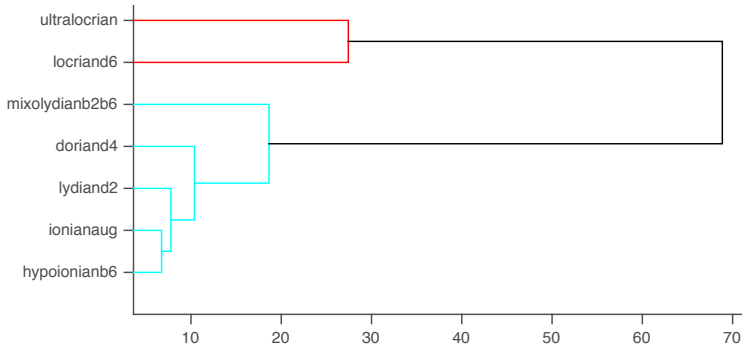
Figures 9.6b and 9.6c show the clusterings of the modes derived form the melodic and harmonic minor scales, respectively. In both cases the shapes result divided into two main clusters, one of them consisting at least of a mode built on a diminished triad. The bigger cluster of Figure 9.6b is formed by two pairs of modes grouped together: mixolydian♭6 - locrian♯2 and hypoionian - lydian augmented respectively. The mixolydian♯4, which is considered a *blues* mode (containing an augmented fourth on a dominant chord) is a well characterised *sonority*, and it is segregated from the other modes. Mirroring the structure of the clustering associated to the major scale's modes the last mode of the bigger cluster in Figure 9.6c is the mixolydian♭2♭6, also known as Spanish phrygian and the most identifiable mode of this dataset.



(a) Modes deduced from the major scale.



(b) Modes deduced from the melodic minor scale.



(c) Modes deduced from the harmonic minor scale.

Figure 9.6: Hierarchical clustering of modes interpreted as point clouds of the consonance-deformed *Tonnetz*.

An overview on the organisation of modes

Here we consider the grouping induced by the 0-persistence representation of the whole collection of modes we considered. The dendrogram representing this clustering is depicted in Figure 9.7.

From top to bottom we find the ultralocrian and superlocrian modes grouped together. Both are diminished modes and are generally considered the two most tense sonorities among the ones we analysed. In the second cluster are grouped three modes characterised by the presence of the minor second. The modes composing

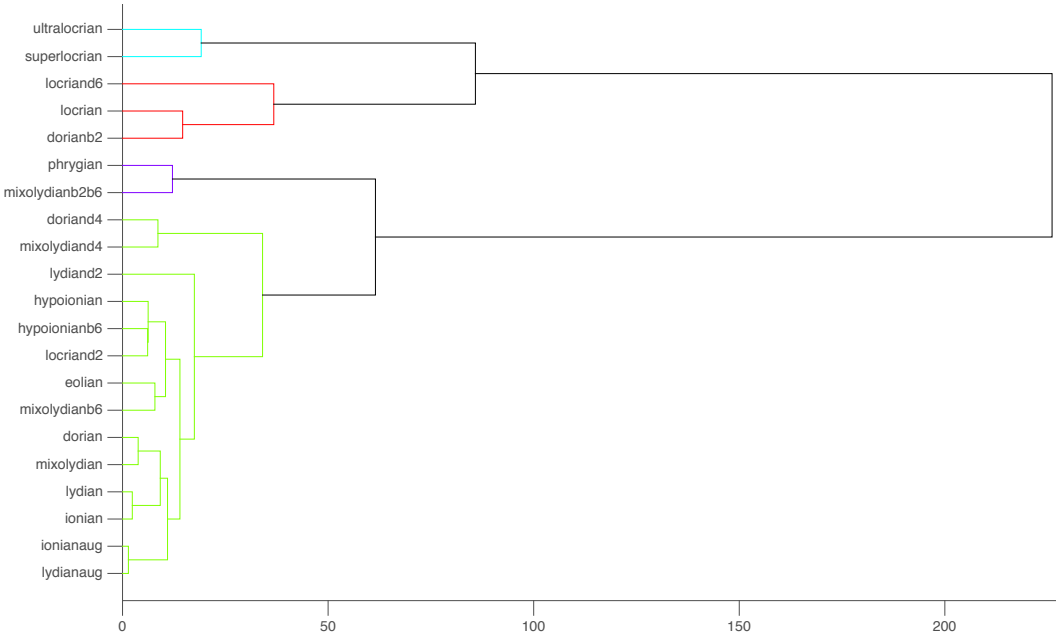


Figure 9.7: Hierarchical clustering of the 21 modes of Table A.1.

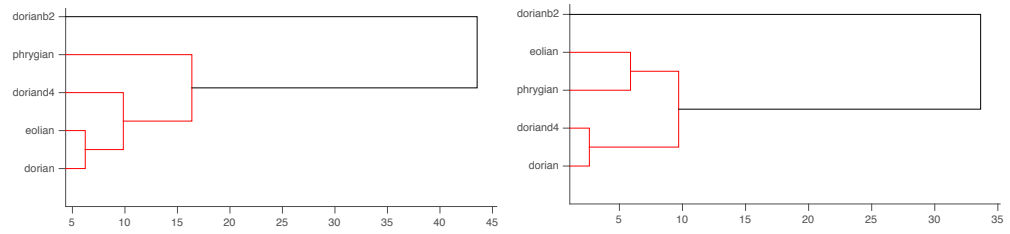
the third cluster are also equipped of a minor second but among the modes we considered, they are the ones associated to *Spanish* sonorities.

The other 12 modes are clustered together. Taking a closer look to this bigger cluster we observe an interesting class of groupings: (dorian♯4, mixolydian♯4), (eolian, mixolydian♭6) and (dorian, mixolydian). The modes composing these pairs (that are built on a major and on a minor triads respectively) are commonly interchanged in jazz and fusion melodic phrasing on dominant chords: the minor third of the dorian and eolian sonorities provides the *blue note* typically used in such contexts. Thus, this representation is coherent from a harmonic and melodic viewpoint. It is also interesting to notice how, despite the augmented fourth, the lydian and ionian modes are represented as similar sonorities. These clusters represent well superposition of scales used commonly in jazz composition and improvisation.

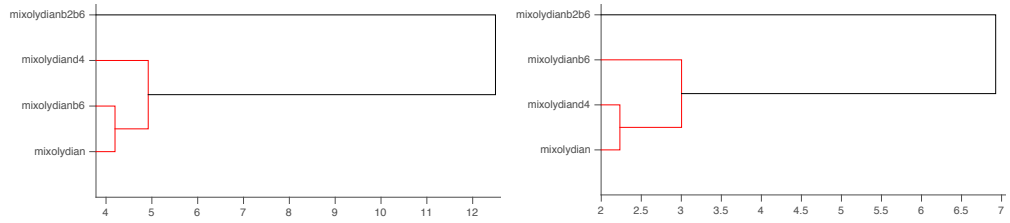
Octave dependency of the consonance function

It is well known in musical practice, that a (altered) chord sounds more consonant in open than in root position, or that a bass player prefers to play the 12th rather than the 3rd of a chord while accompanying. What we expect analysing the modal point clouds played an octave higher than the accompanying bass, is a smaller distance between their persistence diagrams, and hence a fusion of the clusters in the dendrograms.

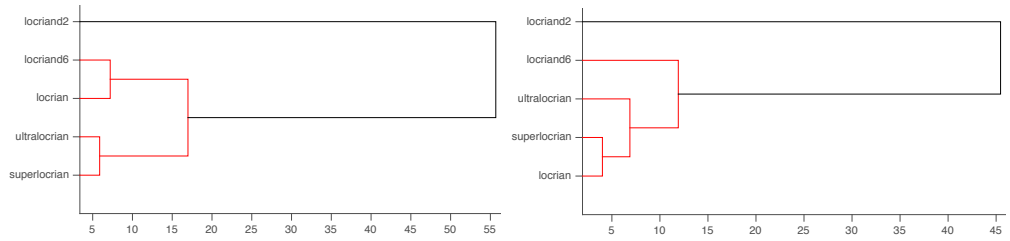
The clusterings representing the distance between modes built with pitches belonging to two different octaves are depicted in Figure 9.8. In this case we divided modes in three groups, consisting of minor seven, dominant and diminished modes, respectively. This subdivision has been obtained considering the chord built on the root of the modal scales.



(a) Minor seven modes.



(b) Dominant modes.



(c) Diminished modes.

Figure 9.8: Octave dependency of the harmonic-oriented modes clustering. On the right the organisation of modes represented as point clouds of $\mathcal{T}_3^3$, on the left their counterparts in $\mathcal{T}_3^4$.

First, we remark that the maximal distance between the shapes decreases when considering point cloud derived from $\mathcal{T}_3^4$. Consider the two clusterings associated to the minor seven modes in the first row of the figure. The dorian♭2 is an outlier: the tensions of the modes on the simplicial structure of the *Tonnetz* makes the point cloud distinguishable from the others even when the scale is played an octave higher. On the contrary, as we expected the phrygian is not an outlier in the diagram on the right. The minor ninth is less tense than a minor second and it is the only notes that differs between the eolian and the phrygian scales.

The dendrograms representing the distances between dominant modes segregates the mixolydian♭2♭6 which is the most recognisable sonority among them. It is interesting to notice how this is the only diagram which does not change its shape passing from one octave to the other. Containing the tritone (the interval between the third and the seventh of a dominant chord measures three whole steps) dominant chord are structurally tense. This feature makes their configuration invariant modulo octave. The same phenomenon occurs for the locrian♯2 in the last row of the figure.

9.2.4 Discussion

We used an audio feature to generate a deformation of the structure of the *Tonnetz* and then considered the point cloud generated by a vertical displacement of its vertices. The analysis of the Rips complex built on the point cloud representing the modal scale allowed to compare different melodic patterns generated by an accompaniment and a scale. This paradigm can be extended to the analysis of more than one octave superposing different deformed fundamental domains of the *Tonnetz* as it has been shown in Figure 9.5. Persistent homology has been used to provide a quantitative analysis of different sonorities, proving that our representation is suitable for the classification of different tension/resolution patterns. Finally, despite the model we proposed is limited by the choice of a particular harmonic spectrum for the computation of the consonance, the stability of persistence diagrams, assures that small variation of the consonance function shall correspond to small variation of the diagrams.

9.3 *Tonnetz* deformation through triads' consonance

It is possible to generalise the dissonance function defined on intervals to chords. In this section we present a comparison between the consonance function evaluated on six different families of triads.

As a first application, we study the clustering of triads of different types having varying their root on the whole chromatic scale and considering two different harmonic spectra. Thereafter, for a fixed triad, we will consider the dissonance generated by the block voicings composed by the superposition of the triad and each pitch of the chromatic scale an octave higher than the root of the triad. In particular, we will use a harmonic spectrum composed by six equal partials, in order to highlight the *pathological* behaviour of the consonance function.

Again, this computation will be used to define the displacement of the vertices of the *Tonnetz*. The surfaces created by varying the triad's class, shall be analysed by utilising both their metric properties and classified computing their persistent homology, for several filtrations.

9.3.1 The consonance function for triads

We consider here six different classes of chords, namely: the major, minor, augmented, diminished, suspended fourth, and suspended second triads. These chords are built using fifths, fourths, major thirds, and minor thirds. The list of these chords, along with their usual notation and a representative pitch-class set, is presented on Table 9.1. Each triad is considered in root position as composed by pitches belonging to the third octave of the piano. For each type of triad, we consider the twelve different triads obtained on the twelve different roots in the set of pitch-classes

$$S = \{C^4, C\sharp^4, D^4, D\sharp^4, E^4, F^4, F\sharp^4, G^4, G\sharp^4, A^4, A\sharp^4, B^4\}.$$

The consonance of the various triads is calculated using the theory of Plomp and Level as exposed in Section 9.1. For each tone with a given frequency, an harmonic spectrum consisting of six partials is generated. The final consonance value of a chord

Triad type name	Notation	Representative pitch-class set
Major	M	{C, E, G}
Minor	m	{C, E♭, G}
Diminished	∘	{C, E♭, G♭}
Augmented	aug	{C, E, G♯}
Suspended second	sus2	{C, D, G}
Suspended fourth	sus4	{C, F, G}

Table 9.1: Names of the studied triads and their corresponding representative pitch-class set.

is calculated by evaluating the individual consonance for each pair of frequencies between the partials of all tones. It should be noted that, for a given chord type, the consonance value decreases when the frequency of its root increases. In order to compensate for this effect, we have renormalized the frequencies to the C^4 reference tone.

The triads' consonance-hierarchical organisation

Once the calculation of the consonance value of each chord has been performed, a distance matrix between chords is obtained, wherein the value of each entry (i, j) is equal to the difference of the consonance values associated to the chord j and the chord i. Figure 9.9 shows two distance matrices representing the consonance relationships among triads, computing by using the harmonic spectra $h_1 = (1, 1/2, 1/3, \ldots, 1/6)$ and $h_2 = (1/3, 1/5, 1, 1/6, 1/3, 1/6)$, respectively. Notice that in both distance matrices, all the block on the diagonal have zero values, which is a direct result of the nature of equal temperament: since all intervals have an equal size, the consonance value of a triad of a given type is therefore independent of its root. Moreover, notice how the consonance function depends on the harmonic spectrum by considering the two distance matrices.The colours associated to each block of the matrices describe the gain and loss of consonance passing from one class to another. Consider the first column of each matrix. The one associated with a decreasing spectrum tells us that passing from a major triad to another class we always lose consonance (the block matrices associated to these classes are red). The same occurs for the first column of the second matrix. However the harmonic spectrum where the third harmonic (the fifth of each note composing the chord) is more powerful the the others, alters in an obvious way the perception of the triads. For example, minor and suspended fourth triads share the same consonance value.

Each distance matrix allows us to calculate a corresponding dendrogram, which illustrates the hierarchical clustering of the different triads. In Figure 9.10 we show how a distance-based clustering represents the triads classes as six different clusters. In addition, the dendrograms in the second row of the figure show how different inversions of a major chord (and for every other classes in equal tuning) are characterisable in terms of consonance.

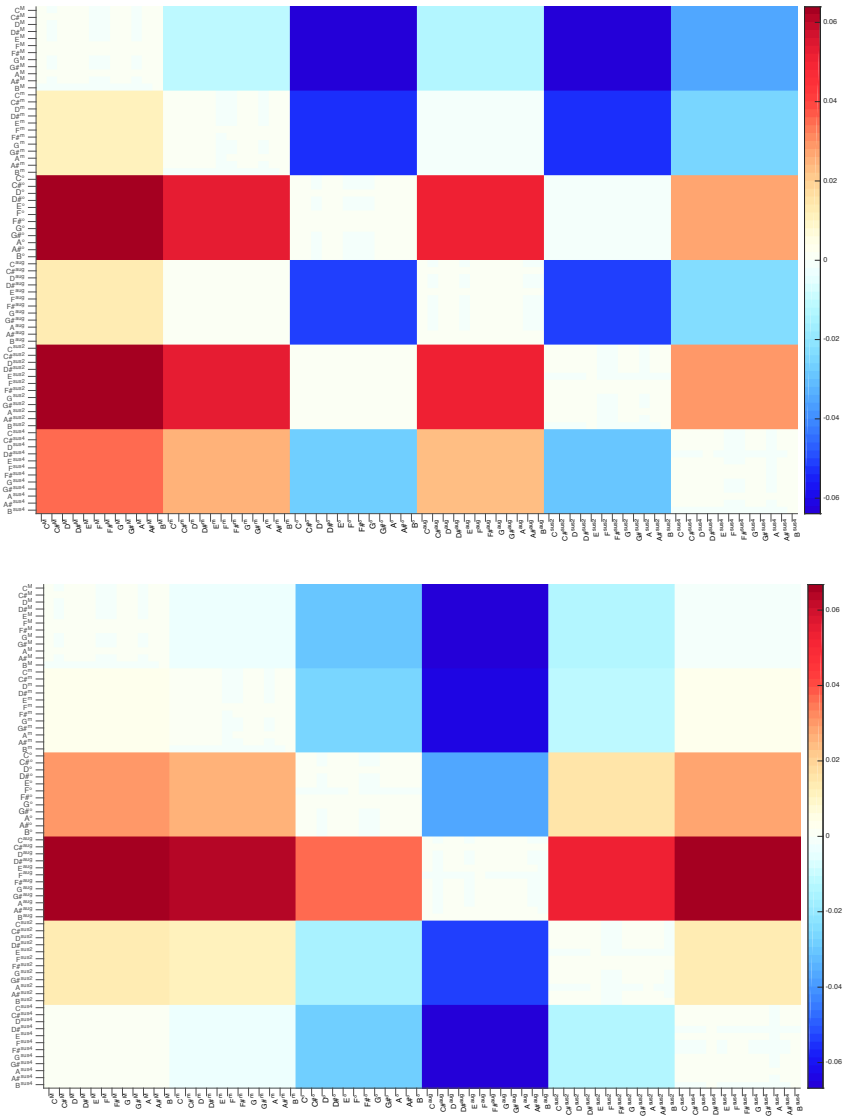


Figure 9.9: Distance matrices between triads in equal temperament. The value of each cell (i, j) is equal to the difference in calculated consonance between the chord j and the chord i. The matrices have been computed using h_1 and h_2 as harmonic spectra, respectively.

9.3.2 Analysis of block voicings on the consonance-deformed *Tonnetze*

The *Tonnetz* labelled with the pitches of the chromatic scale built on the 5th octave of the piano and deformed by the six classes of triads we considered is depicted in Figure 9.11. We start our analyses by describing the the dissonance values associated to each vertex of the six configurations of the *Tonnetz* induced by the triads. We shall see how the generalisation of the consonance function reflects our perception. Note that, at this stage, the deformed *Tonnetze* provide only a comfortable visualisation of chords.

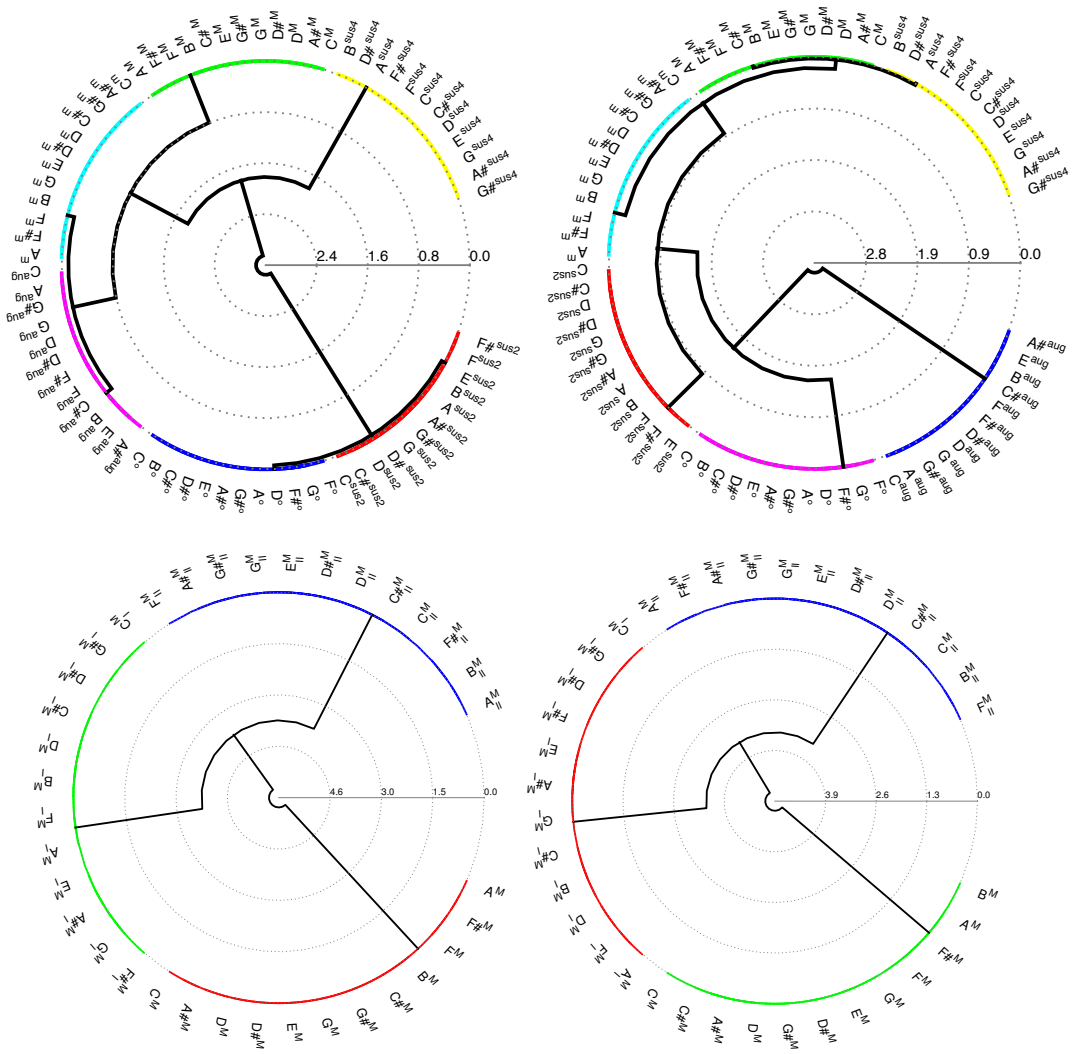


Figure 9.10: Hierarchical structure of triads' consonance. In the first row it is possible to observe how the consonance classify triads according to their classes, by using two different harmonic spectra. In the second row the inversions of the major triads are classified according to their consonance value, computed with h_1 and h_2, respectively.

Major triad deformation, $\mathfrak{T}_{maj}$. The major triad (C^4, E^4, G^4) is depicted in Figure 9.11a. The height of each vertex of the deformed configuration corresponds to a block voicing whose highest voice is the label of the vertex. Thus, it is not surprising to observe that the chord $(C^4, E^4, G^4, C\sharp^5)$ is the most dissonant point on the polyhedral surface, followed by the vertices labelled by $G\sharp$ and $E\flat$.

We also retrieve that the two smallest values of the consonance function correspond to the vertices labelled with G^5 and C^5, in particular we have

$$d(C^4, E^4, G^4, G^5) < d(C^4, E^4, G^4, C^5),$$

this behaviour corresponds to the fast decreasing that characterises the consonance

model we considered and the interaction between the two sequences of overtones generated by the pitches of each chord. Ordering the vertices by increasing heights, the next block voicing to be considered is (C^4, E^4, G^4, A^5), or a C^{add13} in standard jazz notation. This kind of harmonic solution is largely used in modern and classical music[4]. The same holds for the configuration corresponding to C^{add9}.

From a modal point of view, the set of notes belonging to the ionian scale $\{C, D, E, F, G, A, B\}$ represents the *least dissonant* seven notes set in this particular configuration, while the substitution of the perfect fourth with the augmented one of the lydian scale, $(F\sharp)$, introduce a local maximum in terms of musical tension.

The same argument can be applied considering the tension climax provided by the mixolydian scale $\{C, D, E, F, G, A, B\flat\}$; the mixolydian $\flat 6$ $\{C, D, E, F, G, A\flat, B\flat\}$; and the phrygian dominant scale $\{C, D\flat, E, F, G, A\flat, B\flat\}$.

Minor triad deformation: $\mathcal{T}_{min}$. As expected, the *Tonnetz* deformed with a minor triad C^m the vertex corresponding to a minor third is a minimum. Comparing this geometrical state to the one associated to the major triad, we can observe that the major second D will result in a less consonant choice, being the leading-tone of the third $E\flat$. The minor seventh $(B\flat)$, has a low dissonance configuration compared to the major seventh (B). Furthermore, the vertex associated to the chord (C^4, Eb^4, G^4, F^5) is a minimum of $\mathcal{T}_{min}$.

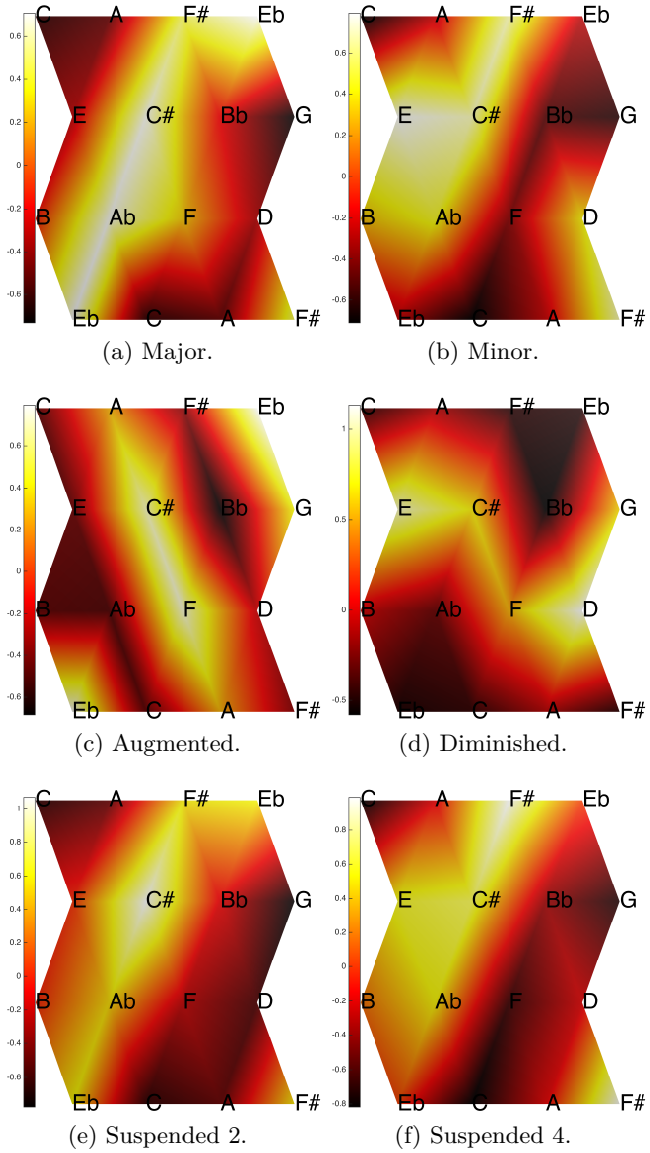
From the modal point of view, we retrieve the results obtained by persistent homology in the previous application, for instance the vertices labelled with the pitches corresponding to the eolian scale $\{C, D, E\flat, F, G, A\flat, B\flat\}$, and the dorian scale $\{C, D, E\flat, F, G, A, B\flat\}$ share similar configurations, as well as the ones corresponding both to the hypoionian $\{C, D, E\flat, F, G, A, B\}$ and the hypoionian $\flat 6$ scales.

Augmented triad deformation: $\mathcal{T}_{aug}$. The state generated by the augmented triad is characterised by a low dissonance configuration for the augmented fifth interval, which is part of the underlying chord. The major seventh has a low dissonance configuration and that should not surprise the reader, for the same frequency-spacing argument used before and for the standard use of $\Delta^{\sharp 5}$ chords, which are naturally generated, for instance, on the third degree of the seventh-chord harmonisation of the harmonic minor scale. The augmented fourth loses its role of leading-tone, since G is not part of the chord, thus the $F\sharp$ dissonance configuration is lower than the F.

Diminished triad deformation: $\mathcal{T}_{\circ}$. Among the cases we analysed, the diminished state is the only one where the vertex associated to the minor second $(C\sharp)$ is not the absolute maximum. On the modal side, this configuration appears to be reasonable, when considering the five standard possible modes associated to a diminished triad, as detailed in Table A.4.

On the tonal harmonic point of view, the modal argument we just introduced can be translated in terms of chord or non-chord tones. Generally diminished chords (occasionally equipped either with a minor or a diminished seventh) bear tension

[4]It suffices to think about *A Foggy Day* by Gershwin, or to the chorus of *Man In The Mirror* by M. Jackson.



(a) Major. (b) Minor.

(c) Augmented. (d) Diminished.

(e) Suspended 2. (f) Suspended 4.

Figure 9.11: *Tonnetze* deformed with the dissonance computed from the interaction of a triad in root position and the chromatic scale an octave higher than the chord.

given either by the perfect fourth, represented as a local minimum in Figure 9.11d, or by the minor sixth[5].

Suspended triads deformation: $\mathcal{T}_{sus2}$ and $\mathcal{T}_{sus4}$.

In a tonal composition suspended triads allow to circumvent or delay a precise tonal choice, since the lack of the third makes their univocal association to a key

[5]Examples of the use of diminished chord in modern music can be found in *Georgia on My Mind* by H. Carmichael, or *Make a Mistake with Me* by Brad Paisley

impossible[6]. This feature is well mirrored in the two simplicial complexes associated to the sus2 and sus4 triads, depicted in Figures 9.11e and 9.11f, respectively. For instance, observe how in Figure 9.11e, the consonant vertices correspond to the set of pitches $\{G, A, B\flat, B, D, C, F\}$, which is the union of a G^7 and a G^{m7} arpeggio with the addition of the perfect fourth C, that loses the central role it has in the other configurations of the *Tonnetz*.

What about the geometry? The structure of the *Tonnetz* is based on the harmonic relations among the pitches that label its vertices. The displacements of the vertices induced by the consonance of the triads generates maximal and minimal subcomplexes on the *Tonnetz* that characterise the triads and allow to recognise them even removing the labels from the vertices of the surface. The heat-map used to highlight the consonant and dissonant regions of the six deformed *Tonnetze* represented in Figure 9.11 makes them distinguishable at first sight. The geometries generated by the major triads are characterised by tensions lying on the perfect fifth axis. The configuration induced by a minor triad is characterised by a relevant minor triad of tensions, $(C\sharp, E, G\sharp)$ in the example. The same holds for the two classes of suspended 2 and suspended 4 triads, that can be assimilated to the geometries corresponding to a major and a minor triad, respectively. Moreover, the deformed *Tonnetze* associated to the augmented and diminished triads present completely different configurations in terms of block voicings' consonance.

If in our first analysis of the deformed *Tonnetz* the study of the sub-level sets of the height function was a natural consequence of our construction, here we can explore different geometric properties of these simplicial complexes, that will be used at the end of this chapter to induce several filtrations on these shapes, in order to classify them through persistent homology.

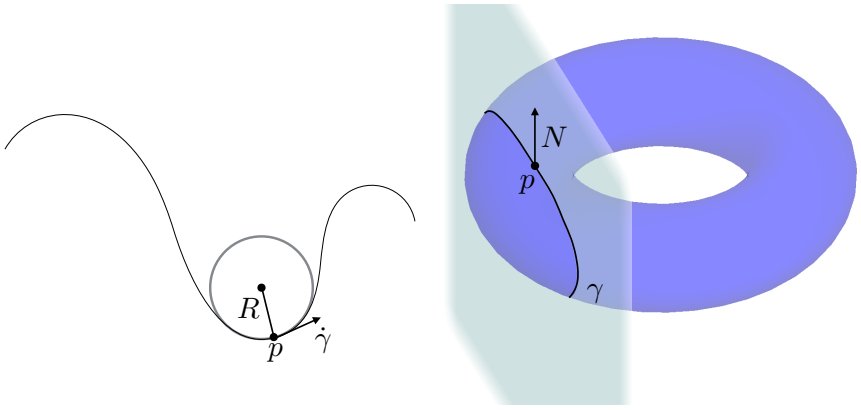
9.3.3 Gaussian curvature: a geometric music feature

In this section we use the discrete Gaussian curvature to analyse the different geometric states of the *Tonnetz*. In the next paragraph we provide an intuition concerning this geometrical property. Thereafter, we will give its musical interpretation, in the case of consonance-based deformed *Tonnetz*

Intuition

Consider a planar, unit speed curve $\gamma : [0, 1] \subset \mathbb{R} \to \mathbb{R}^2$. Its curvature is defined as the length of its acceleration $\kappa(t) = \ddot{\gamma}(t)$. Geometrically, the curvature at the point $p = \gamma(t)$ corresponds to the circle tangent to γ at p, having acceleration vector at p equal to the one of γ. This circle is called *osculating circle*, see Figure 9.12a for its representation. Thus, the curvature is $\kappa(t) = 1/R$, where R is the radius of the osculating circle. By this definition, the curvature is positive for every $t \in [0, 1]$. By choosing a normal vector field N along γ, the curvature can take both positive and negative values. The resulting function κ_N is called a *signed curvature*. The curvature of a surface S is described by a couple of numbers at each point p. Let

[6]Normally, it is possible to associate a precise tonality to a suspended triad, by analysing the harmonic context in which it is used.



(a) The osculating circle. (b) Computing the principal curvature.

Figure 9.12: Visualisation of the curvature for planar curves and surfaces.

N be the normal vector to the surface at p. Let π be a plane containing N. Its intersection with the surface generates a curve γ, as it is depicted in Figure 9.12b. We call the *principal curvatures* of S at p the numbers k_1 and k_2 realising the minimum and the maximum of the signed curvature κ_N, when considering all the normal plane π. The Gaussian curvature of a point $p \in S$ is defined as $K = \kappa_1 \cdot \kappa_2$. This definition allows one to classify the points of a surface as follows.

1. $K > 0$: elliptic points, $\kappa_1 \cdot \kappa_2 > 0$, the quadratic approximation of the surface in a neighbourhood of p is an elliptic paraboloid;

2. $K = 0$: parabolic points, one of the principal curvature is equal to zero, the quadratic approximation of the surface near p is a parabolic cylinder;

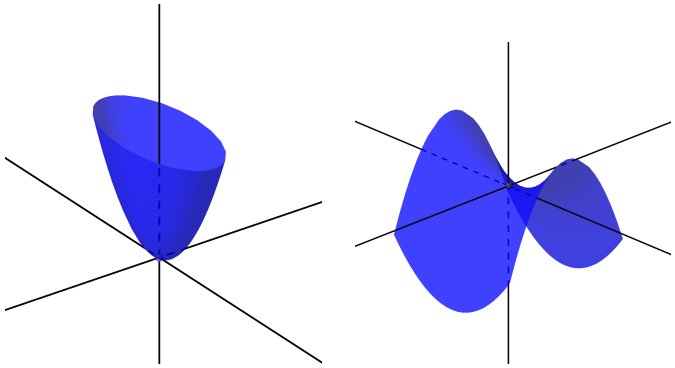
3. $K < 0$: hyperbolic points $\kappa_1 \cdot \kappa_2 < 0$, the quadratic approximation of the surface in p is given by a hyperbolic paraboloid.

4. Umbilical points: $\kappa_1 = \kappa_2 \neq 0$ elliptic point, or $\kappa_1 = \kappa_2 = 0$, planar points: it is not possible to determine the shape of the surface near p examining the second order derivate. The typical example is the point at the origin of a monkey saddle.

Intuitively, the discrete Gaussian curvature measures the bending of a polyhedral surface at each vertex. Let $v \subset K$ be a vertex, the discrete Gaussian curvature (or angular defect) at v is defined as

$$K_v = 2\pi - \sum_{i=1}^{n} \theta_i,$$

where θ_i are the interior angles at v of the triangles included in St_v. Thus, for instance, a positive discrete Gaussian curvature is associated to vertices giving rise to maxima and minima, a negative curvature to saddle points and trivial curvature to planar points. The algorithm we used to compute the discrete Gaussian curvature of the consonance-deformed *Tonnetze* is described in (Cohen-Steiner and Morvan, 2003).



(a) (b)

Figure 9.13: The elliptic paraboloid (a) and the hyperbolic paraboloid (b).

Labels	C^M	C^m	C^{aug}	C^{dim}	C^{sus2}	C^{sus4}
C^5	+	−	−	+	+	+
$C\sharp^5$	+	+	+	+	+	−
D^5	−	+	−	+	−	−
$E\flat^5$	+	+	+	−	+	−
E^5	−	+	−	+	−	+
F^5	−	−	+	−	−	+
$F\sharp^5$	+	+	−	+	−	+
G^5	+	+	+	+	+	+
$A\flat^5$	−	+	−	+	−	+
A^5	+	+	−	−	+	+
$B\flat^5$	+	+	+	+	+	−
B^5	+	−	−	+	−	−

Table 9.2: The sign of the discrete Gaussian curvature characterise the each vertex of the by considering its interaction with its star. Here it is possible to compare the curvature values associated to each pitch, in the six classes of triads that we analysed.

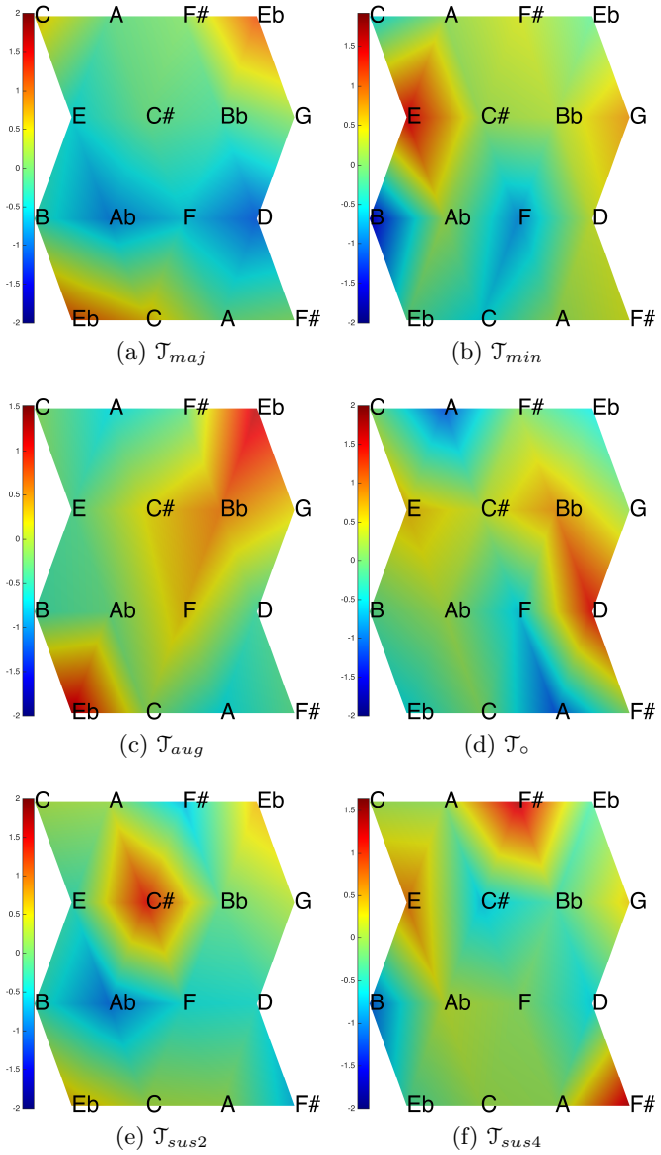
9.3.4 Musical interpretation

Consider Figure 9.14a, the curvature is positive in at the vertices labelled by the pitches

$$C^5, G^5, A^5, B\flat^5, B^5, E\flat^5, F\sharp^5, C\sharp^5.$$

These vertices correspond either to particularly high maxima or particularly low minima, hence, they are endowed with a strong characterisation in terms of consonance/dissonance. On the contrary, the vertices labelled with $D^5, E^5, F^5, A^5\flat$ are not maxima, nor minima with respect to the directions defined by the triangles included in their star. Symmetrical arguments hold for the other configurations of the *Tonnetz* depicted in Figure 9.14.

$\mathfrak{T}_{min}$ and $\mathfrak{T}_{aug}$ are the only deformations in which the vertex labelled with C^5 has negative curvature (see Table 9.2). The vertex corresponding to G^5 has positive curvature in every configuration. It generates highly consonant block voicings, when



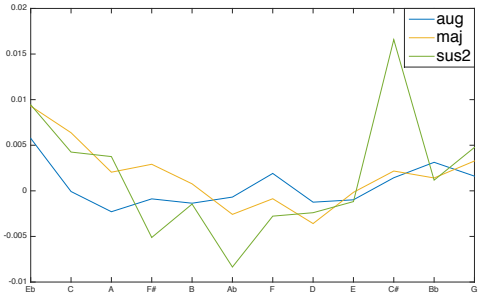
(a) $\mathfrak{T}_{maj}$ (b) $\mathfrak{T}_{min}$

(c) $\mathfrak{T}_{aug}$ (d) $\mathfrak{T}_{\circ}$

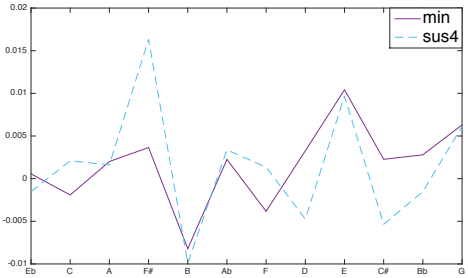
(e) $\mathfrak{T}_{sus2}$ (f) $\mathfrak{T}_{sus4}$

Figure 9.14: Discrete Gaussian curvature on the deformed *Tonnetz* in different state. The colormap shows the smoothed value of the curvature in every point of the surface. We recall that the labelling of the *Tonnetze* is given by pitches of the chromatic whose root is one octave apart from the root of the triad used to produce the deformation of the *Tonnetz*.

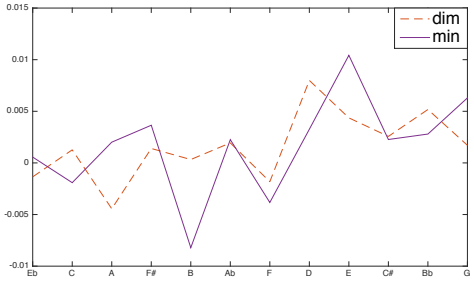
superposed to the major, minor and the suspended triads. On the contrary, it is extremely dissonant when coupled with either the augmented or the diminished triad. Both the vertices associated to $C\sharp^5$ and $B\flat^5$ have negative curvature only in the $\mathfrak{T}_{sus4}$. The suspended4 triads is meant to be perceived as a centre of gravity between two consecutive tonalities on the circle of fifths, in this case C and F major. Hence, $B\flat^5$ lose its strength both in terms of tension (in C major) or resolution to F. Furthermore, $C\sharp^5$, labelling the most dissonant chord in the other configurations,



(a) M vs aug vs $sus2$.



(b) m vs $sus4$.



(c) m vs dim

Figure 9.15: Values of the discrete Gaussian curvature on the vertices of deformed fundamental domain of the *Tonnetz*.

here is outclassed by E and $F\sharp$ (see Figure 9.11).

The discrete Gaussian curvature allow to classify the four notes chords whose consonance determine the height of the vertices of the *Tonnetz* for a fixed triad. We saw how the value of the curvature associated to certain pitches recurs in configurations induced by different triads. A comparison of the columns of Table 9.2 reveals that the sign of the curvature is almost identical for the configurations induced by major, augmented and suspended2 triads. The values of the discrete curvature associated to the vertices of different deformed *Tonnetze* is represented in Figure 9.15. Although columns of Table 9.2 associated to the suspended4 and the minor triads are different, it is interesting to notice how five of their vertices share almost the same curvature value. Moreover, notice how the union of the set of pitches of the major and suspended4 triads gives four or five note of the major pentatonic scale built on the root of the triads, respectively. The same holds for the minor and the suspended4 triads, with respect to the minor pentatonic scale.

9.3.5 Classification of the consonance-deformed *Tonnetze*

To conclude the analysis of these consonance-deformed shapes, we compute their persistent homology by considering the three different filtrations, induced by the Rips complex (proximity of the vertices as a point cloud), the height function (block voicing consonance) and the more exotic discrete Gaussian curvature.

The dendrograms computed by considering the distance between the 0-persistence diagrams of the shapes for each filtration are depicted in the left column of Figure 9.16.

The three filtrations highlight different musical properties of the chords:

(a) *Rips complex.* The six classes of triads are grouped into two almost specular clusters. The first by the two suspended triads and the major one. The second composed by the diminished, augmented and the minor triad. The characterisation of the shape given by the Rips complex classify the suspended triads near to their resolution, while the minor triad is an outlier respect to the two most dissonant triads that have distance 0 in this representation.

(b) *Consonance.* The filtration induced by the sub-level sets of the height function order the simplices of T respect to the consonance of the voicings built on their vertices. As expected this filtration classifies the triads according to their own consonance value.

(c) *Curvature.* The grouping of the triads obtained by studying the sub-level sets of the discrete Gaussian curvature reflects the our observations. We retrieve the pentatonic link between the major-suspended2 and minor-suspended4 triads, respectively. As well as the segregation of the diminished mode and the similarity of the triplet suspended2, augmented and minor triad.

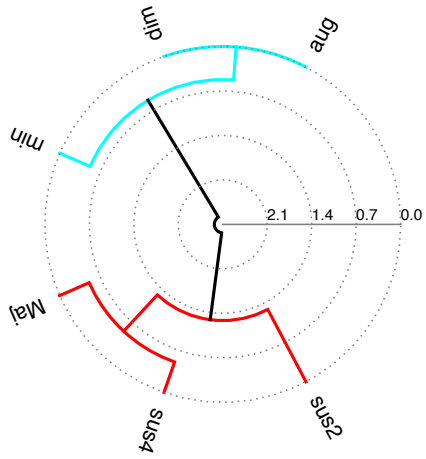
Once the musical interpretation of the geometric properties of the shapes we generated is clear, persistent homology allows us to explore different parameters of the consonance function. The left column of Figure 9.16 has been obtained by changing the harmonic spectrum used in the consonance function to $h = (1, 1/2, 1/3, 1/4, 1/5, 1/6)$. The classification of the shapes provided by the Rips complex is almost unchanged; the filtration induced by the height function points out the similarity between the shapes generated by the major and minor triads, while the others are segregated. The consonance function, in this second and more realistic configuration groups the shapes associated to a suspended2 and major triads, while another cluster contains the augmented and diminished triads. The minor and suspended4 triads are outliers.

9.4 Discussion

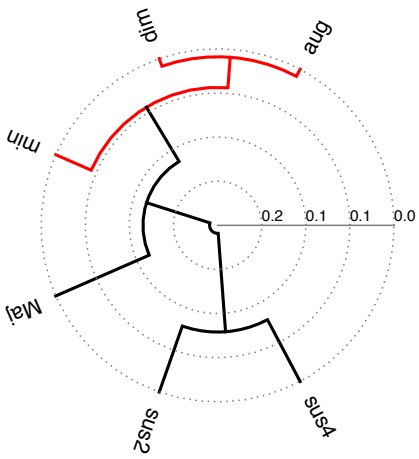
In Sections 9.2 and 9.3 we suggested two applications of consonance calculation for the hierarchical organisation of musical entities. The filtration induced by the Rips complex, built on a point cloud representing each mode, allowed to compute a dissimilarity among modal scales. The octave dependency of the consonance function has been discussed. It has been shown how the mixolydian modes maintain the same relationship among them, when considering a fixed reference pitch and scales built on two consecutive octaves. In general, the filtration defined by the Rips complex segregates recognisable sonorities, i. e. modal scales that together with a bass note used as accompaniment give rise to what we could refer to as *Spanish sonority*, or a *bluesy* one, and so forth.

In Section 9.3, we have studied the organisation of common triads. A simple generalisation of the Plomp and Levelt consonance model allows to associate a unique consonance value to each triad class. It has been shown how the consonance is invariant modulo transposition of the same triad and how different classes of triads are recognised, despite changes of the harmonic spectrum used to compute the their

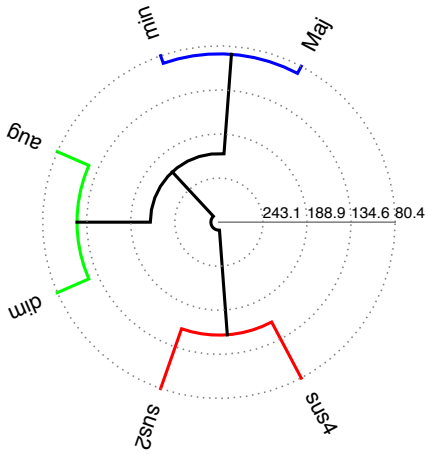
consonance. Later on, this was used to study the relations between different pitch classes in a given chordal context, by the usual update of the height of the vertices of the *Tonnetz*. This led to a variable geometry space which is suitable for the analysis of different chord classes, and provides a link between the harmonic and the melodic level intended as the block voicings of a triad on the chromatic scale. The surfaces obtained in this way have been discussed, considering the height function, as well as their discrete Gaussian curvature. In conclusion, three different filtrations have been used to classify these six classes of triads, according to the geometric properties of their representations as deformed *Tonnetze*. Once the meaning of the function used to induce the filtration is clear, the classification provided by persistence homology allows to study the effect of the harmonic spectrum on the geometry of the deformed *Tonnetze* and hence with respect to the tension/resolution patterns of each pairing of a triad class and chromatic scale.



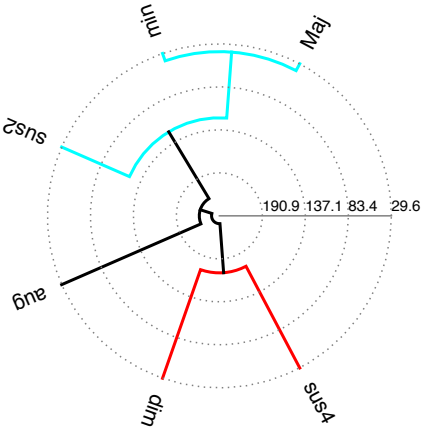
(a) Rips complex



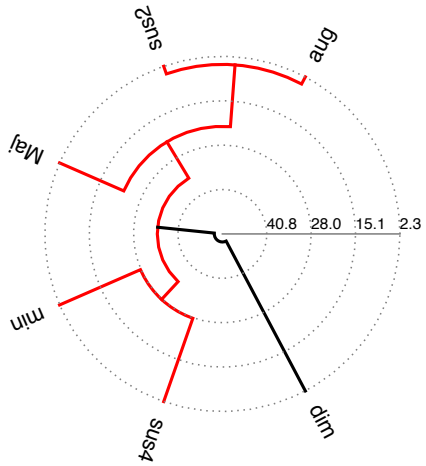
(b) Rips complex



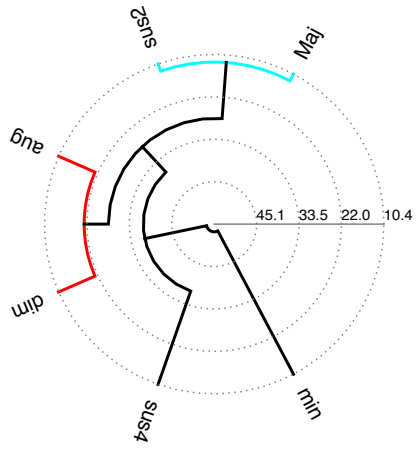
(c) Height function



(d) Height function



(e) Discrete Gaussian curvature



(f) Discrete Gaussian curvature

Figure 9.16: Hierarchical clustering of consonance-deformed *Tonnetze* generated by triads and two harmonic spectra: $(1, 1, 1, 1, 1, 1)$ on the left column and $(1, 1/2, 1/3, 1/4, 1/5, 1/6)$ on the right.

Ten

Discussion and future works

The three applications described in this part represent a first formalisation of the topological and geometrical music analysis. Music features have been represented as polyhedral surfaces and point clouds. After the analysis and the discussion of several dataset, persistent homology revealed itself an efficient tool both in a purely symbolic or in a hybrid signal/symbolic context.

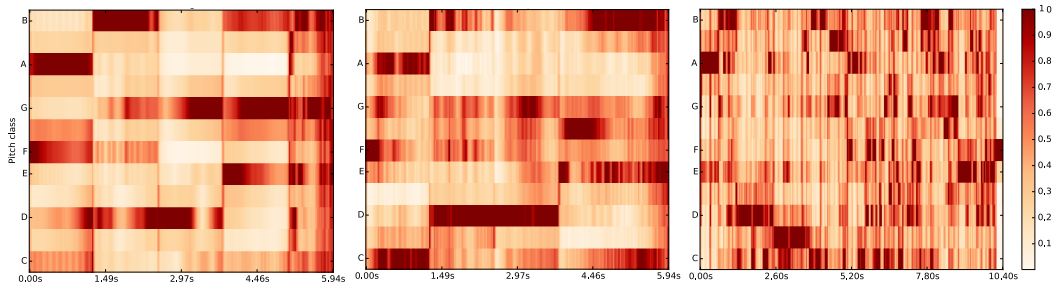


Figure 10.1: Chromagrams.

The first model for the analysis and classification of music based on pitch classes and notes' durations we suggested has been realised considering different datasets of MIDI files. The stability of the height function allows to extend this analysis to audio files. It is possible to describe the pitch classes and duration of the notes in an audio file, by computing a chromagram (Harte and Sandler, 2005). Such a representation allows to pass from the domain of time to the domain of frequencies through a fast Fourier transform. Then pitch classes are obtained wrapping frequencies on a single octave. Each pitch class is represented taking into account its magnitude. The first two chromagrams[1] in Figure 10.1 describe the pitch-class contribution during a perfect cadence $Dm7 - G7 - Cmaj9 - Cmaj7$ played using two virtual instruments emulating a Fender Rhodes and a Bösendorfer respectively. The third one represents a small fragment (≈ 10 seconds) of a jazz composition involving two guitars, a basic drum set and a bass. From such representation it is possible to deduce what notes are played considering their magnitude (colour) and how much time they last (horizontal axis). In this case, the height function would surely be affected by the noisy data coming form the signal. However, the stability of the persistence diagrams for tame functions assures that a small perturbation of the

[1]The figures have been realized through the Librosa Python library available at `https://bmcfee.github.io/librosa/`.

function that induces the filtration corresponds to small variations of the persistence diagrams.

In order to generalise the consonance based applications to the analysis of complex signals, it is necessary to retrieve more information than the one represented in a chromagram. One possibility is represented by the *Snail Analyzer-Tuner* (`http://medias.ircam.fr/x1b825e` at minute 20). A software developed at the IRCAM, that allows to visualise the same information represented in a chromagram, on the whole audible frequency spectrum. This information coupled with a chord detection algorithm(Mauch et al., 2009; Ellis and Weller, 2010), would allow to compute the consonance values and hence to update the height of each vertex of the *Tonnetz* in time, directly from an audio signal.

The model itself can be largely refined in several ways. It is possible to augment its dimensionality, losing its property to be easily visualisable, but having the possibility to encode more information. For instance, it could be possible to associate to each pitch class of the *Tonnetz* its velocity, or merge the two pitch-class/duration and consonance approaches. Moreover, topological persistence offers further tools to improve the strategies we suggested. A natural development is the study of the multidimensional persistent homology (Cagliari et al., 2010; Cerri et al., 2013; Carlsson and Zomorodian, 2009) of musical spaces.

Part IV

Harmonic sequences and persistence time series

Table of Contents

Abstract

This part is devoted to the analysis of the time-dependent nature of music. In Chapter 11 we suggest a novel approach to the analysis of popular music, based on the multiple (read simultaneous) alignment of symbolic sequences. We will consider the sequences derived from several harmonic-oriented analyses of the harmonic progressions of a dataset of 138 compositions. Given a harmonic progression, music theory provides the tools allowing to retrieve tonal centres, cadences and modulations. We take advantage of these high-level features to define three families of symbolic sequences associated to our dataset. Their global pairwise alignment is used to tackle several problems such as detection of cover tracks and the retrieval of genres and artists. Finally, multiple sequence alignment is computed to produce an encompassing analysis of the transfer of musical patterns among the heterogeneous collections of songs, artists and genres of the dataset we analysed. This chapter represents joint work with Philippe Esling (Esling and Bergomi, 2015).

The main contribution of Chapter 12 is the adaptation of the model proposed in Chapter 8 (whose aim was the analysis of *persistent properties* of music) to the study of the time-varying geometry of the *Tonnetz*, when its vertices are deformed considering the pitch classes and durations of the notes of a composition. In particular, we consider the time series arising from the topological analysis of the natural time framing of music, provided by its subdivision in bars. The state of the art concerning the representation of time-varying systems in the formalism of topological persistence is discussed and a method to align these time series composed by *topological subfingerprints* is provided. We investigate the musical relevance of the information carried by the time series of persistence diagrams, as well as the analysis of their dissimilarity. In particular, we will focus on three datasets collecting classical, pop and jazz compositions, respectively.

Eleven

Harmonic time series and pop music analysis

The past decade has witnessed a growing interest in *content-based retrieval* for multimedia databases (Yoshitaka and Ichikawa, 1999). Large amounts of work have been devoted to performing similarity queries and genre recognition over musical songs databases, leading to the field of *Music Information Retrieval* (MIR) (Casey et al., 2008). However, in this field, the signal-based and symbolic-based approaches to musical analysis are often considered antipodal strategies. Could these viewpoints cohexist as complementary? Would it be possible to improve the results provided by signal analysis by augmenting its abstraction level through a symbolic framework? The limitations of the *genre recognition* tasks have recently been exhibited. Indeed, since the first work in musical genre recognition in 1995 (Matityaho and Furst, 1995), it seems that most research still revolves around a signal-based classification of a ground truth annotation of music genre provided by human experts and a train/test paradigm (Sturm, 2014). However, the reference databases for the evaluation of these systems suffer from multiple flaws such as duplicatas, corrupted files, genres made of single artists and wrong (or too subjective) genre labelling (Sturm, 2013a,b). Furthermore, this simplifying task is far from accounting the fact that musical *inspiration* and transfer of different *musical patterns* go way beyond the notion of musical genre.

We introduce an innovative way to analyse pieces of popular music (termed here *pop* music), by performing a high-level symbolic analysis of their content. Our main goal is to provide an encompassing view over the cadential patterns and modulations motifs in pop music (Everett, 2000; von Appen et al., 2015) and how these can show artistic influences across various musical genres. In order to achieve this aim, the harmonic progressions corresponding to a dataset of pop songs is derived from their corresponding audio signals and a harmonic analysis is performed. As a result, we retrieve three different classes of symbolic sequences describing high-level tonal features of each song. We further analyse the similarity among pairs of sequences belonging to the same symbol class by relying on several state-of-the-art global sequence alignment algorithms typically used in time series (Esling and Agon, 2012) and genetic analyses. Some interesting properties of this approach are discussed. First, it is shown how the similarities of these sequences can provide a valuable instrument to refine hand-made semiotic segmentations of the songs. Second, the accuracy of the harmonic transcription and the harmonic analysis is evaluated by performing a cover track recognition task. In addition, the clusterings generated by

each possible combination of weight matrix and alignment algorithm are evaluated by measuring their cluster-wise accuracies with respect to the metadata corresponding to the considered collection of songs. Although our analyses provide higher-level assessments of melodic and harmonic similarities across musical pieces, we show that it still bears some coherence in the traditional evaluation paradigm of genre and artist recognition.

We further introduce the application of Multiple Sequence Alignment (MSA) (Thompson et al., 2011) in the analysis of music at a symbolic level. This analysis is performed by considering several state-of-art MSA algorithms, which are evaluated through a variety of quality metrics. As a result of this multiple alignment procedure, it is possible to compute both the consensus sequences associated to each cluster and perform an analysis of motifs over the whole dataset. The former represents a paradigmatic sequence of modulations, which exhibits the typical path followed in the songs belonging to a certain cluster, while the latter provides an evident representation of the harmonic contamination and artistic influences among pop songs. Hence, we perform an encompassing analysis across pop music, based on the MSA structure organised across different clusters. By analogy to the well-known *molecular clock* hypothesis in genetics (Martin and Palumbi, 1993) which allows to evaluate the similarity between species based on their shared amount of genetic similarity, we show that these types of analyses provide higher-level views on musical similarity across genres, with various artists influencing each other over time.

11.1 Symbolic sequence alignment

11.1.1 Pairwise sequence alignment

The goal of pairwise alignment is that given a pair of symbolic sequences $\mathcal{S}_1$ and $\mathcal{S}_2$ of potentially different length $n, m \in \mathbb{R}^2$ composed of symbols from a given alphabet Σ and a scoring function $\delta(x, y)$ that defines the similarity between two symbols $x, y \in \Sigma^2$ we want to find the sequences $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$ of equal-length k such that the sum of similarity scores is maximized by inserting gaps in the two original series.

Definition 11.1.1. Given two sequences $S_1 \in \Sigma^n$ and $S_2 \in \Sigma^m$, pairwise alignment seeks the sequences $S_1' \in \Sigma^k$ and $S_2' \in \Sigma^k$ $(k \geqslant max(m, n))$ such that $\left\|S_1' - S_2'\right\|_\delta$ is minimal.

Based on a set of symbolic sequences, we can evaluate their pairwise similarities by aligning two sequences at a time (Sankoff, 1972). This alignment can be based on any type of symbolic information, given that the two sequences are composed of symbols with the same underlying signification. Pairwise alignment allows to gain information about the similarity between two sequences, but also about their inner structure. Hence, this can allow to find *common patterns,* or to assemble together set of sequences (*fragment assembly*). The different issues related to pairwise alignment are that

- Most of the sequences we are comparing will differ in length
- There may be only relatively small matching regions in the sequences

```
GSAQVKGHGKKVADALTNAVAHV---D--DMPNALSALSDLHAHKL
::  ::::|: ||    : :|   ::    |   ||||:|: |:::|: |
NNPELQAHAGKVFKLVYEAAIQLQVTDVVDMPNTLKNLGSVHVSKG
```

Figure 11.1: Example of global alignment between two (apparently) lowly-related sequence. Exact matches are identified by (|) and related matchs are identified by (:). Even though the symbols in both sequences are quite different, most of these are actually closely related in their functions, which implies that the sequences share a high amount of similarity.

- We want to allow variable matches between the symbols

It should be noted that three types of alignments can be performed. A *global alignment* seeks the best match between both sequences in their entirety and it is the only type of alignment utilized in this work. A *local alignment* will find the best *subsequence* match, even in very small portions of the sequences. Finally a *semi-global alignment* seeks the best global match without penalizing gaps on the ends of the alignment. An example of *global pairwise* alignment is displayed in Figure 11.1.

Levenshtein (edit) distance

The first way to obtain the alignment between two symbolic sequences is through the Levenshtein distance (also called *edit distance*) (Levenshtein, 1966), which considers that three types of differences can arise wen comparing two symbolic sequences

$$\begin{array}{ccccc} \textit{Substitutions} & ACGA & \Rightarrow & A\mathbf{G}GA \\ \textit{Insertions} & ACGA & \Rightarrow & AC\mathbf{C}GA \\ \textit{Deletions} & AC\mathbf{G}A & \Rightarrow & ACA \end{array}$$

The Levenshtein distance is defined as the minimal number of applications of these operations that are required to transform one sequence into another. The main problems of this distance are that all operations are considered equivalent (the same score is assigned to any change) and only the binary match/mismatch relationship is taken into account (symbols cannot be more or less related).

Dynamic Programming

Dynamic Programming (DP) provides an optimal solution to the global alignment. Its basic assumption is that the optimal solution can be found by aggregating several optimal solutions computed considering smaller parts (subsequences) of the problem (Berndt and Clifford, 1994).

Scoring scheme This approach relies on a *substitution* (or *weight*) *matrix* $\delta(x, y)$ which indicates the score of aligning any characters x and y from our alphabet Σ. Moreover, the scoring use a *gap penalty function* $w(k)$ which indicates the cost of a gap of length k, usually through a *linear cost* $w(k) = g \cdot k$ where $g \in \mathbb{R}$ is a constant.

The idea of dynamic programming is that given a sequence S of length n and a sequence T of length m, we can construct a $(n+1) \times (m+1)$ matrix $\mathcal{F}$ such that $\mathcal{F}_{i,j}$ is the score of the best alignment of $S[1 \dots i]$ with $T[1 \dots j]$. This means that the score of any cell can be deduced by the scores of its three previous neighboring (up and left) cells. Therefore when extending an alignment in the cell $\mathcal{F}_{i,j}$, three choices can be made

- align $S[1 \dots (i-1)]$ with $T[1 \dots (j-1)]$ and match $S[i]$ with $T[j]$.
- align $S[1 \dots i]$ with $T[1 \dots (j-1)]$ and match a gap with $T[j]$.
- align $S[1 \dots (i-1)]$ with $T[1 \dots j]$ and match a gap with $S[i]$.

Hence one way to specify the DP problem is in terms of its recurrence relation

$$\mathcal{F}(i,j) = max \begin{cases} \mathcal{F}(i-1, j-1) + \delta(S[i], T[j]) \\ \mathcal{F}(i-1, j) + g \\ \mathcal{F}(i, j-1) + g \end{cases}$$

Several algorithms have been developed based on this idea, such as the Dynamic Time Warping (DTW) algorithm (Berndt and Clifford, 1994) which is the first to use these principles. However, it is extremely brittle to the presence of outliers and noisy regions. These problems can be alleviated by allowing gaps in matching two sequences, with algorithms such as the Longest Common Subsequence (LCSS) (Das et al., 1997). Finally, the Edit Distance with Real Penalty (ERP) (Chen and Ng, 2004) attempts to combine the merits of DTW and edit distance by using a *constant reference point.* We will use these three algorithms in our subsequent analyses.

Needleman-Wunsch algorithm

The linear gap penalty function ($w(k) = g \cdot k$) of the DP approach implies that a long gap of n positions has the same impact on the alignment as n gaps disseminated along both sequences. However, it seems obvious that we should favour a single long gap between highly matching sub-sequences (putting emphasis on the similarity of local structures shared between sequences). The Needleman-Wunsch (NW) algorithm (Needleman and Wunsch, 1970) was introduced to handle this mechanism by providing an *affine gap penalty* function

$$w(k) = \begin{cases} \alpha + \beta k & k \geqslant 1 \\ 0 & k = 0 \end{cases}$$

where α defines the cost for *opening* the gap and β defines the cost for *extending* it. By choosing $\alpha > \beta$, we implicitly penalize small sporadic gaps as it costs more to open a gap than to extend an existing one. In order to keep the computational complexity of this refined alignment in $O(n^2)$ time, the NW algorithm relies on three different scoring matrices instead of a single one. First, the matrix $\mathcal{M}(i,j)$ defines the best score given that $S[i]$ is aligned to $T[j]$. Second, $\mathcal{I}_S(i,j)$ defines the best score given that $S[i]$ is aligned to a gap and $\mathcal{I}_T(i,j)$ defines the best score given that $T[j]$ is aligned to a gap. Hence, the NW algorithm redefines the previous recurrence relations as

$$\mathcal{M}(i,j) = max \begin{cases} \mathcal{M}(i-1,j-1) + \delta(S_i, T_j) \\ \mathcal{I}_S(i-1,j-1) + \delta(S_i, T_j) \\ \mathcal{I}_S(i-1,j-1) + \delta(S_i, T_j) \end{cases}$$

$$\mathcal{I}_S(i,j) = max \begin{cases} \mathcal{M}(i-1,j) + \alpha + \beta \\ \mathcal{I}_S(i-1,j) + \beta \end{cases}$$

$$\mathcal{I}_T(i,j) = max \begin{cases} \mathcal{M}(i,j-1) + \alpha + \beta \\ \mathcal{I}_T(i,j-1) + \beta \end{cases}$$

The overall NW algorithm can be drafted in three steps

1. Initialization
   a) $\mathcal{M}(0,0) = 0$
   b) $\mathcal{I}_x(i,0) = \alpha + \beta.i$
   c) $\mathcal{I}_y(0,j) = \alpha + \beta.j$
2. Fill the three matrix ($\mathcal{M}$, $\mathcal{I}_x$ and $\mathcal{I}_y$) together iteratively
3. Traceback
   a) Start at the largest value between $\mathcal{M}(m,n)$, $\mathcal{I}_x(m,n)$ and $\mathcal{I}_y(m,n)$
   b) Stop at any of $\mathcal{M}(0,0)$, $\mathcal{I}_x(0,0)$ and $\mathcal{I}_y(0,0)$

On the influence of the scoring matrix

One of the core concepts shared by most variants of the DP and NW algorithms is that they rely on a scoring function $\delta(x,y)$ which provides a mechanism to define variable symbolic matching. Hence, one of the key factor in the success of alignment algorithms lies in this prior knowledge of the symbols (dis)similarities. This can be defined as the *dissimilarity measure* $\delta(x,y)$ between symbols x and y (usually summarised in a *weight matrix*)

Remark 16. The scoring function is not necessarily a metric. We recall that a function δ is called a metric if it is symmetric

$$\delta(x,y) = \delta(y,x)$$

and subadditive

$$\delta(x,z) \leqslant \delta(x,y) + \delta(y,z).$$

The definition of this scoring matrix highly influences the resulting alignment. Furthermore, we can evaluate the score of an alignment by using the sum of all distances

$$\mathcal{D}(\mathcal{S}_1, \mathcal{S}_2) = \sum_k \delta\left(s_1^k, s_2^k\right)$$

or by minimising the *entropy* of each column given by

$$\mathcal{D}(s_i) = -\sum_a c_{ia} log_2(p_{ia})$$

where s_i is the i^{th} column of an alignment s, c_{ia} is the number of occurences of character a in column i and p_{ia} is the probability of character a in column i. The effect of devising different scoring matrix is displayed in Figure 11.2.

Applications to music

Over the past years, there has been several researches relying on pairwise sequence alignment for musical data analysis. Most of these works are devoted to *content-based querying* such as *Query By Humming* (QBH), which allows to retrieve a song inside a database based on a query hummed by the user (Pardo and Sanghi, 2005). Other works have targeted the use of sequence alignment to improve optical music recognition from multiple recognizers (Bugge et al., 2011), score following in order to distinguish between aligned and non-aligned audio frames (İzmirli and Dannenberg, 2010), cover detection using local alignment algorithms (Martin et al., 2012) and folk music analysis (Bergomi and Andreatta, 2015; Bergomi et al., 2015).

11.1.2 Multiple sequence alignment

Pairwise alignments allows to define a similarity but also to find some common local structures between sequences. However, if an entire set of sequences is analysed, pairwise alignment fails to provide a more encompassing level of reasoning, as it is unable to *align multiple sequence at the same time.*

Definition 11.1.2. The problem of *Multiple Sequence Alignment* (MSA) can be defined as finding from a set of k sequences of various length $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$, the aligned set of k equal-length sequences $\mathcal{S}^* = \{\mathcal{S}_1^*, \mathcal{S}_2^*, \ldots, \mathcal{S}_k^*\}$, where $\mathcal{S}_i^*$ is obtained by inserting gaps into $\mathcal{S}_i$, $\forall i \in [0, k]$ while minimizing the overall dissimilarities between the symbols.

Compared to pairwise alignment, MSA requires a global objective to minimise across the whole set of sequences. The most straightforward way to define an error function to minimise in a MSA problem is to rely on the *Sum-of-Pair* (*SP*) score

$$SP_{score}(a_1, \ldots, a_k) = \sum_{1 \leqslant i < j \leqslant k} \delta(a_i, a_j)$$

where a_i is a column of the alignment composed of symbols from our dictionary (or gaps) and $\delta(a_i, a_j)$ is the distance defined in our weight matrix. Then, the overall score of an alignment $\mathcal{S}^*$ can be defined as

$$SP_{score}(\mathcal{S}^*) = \sum_x SP_{score}(S_1^*[x], \ldots, S_k^*[x])$$

In other words, we are trying to minimise the position-wise differences in symbols simultaneously for all sequences in the alignment.

As opposed to pairwise alignment, there has been, to the best of our knowledge, no application of this approach to musical data. The only work relying on MSA was aimed at lyrics alignment (Knees et al., 2005) where lyrics extracted from the internet were used to perform faster retrieval of songs. We will provide in this work the first assessment of MSA for harmonic and motivic analyses.

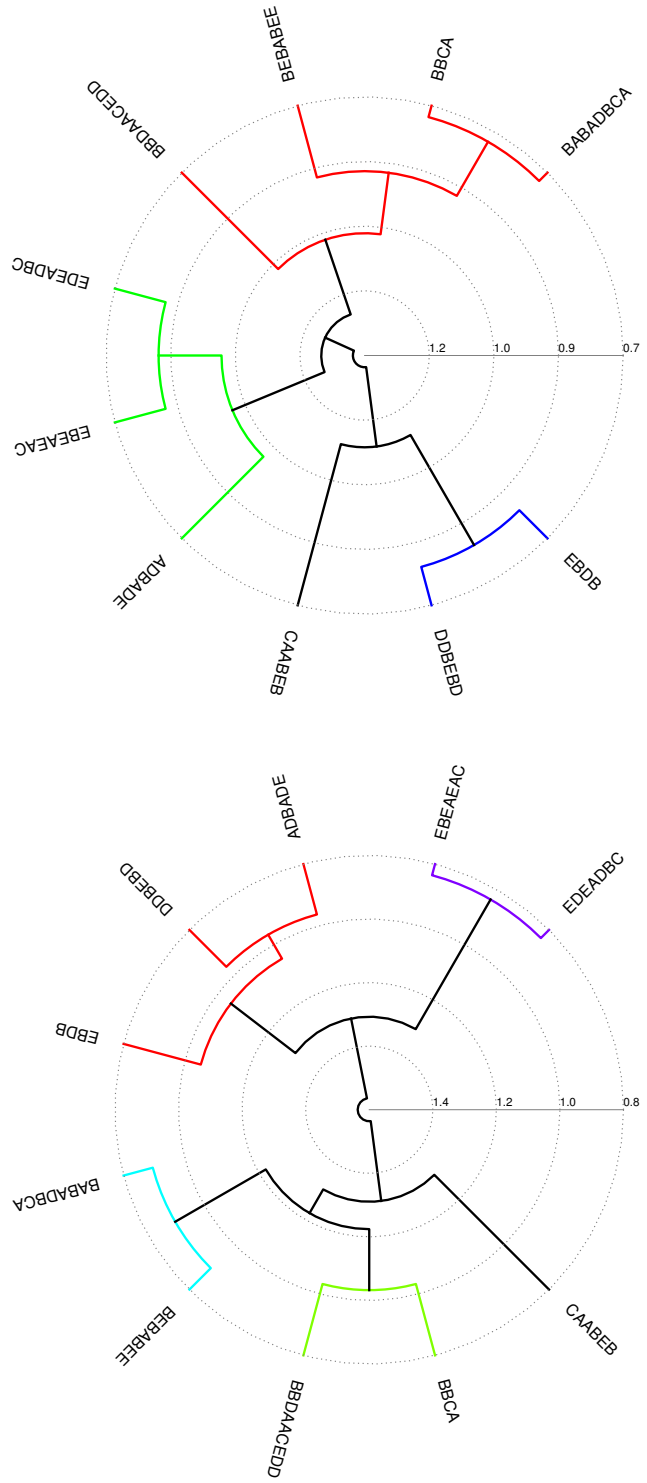


Figure 11.2: The effect of using different grammars (symbolic information) and different weighting matrix can lead to dramatically different results in the final alignments and similarities between the sets of sequences.

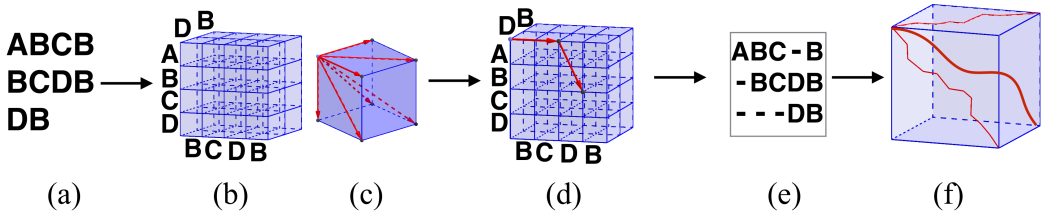


Figure 11.3: Multiple sequence alignment of 3 sequences through dynamic programming. (a) Given a set of 3 sequences to align, (b) we can construct a 3-dimensional matrix in which (c) each cell defines 7 different paths. (d) Following the same procedure as pairwise alignment, we can find the optimal (e) multiple sequence alignment. (f) An interesting property is that we can project the multidimensional path on bi-dimensional planes to obtain pairwise alignments between any sequence of the set.

Dynamic programming

As seen previously, dynamic programming is an excellent tool to perform the alignment of two sequences, as it provides the global optimum to this problem. This technique can be extended to perform the alignment of a set of k sequences and provides the optimal solution for this set. We can rewrite the original dynamic programming equation as

$$V(i_1, i_2) = \underset{(b_1,b_2)\in\{0,1\}^2-\{(0,0)\}}{max} \{V(i_1 - b_1, i_2 - b_2) + \delta(S_1[i_1 b_1], S_2[i_2 b_2])\}$$

This equation simply states that the best path from one cell depends on its 3-neighbourhood of previous cells in the scoring matrix. As this form is closely related to that of the *SP*-score, we can extend it by considering

$$V(i_1, \ldots, i_k) = SP_{score}\{align(S_1[1 \ldots i_1], \ldots, S_k[1, \ldots i_k])\}$$

Hence, the score of the last column would be the *SP*-score of the optimal alignment between the k sequences. Therefore, for each cell of the $k-$dimensional matrix

$$V(i_1, \cdots, i_k) =$$
$$= \underset{(b_1,\ldots,b_k)\in\{0,1\}^k}{max} \{V(i_1 - b_1, \ldots, i_k - b_k) + SP_{score}(S_1[i_1 b_1], \ldots, S_k[i_k b_k])\}$$

Therefore, the *SP-score* of the optimal multiple alignment of $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_k\}$ is $V(n_1, \cdots, n_k)$ where n_i is the length of $\mathcal{S}_i$ (ie. the "last" cell of the scoring matrix). Overall, we fill the $k-$dimensional scoring matrix similarly to the two sequences to compute $V(n_1, \cdots, n_k)$. This process is detailed in Figure 11.3.

However, it should be noted that the complexity is exponential in the number of sequences to align. Therefore, this algorithm rapidly becomes impossible to apply both in terms of computation time and memory requirement.

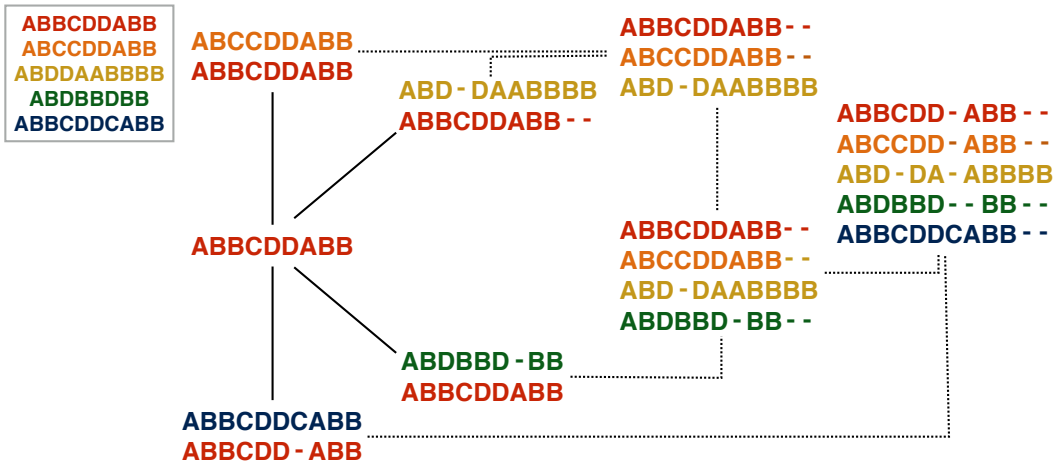


Figure 11.4: Summary of the centre star algorithm.

Center-star method (approximation)

It would be preferable to obtain a good approximation of the optimal alignment using polynomial time. The *centre star* method was one of the first method proposed to minimise the SP-score in an efficient way.

The main idea behind this method is to find a *reference (centre) sequence* inside the set of sequences to align, and then aligning all other sequences with this reference. In order to find the reference sequence, we compute the pairwise alignments of all pairs of sequences and select the sequence that minimises the sum of distances (which represents the *centroid* of the set). Then, based on the pairwise alignments, we can iteratively find the multiple alignment by simply adding gaps in the current alignment. The overall workflow for the center-star method is presented in Figure 11.4

Center_Star *Input:* A set $\mathcal{S}$ of sequences.

Output: A multiple alignment $\mathcal{M}$ with a SP-score *at most* twice that of the optimal alignment of $\mathcal{S}$.

1. Compute $\mathcal{D}(\mathcal{S}_i, \mathcal{S}_j)$ for all $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{S}$.
2. Find the center sequence $\mathcal{S}_c$ which minimizes $\sum_{i=1}^{k} \mathcal{D}(\mathcal{S}_c, \mathcal{S}_i)$.
3. For every $\mathcal{S}_i \in \mathcal{S} - \{\mathcal{S}_c\}$, choose the optimal pairwise alignment between $\mathcal{S}_c$ and $\mathcal{S}_i$.
4. Introduce gaps into $\mathcal{S}_c$ so that the multiple alignment $\mathcal{M}$ satisfies the alignments found in Step 3.

Heuristics methods

The star method suffers from several flaws in terms of time and space requirements, but also in the quality of the final alignment. Furthermore, the centre star method is highly brittle to the choice of the reference sequence. Hence, several heuristics have been devised to alleviate this problem.

Progressive alignment *Progressive alignment* is based on the idea of iteratively aligning the most closely related sequences until all sequences are aligned. Several algorithms have been developed based on this idea such as *ClustalW* (Thompson et al., 2002), *T-Coffee* (Notredame et al., 2000) and *ProbCons* (Do et al., 2005). All these algorithms follow the same three main steps

1. *Computing pairwise distance scores for all pairs of sequences*, through the (triangular) distance matrix containing $\mathcal{D}(\mathcal{S}_i, \mathcal{S}_j)$ for all $\mathcal{S}_i, \mathcal{S}_j \in \mathcal{S}$ (often expressed as the percentage of mismatches). The choice of *alignment algorithm* and *weight matrix* both highly influence the final multiple alignment.

2. *Generating the guide tree based on sequence similarities* in the distance matrix to obtain a *hierarchical clustering*. Any linkage function and clustering algorithm can be used to obtain this iterative grouping (dendrogram of sequences similarities).

3. *Aligning the sequences iteratively along the guide tree*, by starting from the leaves and moving up the tree. Each internal node connecting several sequences represents an alignment of the corresponding sequences. This process is repeated until the root node.

The process of iterative alignment implies several multiple alignments between subsets of the sequences (some of which already aligned in a previous step), which can be done through the principle of *Profile-Profile alignment*.

Profile-Profile Alignment Given two aligned *sets* of sequences $\mathcal{A}_1$ and $\mathcal{A}_2$, the profile-profile alignment introduces gaps to $\mathcal{A}_1$ and $\mathcal{A}_2$ so that both of them have the same length. In order to determine this alignment, we need a scoring function such as

$$PSP\left(\mathcal{A}_1\left[i\right], \mathcal{A}_2\left[j\right]\right) = \sum_{x,y} g_x^i g_y^j \delta\left(x, y\right)$$

where g_x^i is the observed frequency of symbol x in the column i and $\delta(x, y)$ is the distance between symbols x and y (as defined in our weight matrix). Hence, our aim is to find an alignment between the two sets that in order to maximise the PSP score. The overall workflow for the progressive alignment methods is summarised in Figure 11.5

Iterative methods The main limitation of the progressive alignment is that it will not try to realign the sequences once the MSA is found. Hence, the final alignment is highly brittle to the quality of initial alignments, and is not guaranteed to converge to the global optimum. In order to alleviate these flaws, *iterative methods* introduce heuristics that starts with a progressive alignment and then iteratively improves it. Examples of iterative methods are *MAFFT* (Katoh et al., 2002) and *MUSCLE* (Edgar, 2004). These algorithms are based on two ideas

1. Generating a *draft* multiple alignment as fast as possible, usually through slightly modified progressive alignment method. First, the distance matrix is

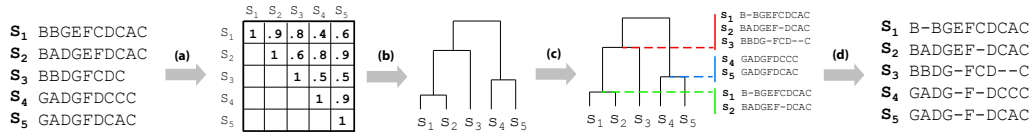


Figure 11.5: Summary of the *progressive alignment* algorithm. (a) The similarity matrix is computed based on pairwise alignments. (b) The guide tree is obtained from this matrix. (c) By going up the tree, each node generates a specific alignment, between subsets of sequences. (d) When the root of the tree is reached, we obtain the set of multiple alignments.

computed faster by discriminating sequences based on the symbols frequency. Second, the *Unweighted Pair-Group Method using Arithmetic mean* (UPGMA) method is used to perform clustering Finally, the PSP score may favor gaps as it relies on the direct sum of weighted symbol distances (and gaps are considered as symbols). This can be alleviated by using the *log-expectation* score in the profile-profile alignment

$$LE\left(\mathcal{A}_1\left[i\right],\mathcal{A}_2\left[j\right]\right)=\left(1-f_i^G\right)\left(1-f_j^G\right)log\left(\sum_{x,y}f_i^x f_j^y\frac{p_{xy}}{(p_x p_y)}\right)$$

where f_i^G is the proportion of gaps in $\mathcal{A}_1$, f_i^x is the proportion of symbol x in $\mathcal{A}_1$, p_x is the overall proportion of symbol x and p_{xy} is the probability that x aligns with y. It should be noted that

$$\frac{p_{xy}}{(p_x p_y)}=e^{\delta(x,y)}.$$

2. In the second stage, the distance matrix is computed by first finding the fraction D of identical symbols shared by two aligned sequences, and computing

$$-\log\left(1-D-\frac{D^2}{5}\right).$$

The progressive alignment is iterated but re-alignment of the sequences is performed only when there are changes relative to the original tree.

The MSA algorithms that will be used here are *ClustalW* (Thompson et al., 2002), *Muscle* (Edgar, 2004), *MAFFT* (Katoh et al., 2002), *ProbCons* (Do et al., 2005) and *TCoffee* (Notredame et al., 2000). We will compare different algorithms with various types of symbolic informations and different weighting matrix to assess different structural properties of music.

Evaluating MSA results

As MSA can produce widely varying results, we need objective measures of the alignment quality. In genetics, this is usually performed by comparing the alignment to a known reference sequence. However in our case, as we do not have a specific

reference, we rely on reference-free evaluation methods. The simplest quality metrics are the *Total columns* (TC) aligned, the *Q-Score (*percentage of aligned pairs over the total number of pairs) and the *Sum of Pairs* (SP) score as defined previously (Section 11.1.2). However, more advanced reference-free metrics have also been developed. The *Z-Score* (Ahola et al., 2006) relies on importance sampling and statistical profile analysis for counting the number of significantly conserved positions in the alignment. The *Multiple Overlap Score* (MOS) (Lassmann and Sonnhammer, 2005) identifies alignment quality by expressing the overlap among groups of aligned sequences. The *Information Content* (IC) (Hertz and Stormo, 1999) provides a log-likelihood scoring scheme based on *a priori* probabilities of symbol occurrence. The *APDB* distance (OSullivan et al., 2003), *Root Mean Square Deviate* (iRMSD) and *Normalized iRMSD* (iRMSD) (Armougom et al., 2006) are based on the idea that if an aligned pair is correct, then the neighborhood of this pair should also be aligned. Therefore they are derived by computing the percentage of aligned neighbors across all positions in the alignment. Finally, the *Mean Distance* (MD) and *Normalized MD* (NorMD) (Thompson et al., 2001) combines column scoring and similarity scores by performing a ratio of the number, length and similarity of aligned subsequences.

Motif mining

Once the multiple alignment is obtained, it is straightforward to perform a *motif mining* analysis. Indeed, as sequences are now all globally aligned, the search of motifs can be performed by looking for highly conserved "blocks" of symbols. A motif is a particular subsequence that occur with a significant number of repetitions across the set of aligned sequences and with eventual small variations. Here, we rely on the MEME (Multiple EM for Motif Elicitation) algorithm (Bailey et al., 2006) to perform motif discovery from the results of the multiple alignment. MEME works by searching for repeated, ungapped sequence patterns that occur in an aligned set of sequences. Using a process akin to local sequence alignment based on a set of selected *seeds*, MEME searches for statistically significant motifs in the input sequence set by sliding these seeds over the multiple alignment.

Computing the consensus sequence

Once the multiple alignment and the motif analysis are computed, we can obtain the consensus sequences of different motifs, which represents the "mean" sequence of a particular motif. These consensus sequences can allow to study different properties of a group considered as a motif in a single glance (Bailey et al., 2006). Even though several statistical methods have been developed for constructing the consensus sequences, we will only rely here on the method based on frequencies, provided by the MEME Suite. An example representation of the consensus sequence is displayed in Figure 11.6

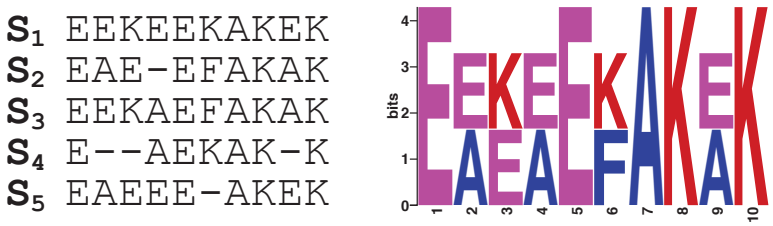


Figure 11.6: Possible representations of the consensus sequences

11.2 Harmonic sequences

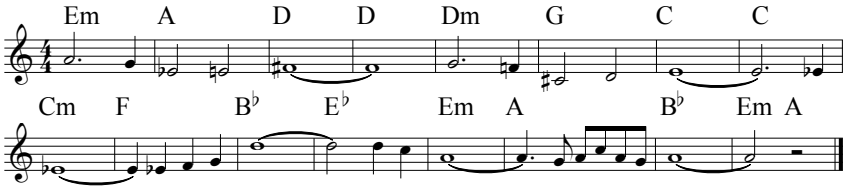
In this section, we define different types of symbolic sequences obtained analyzing the harmonic features of each song. Moreover, we describe the construction of *ad hoc* weighting matrices for the alignment of harmonic-based sequences.

11.2.1 From harmonic progressions to symbolic sequences

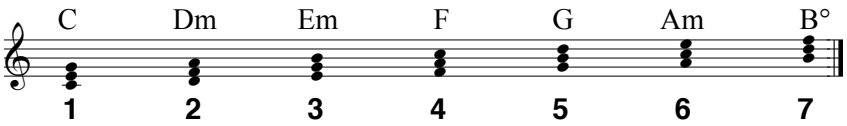
The lead sheet notation depicted in Figure 11.7a provides a natural interpretation of a harmonic progression as a symbolic sequence. The idea here, is to produce a higher abstract description of the harmonic content of a song, based on the sequence of symbols describing its chords. To achieve this aim, we interpret chords as the degrees of a major tonality (see (Piston et al., 1978, Ch. 2)). In music, a tonality is defined as the collection of triads constructed from a certain scale as it is depicted in Figure 11.7b. Assuming a whole song to be written in a single tonality would be simplistic. However, it is possible to segment it in tonal regions, defined as subsequences of consecutive chords belonging to the same tonality.
The algorithm associating a tonality to each chord is based on the *spiral array* (Chew, 2002). In this model pitch classes are represented as points of a helix. Thus, a chord (a collection of several pitch classes) is represented by the convex hulls of the points describing its pitches on the spiral. The centre of gravity of the convex hull is the representative point of the whole chord. This construction allows to describe several musical entities. Even a whole tonality can be represented in the spiral array, when considered as a collection of pitch classes. Given a sequence of chords, the computation of the 3 nearest tonalities to each chord in the spiral array allows to define stable tonal regions on the whole harmonic structure and to avoid sudden tonality changes for small modulations[1].

We shall consider four different kind of symbolic sequences. Three of them are deduced directly form the tonal analysis of the harmonic progression, the fourth is

[1]A harmonic sequence like $Dm - C - Am$ is interpreted as a collection of chords belonging to the tonality of C major (see the chord labeled as 2,1 and 6 in Figure 11.7b). However, it is desirable that the sequence $Dm - C - Am - B\flat$ be interpreted as a harmonic sequence in F major rather than a C major modulating to F major or B♭. It is common practice to substitute the third degree of a tonality with a major triad (from Em to E in C major). In this case, if a E major triad appears in the middle of a sequence of chords belonging to the tonality of C major, the algorithm hide this brief modulation, maintaining the same tonality.



(a) Miles Davis - *Tune Up*



(b) Triad harmonization of the C major scale.

Figure 11.7: From chords to symbols. (a) In a lead sheet, the standard chord notation is substituted by symbols. (b) The triad harmonisation of the diatonic scale of C and its seven degrees.

based on a semiotic analysis of music. As a paradigmatic example we refer to the harmonic structure of Tune Up, as transcribed in Figure 11.7a.

Degree

Neglecting the information regarding the tonality of each song, we consider only the degree associated to each chord. For instance, considering the harmonic structure of *Tune Up* we obtain 251251251425125, being *Em* the second degree of the tonality of *D* major, *A* the fifth degree and so forth. The repetition of the same degrees pattern points out the extensive use of perfect cadences across the whole piece (see (Piston et al., 1978, Ch. 12)).

Spike

We consider the sequence built by taking the differences between the tonalities associated to two consecutive chords. This difference is defined as the cardinality of the set given by the union of the altered notes of each tonality. This is equivalent to counting the number of counterclockwise steps separating the two tonalities on the circle of fifths (see Figure 11.8). In the case of *Tune Up*, this sequence is 00020020040440. Geometrically speaking, this sequence corresponds to a sequence of spikes whose height depends on the modulation occurring in the harmonic structure.

Tonality

In this case, each chord is substituted by its major tonality. In our example, we have $DDDCCCB\flat B\flat B\flat DDB\flat DD$. Whereas the previous sequence could be visualized as a succession of spikes, this one can be thought of as a step function. The dictionary used to describe tonality is described in Figure 11.8.

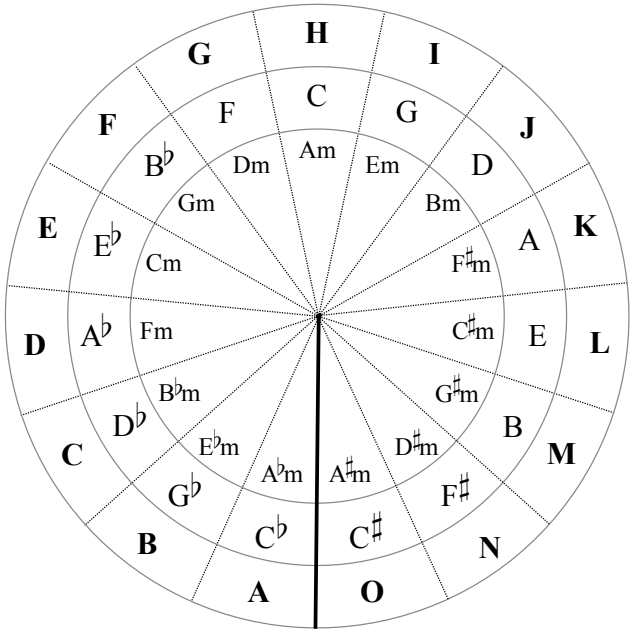


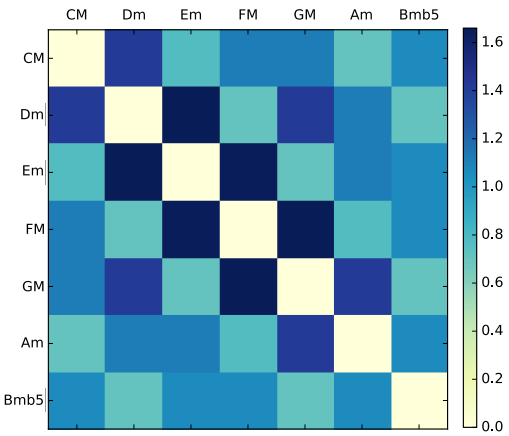
Figure 11.8: In the circle of fifths major (and relative minor) tonalities are organized in relationship to the altered notes they contain. Two tonality a step apart differ of a single note. The only exception is represented by the tonalities of $C\sharp$ and $C\flat$, which are separated by a thick line. The bold letters surrounding the circle correspond to the alphabet used to build the tonality class of sequences.

Semiotic annotation

It is natural for a (trained) listener to intuitively segment music while listening to it. The automatic segmentation of a music piece into meaningful parts (like introduction, choruses and verses) is a difficult task and it has been tackled in several ways. For example, in (Foote and Uchihashi, 2001) a subdivision is derived from the analysis of rhythmic changes occurring in the song. Another strategy described in (Aucouturier et al., 2005) is based on the evaluation of the evolution of the timbre. In (Jensen, 2007) several music features are interpreted to provide a segmentation of the song in choruses, verses and so forth. It is also possible to define formal techniques to obtain a segmentation of a song in semiotic blocks (Bimbot et al., 2012). The semiotic characterization of a song consists of the definition of a labelling function on a set of symbols and taking its values on the set of semiotic blocks identified during the segmentation. The association between blocks and labels highlights the similarity between different parts of the song. For instance, it is possible to associate more than one block to the same label or to define a *variation* of a preexistent symbol. This procedure leads to a straightforward definition of a degree of similarity among the blocks.

Summary

The class *degree* describe the harmonic cadences used in a composition. Its dictionary is composed by 9 symbols. 7 of these symbols correspond to the degrees of the



(a) Degrees' distance matrix.

	C	D	A	B	X	M	AB	AX	M*
C	1	0.7	-1	-1	-1	0.5	-1	-1	-1
D	0.7	1	-1	-1	-1	0.5	-1	-1	-1
A	-1	-1	1	0.4	-1	-1	0.5	0.8	-1
B	-1	-1	0.4	1	-1	-1	0.5	0.1	-1
X	-1	-1	-1	-1	1	-1	-1	0.5	-1
M	-1	-1	-1	-1	-1	1	-1	-1	0.9
AB	-1	-1	0.5	0.5	-1	-1	1	-1	-1
AX	-1	-1	0.8	0.1	0.5	-1	-1	1	-1
M*	0.2	0.2	-1	-1	-1	0.9	-1	-1	1

(b) Semiotic weighting matrix.

Figure 11.9: Two weighting matrices expressing the similarity between degrees of a tonality (left) and semiotic labelling (right). The former is computed considering the distances of chords in the spiral array, the latter is deduced from the similarity of the block retrieved by the semiotic segmentation of music.

tonality depicted in Figure 11.7b, the 8th one denotes the absence of a chord (silences and percussive breaks); a last symbol is used to label chords involved in small modulations or harmonic substitutions. The two classes of sequences named *spike* and *tonality* represent the modulations (Piston et al., 1978, Ch. 8) occurring during the piece. The former class is invariant respect to musical transposition, while the latter is sensitive to it. Their dictionaries have 15 symbols corresponding to the 15 major (and relative minor) tonalities. Finally, the sequences belonging to the semiotic class are the only hand-made ones we consider and they reflect the perception of musical blocks of a trained listener.

11.2.2 Weighting matrices

One of the main ingredient of sequence alignment is the weighting matrix. Here, we consider four different weighting matrices in order to deal with the specific dictionaries of the classes of sequences described above. Let $n \in \mathbb{N}$ the number of considered symbols.

- *The binary matrix* $B = b_{ij}$ is built as an identity matrix associating a positive score to exact matches, i.e. diagonal entries b_{ii} and a uniform negative score to the elements b_{ij} with $i \neq j$ corresponding to a mismatch.

- *The linear matrix* L associates a score to a given pair of symbols by taking into account their distance from the diagonal and thus from maximum score (exact match). Let $1 \leqslant i, j \leqslant n$ be two natural numbers. The entries of the

matrix L are defined as

$$l_{ij} = \begin{cases} n & \text{if } i = j \\ n - |i - j| & \text{otherwise} \end{cases}.$$

- *The constructed matrix* $C = c_{ij}$ is built to deal with the seven degrees of a tonality. It is a symmetric matrix and its entries are elements of the set $\{-1, 0.4, 0.6, 1\}$.

$$c_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0.6 & \text{if } n - |i - j| = 2 \\ 0.4 & \text{if } n - |i - j| = 5 \\ -1 & \text{otherwise} \end{cases}.$$

  These particular choice stresses out the natural relationships among degrees. Let $d \in \{1, \ldots, 7\}$ be a degree of a tonality. By construction, it shares two pitch classes with the $d + 2$ and $d + 5$ degrees, modulo 8.

- *The alternate matrix* is a refinement of the constructed one. It is built by computing the distance between pitch-class triads in the spiral array. There are several possibilities to interpret a chord of n notes as a point of a metric space (Bergomi et al., 2014a), for instance considering its pitches as coordinate of a point in $\mathbb{R}^n$, or its pitch classes as a point of the n-dimensional torus $\mathbb{T}^n = (\mathbb{Z}/12\mathbb{Z})^n$. In Figure 11.9a the distance between triads belonging to the harmonization of a tonality is computed considering the centre of gravity of triangles representing triads in the spiral array. This particular choice reflects the perceptual relation among the degrees of the tonality. For instance the first, sixth and third degrees are near while the second is the farthest, followed by the seventh, fourth and fifth degrees.

- *The semiotic similarity matrix* is obtained considering the similarity defined by the semiotic labelling function. The matrix is depicted in Figure 11.9b. The matches on the diagonal of the matrix have value 1, mismatches between unrelated symbols correspond to -1 entries of the matrix. The distance between similar labels nuanced according to their definition given in (Bimbot et al., 2012).

11.3 Applications

11.3.1 Database, visualisation and notation

We considered a collection of 138 songs belonging to the Quaero database. These compositions are performed by 72 different artists and cover a timespan of 50 years, from 1962 to 2012. In order to show how even heterogeneous music styles can be tagged as *popular*, here follows a list of the artists we considered in our analyses.

Quaero's artists

50 Cents	Daughtry	Joan Baez	Pink Floyd
ACDC	Destiny's Child	Judas Priest	Platiskman
Aerosmith	Dillinger	Justin Timberlake	Pucho and his latin
Ali Farka Toure	Dolly Parton	Kiss	soul brothers
Amy Winehouse	Eminem	Lil Wayne	Puff Daddy
Bjork	Eric Clapton	Ludacris	Faith Evans
Bobby McFerrin	Faith Hill	Madcon	Radiohead
Britney Spears	Finger Eleven	Madonna	Ray Charles
Buckcherry	Flo Rida	Mariah Carey	Run DMC
Buenavista Soc...	Franck Zappa	Massive Attack	Scorpions
Carl Douglas	Georges Michael	Michael Jackson	Shack
Cher	Goran Bregovic	Moby	Sweet
CoCo Lee	Gwen Stefani	Neil Young	The Beatles
Cranberries	Jedi Mind Tricks	Obituary	The Cure
D'Angelo	Jim Jones	Patrick Hernandez	The Fall A Sides

The harmonic transcription of each song has been computed using the algorithm presented in (Mauch, 2010). The association between audio and chord symbols is not injective. Consider a chord composed by the pitch classes C, E, G, A. It can be interpreted as an minor seventh chord, or a major *add6*. In order to deal with a small dictionary of chord symbols and to reduce the ambiguity in their retrieval, we transcribed each songs utilizing only major, minor and diminished triads.

From this computation we construct three classes of symbolic sequences and we will consider a fourth class given by the semiotic sequences.

Let $S = \{Degree, Spike, Tonality, Semiotic\}$ be the set of these classes. Let $M = \{binary, linear, constructed, alternate, semiotic\}$ be the set of weighting matrices and $A = \{DTW, ERP, LCSS, NW\}$ the collection of alignment algorithms we shall consider. We denote a clustering of our dataset as an element of the set $C \subset S \times M \times A$.

We visualize the information retrieved by the computation of the pairwise global alignment of symbolic sequences as a polar dendrograms. The information carried by a dendrogram concerns the similarity and the configuration (clustering) of data. Each joining between sequences (or clusters of sequences) is represented by the splitting of a circular segment into two smaller ones. The position of the split respect to the centre of the circle allows two retrieve the similarity between two clusters. Outliers are fused to preexistent clusters near the centre of the circle. In Figure 11.10, the two sequences grouped in the red cluster are very similar, while the object labeled as *50 Cents* is an outlier of the big grey cluster on its left.

Finally, the multiple sequence alignment of a particular clustering $c \in C$ shall be computed comparing the performances of five algorithms. The analysis of motifs highlighted by the alignment shall be computed using MEME.

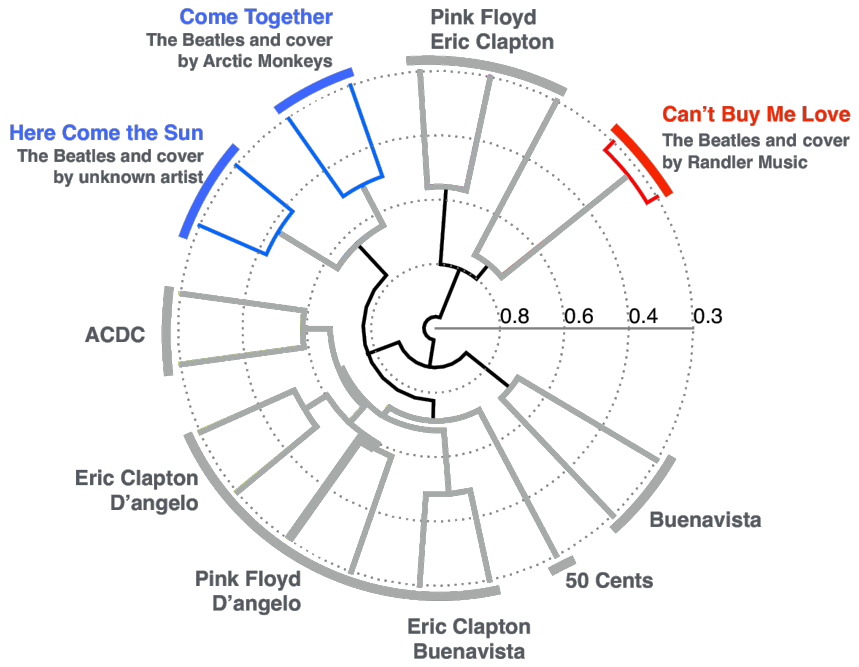


Figure 11.10: Dendrogram obtained by evaluating the dissimilarity among 19 songs of Quaero and 3 Beatles' covers contained in the original set.

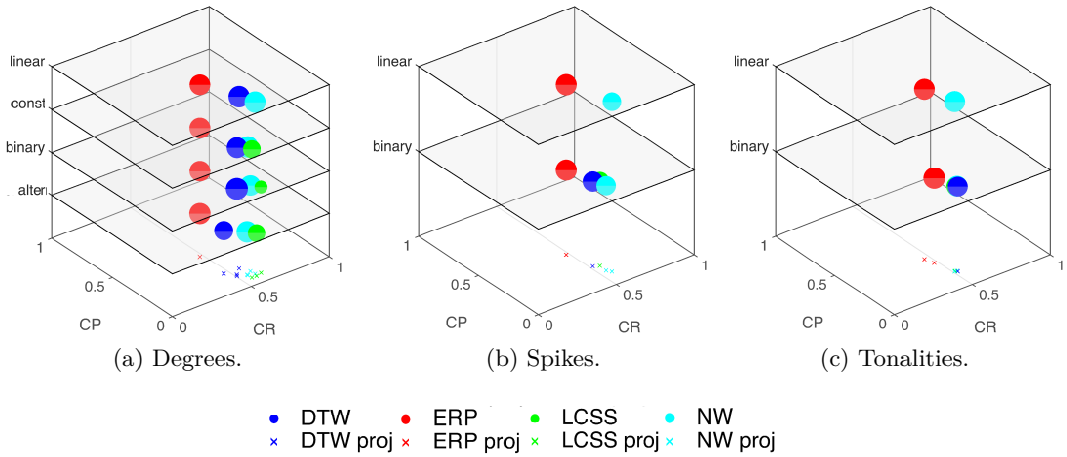
11.3.2 Cover recognition

In this first application, we consider a dataset composed of 19 Quaero's songs belonging to different genres and by 3 cover tracks of songs by The Beatles that are part of the original set. This collection of songs is processed to obtain their sequences of degrees, which are aligned using the NW algorithm weighted with the alternate matrix. This test aims at exhibiting the coherence of the harmonic information and the detection of tonal regions. The resulting dendrogram is displayed in Figure 11.10, where the positions of original songs and their cover tracks is highlighted. As we can see, the original Beatles' songs are always coupled with their respective covers, albeit a non-neglectible distance. This points out the structural changes characterizing the alternative versions of these songs.

11.3.3 Metadata and clustering

Consider a clustering $c \in C$ and denote by c_i its clusters, for $i \in \{1, \dots, n\} \subset \mathbb{N}$. In order to compare these groupings with the traditional genre and artist classification paradigm, we rely on the set of metadata provided with the analyzed songs. The 1-NN accuracy of c respect to the metadata is computed. The cluster precision and the cluster recall in terms of retrieval of genres and artists has been computed for every cluster $c_i \in c$. This information is encoded in the 5-dimensional visualizations depicted in Figure 11.11 on the following page.

As we can see, best results are obtained with the pairings $(alternate, ERP)$ considering degrees, $(linear, NW)$ for spike sequences and $(binary, NW)$ considering



(a) Degrees. (b) Spikes. (c) Tonalities.

Figure 11.11: Evaluation of several harmonic-oriented clusterings in relation to a genre recognition task. Different clusterings are represented as colored spheres of variable radius in the space. The colour represent the alignment algorithm used to obtain the clustering. The size of the spheres corresponds to the 1-NN accuracy of the clustering, while the height of the spheres depends on the weighting matrix used to generate the clustering. On the cluster precision/cluster recall plane ($z = 0$), the projection of each sphere is depicted as a cross.

the class of tonality sequences. The idea, is to focus on the information provided by the evolution of the composition in terms of modulations. The resulting dendrograms are depicted in Figure 11.12.

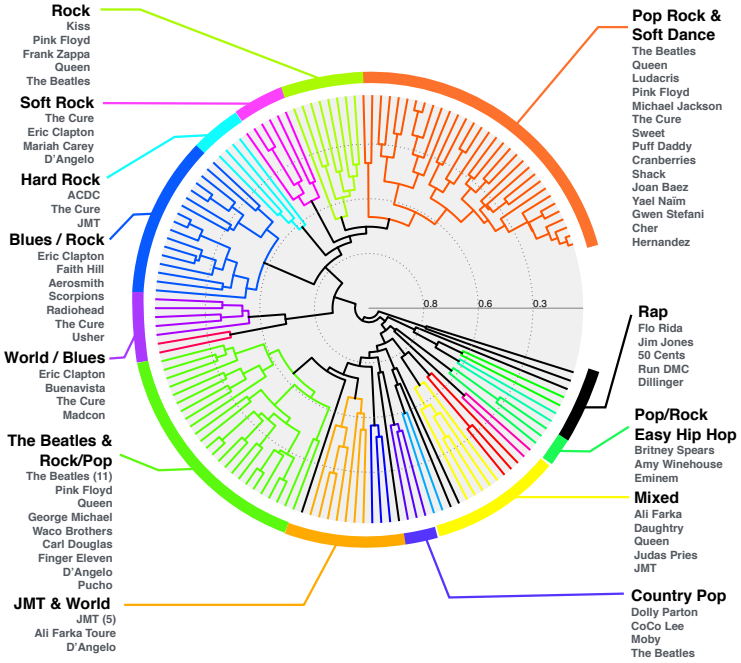
It is important to notice that well recognisable artists and genre tend to be mostly grouped together. For instance, in both clustering several songs by The Beatles are grouped together and the Hip Hop songs are segregated from the others. At the same time, it is possible to notice a certain degree of contamination among songs and artists from different genres. Even though this observation appears obvious for most music listeners, it is difficult to exhibit it using local signal-based features.

11.3.4 Semiotic clustering

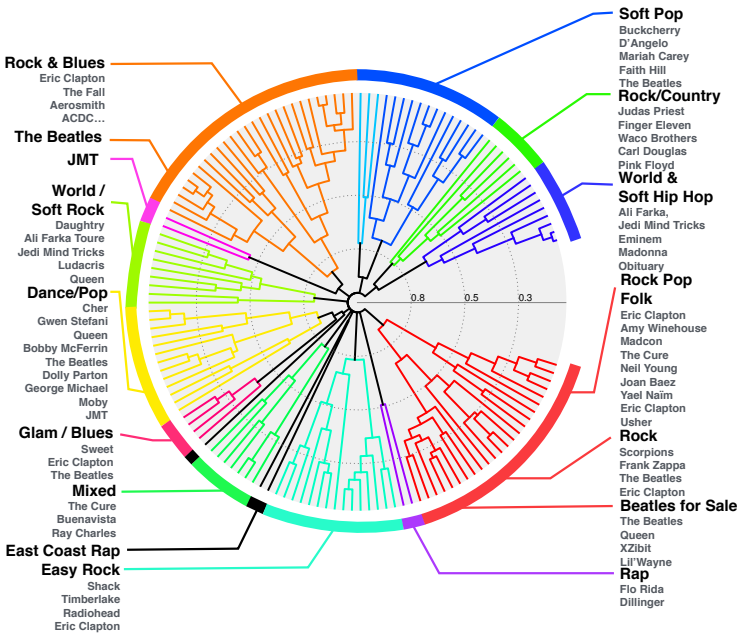
The computation of the *distance* between semiotic symbolic sequences, produces the clustering of Figure 11.13a. Such a dissimilarity has been computed using the NW algorithm and the weighting matrix deduced from the semiotic labelling depicted in Figure 11.9b. The clusters are surprisingly well shaped, however some aberrations appear. For instance a really homogeneous rock/pop group of Eric Clapton's and Beatles's pieces labelled in the figure as *Pop Rock* is followed by another cluster (*Hip Hop*) where hip hop and rock songs are mixed together.

11.3.5 Towards semantic clustering

The previously discussed hand-made semiotic segmentation can be refined by considering the information carried by the analysis of the sequences of degrees. If we consider the *Pop Rock* and *Hip Hop* clusters of Figure 11.13a, we can clearly see

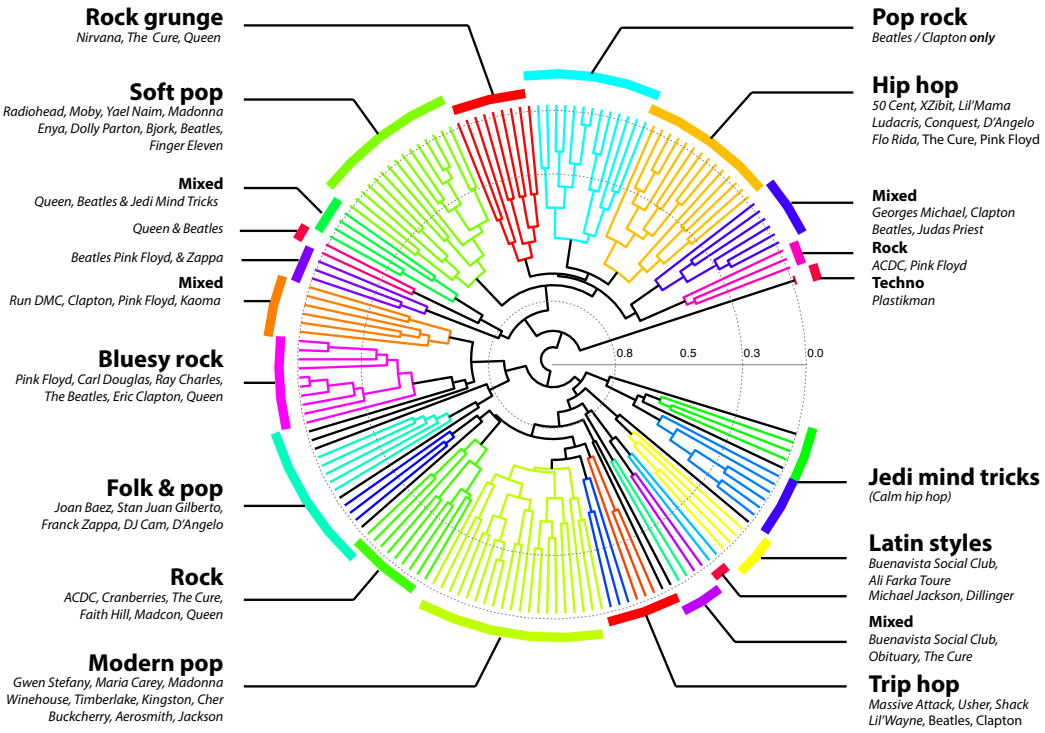


(a) Spikes,linear, NW.

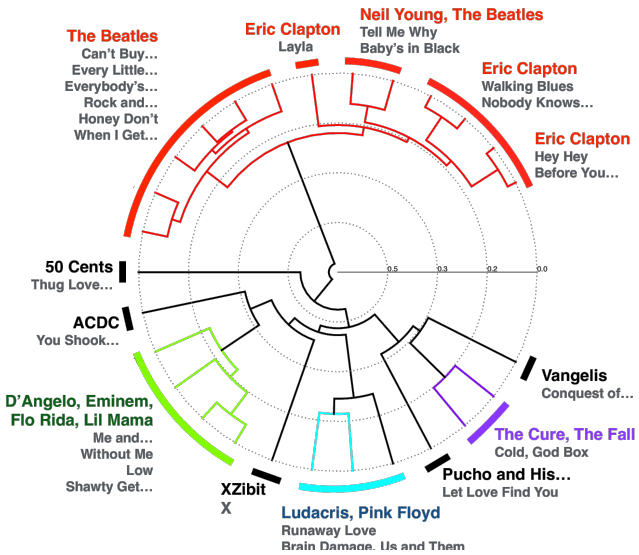


(b) Tonalities, binary, NW.

Figure 11.12: Two possible clusterings. Each cluster has been labeled coherently with the genre represented by its objects. Clusters whose objects do not share a similar genre are labelled as Mixed. Big clusters have been labelled according to their subgroups. Finally, the cluster named as *Beatles for Sale* in (b) owes its name to the presence of a neat groups of songs belonging to this album.

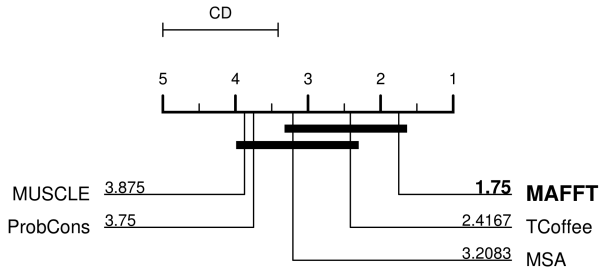


(a)

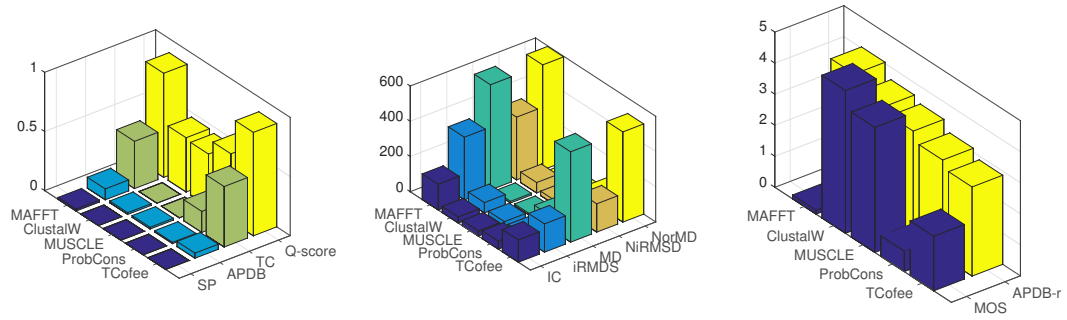


(b)

Figure 11.13: Interaction between the semiotic segmentation and the harmonic-based sequences. (a) The polar dendrogram representing the hierarchical organisation of the semiotic sequences aligned with the NW algorithm and the semiotic weighting matrix. Clusters are genre-wise labeled. Mixed clusters corresponds to incoherent groupings in terms of genre. (b) Re-organisation of the *Pop Rock* and *Hip Hop* clusters of (a) through the alignment given by the combination $(Degrees, alternate, NW)$. The new dissimilarity measure has been computed cluster-wise, enhancing the genre retrieval obtained by the semiotic approach.



(a) MSA comparison.



(b) Reference-free MSA evaluation

Figure 11.14: Reference-free methods are represented in three different barplots according to their order of magnitude.

that the homogeneity of the former and the heterogeneity of the latter both in terms of genres and artists retrieval. In order to reshape these two clusters while maintaining the knowledge provided by the prior semiotic alignment, we compute a further cluster-wise alignment. We considered the sequences of degrees and computed their dissimilarity through the NW algorithm and the alternate weighting matrix. Figure 11.13b shows the new clustering of the two groups of songs, where songs belonging to the *Pop Rock* cluster occupies the upper part of the dendrogram, while the *Hip Hop* cluster's songs are entirely represented in the lower half of the dendrogram. As we can see, the songs belonging to the first cluster are joint in a single group at a dissimilarity value of 0.24, while it is necessary to climb up the whole dendrogram to obtain a cluster composed by the whole set of the *Hip Hop* cluster's songs.

In the upper-half of the polar dendrogram, the songs by *The Beatles* are clustered with those by Eric Clapton. The small cluster composed by four Clapton's blues songs is clustered with *Tell Me Why* and *Baby's Black* by Neil Young and The Beatles respectively. *Layla* is an outlier of this cluster and is the only ballad that has been considered. By observing the reorganisation of the *Hip Hop* cluster, we can notice how the hip hop songs are grouped together on the bottom-left of the figure. The only exception represented by the ACDC song which is considered as an outlier of this cluster. The lower right part of the dendrogram is occupied by rock songs.

11.3.6 Motif mining and molecular clock

The analysis of the clusterings generated by different types of musical information, weighting matrices and alignment algorithms allows to grasp the contamination affecting artists and genres. However, pairwise alignment cannot provide a broad overview on the structure of the whole sequences. The multiple sequence alignment of the clustering $(Tonality, binary, NW)$ has been computed using five different algorithms. Each results has been evaluated on 11 reference-free quality metrics. In Figure 11.14a on page 181 the whole set of evaluation in summarised in a single diagram, that highlights MAFFT as the algorithm giving the best results. A more informative representation of the MSA algorithms is represented in Figure 11.14b. Best results are obtained using MAFFT, that presents a lower quality measure than the other algorithms only evaluating its multiple overlap score. Finally, a motif analysis has been performed using MEME, in order to identify significant modulation patterns in the overall structure of the multiple aligned sequences.

Figure 11.15 shows the results of these analyses. By construction, sequences belonging to the *tonality* class represent the modulations occurring during a song and are sensitive to musical transposition. Hence we retrieve the particular tonal and modulation choices recurring in each cluster.

At first sight, it is possible to notice that although in equal temperament all major (and relative minor) tonalities are equal, they are not equally distributed among the clusters. Moreover, it is not surprising that the most recurrent modulations are given by small displacements on the circle of fifths. Popular music is often composed for a voice, as the singer could not be at ease by singing a suite of incoherent altered notes without any link given by a clear modulation, generally starting on a meaningful pivot chord (Piston et al., 1978, Ch. 8).

The *Rock and Blues* cluster contains four recurring motifs, based mainly on the tonalities of A and D, that are a step apart on the cycle of fifths (depicted in Figure 11.8). This respects the typical blues structure that tends to be stable on a single tonality (considering only triads). It is interesting to note how the blues paradigm influenced other compositions: among the artists listed in the *Rock and Blues* cluster of Figure 11.12b, we can find names like Aerosmith and Pink Floyd but also D'Angelo and CoCo Lee, whose genre is nearer to the classic pop than the other artists. The richer motif corresponding to this cluster is the third one, where three tonalities are considered, again D, A and E are visualisable as three consecutive points on the cycle of fifths. If we look more closely at the following cluster (named *Soft Pop*), we can see a motif shared more than once by the whole set of songs, involving only one tonality (D). The second highlighted motif of this cluster is more complex, suggesting several possible modulations between four consecutive tonalities on the cycle of fifths. Such a variability often characterises bridges and modulations which follow the climax of compositions endowed with a more complex structure. It is surprising how this feature is shared by artists like Faith Hill, Mariah Carey, Buckcherry and finds its origin in songs by *The Beatles*. Clusters with heterogeneous nature are points of interest of our analyses. The motifs characterizing the *World-Soft Hip Hop* are shared among songs by Ali Farka Toure, Madonna and Jedi Mind Tricks. It is interesting to observe how the cluster labelled as *Rock Pop & Folk* represents a counterclockwise shift on the circle of

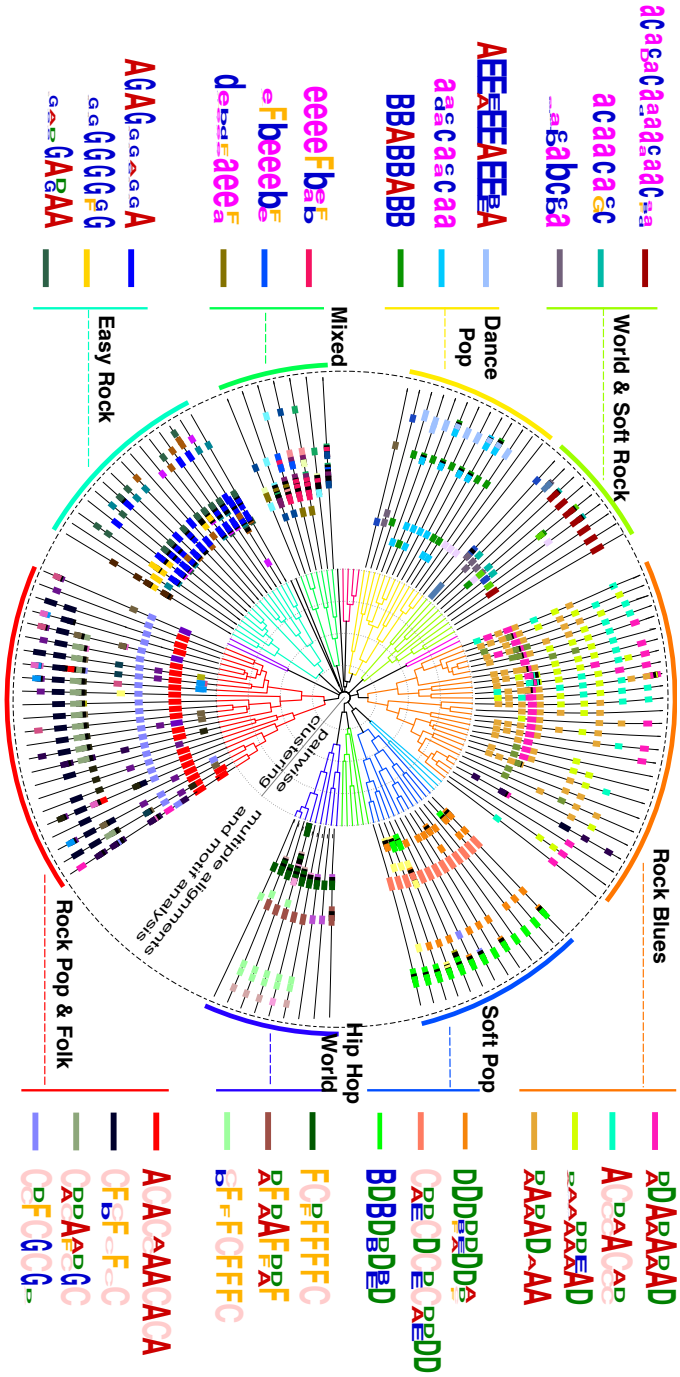


Figure 11.15: The polar dendrogram constituting the centre of the figure is the clustering obtained considering sequences of the class Tonality, aligned with the NW algorithm and the binary weighting matrix. The radial segments represent the result of the multiple sequence alignment. Recurrent modulation patterns have been highlighted as coloured segments. Finally, the consensus of the most relevant motifs have been depicted for each cluster. For the sake of simplicity, the consensus sequences are composed only by capital and lowercase letters, representing natural and flat tonalities, respectively (the symbol C denotes the tonality of C major, while c the major tonality of $C\flat$).

fifths respect to the previous one. The tonalities involved in its *common patterns* are $B\flat, F, C, G, D, A$. The third motif of this cluster coincide with the first of the previous one. In addition, a particular modulation motif concerning the tonalities of A and C suggests the utilisation of parsimonious voice leadings (Pearsall, 2012, Ch. 1, p. 10) or secondary dominants (Piston et al., 1978, Ch. 14). This techniques are subtler than a modulation by pivot chord. This approach is homogeneously shared among the songs forming this cluster.

The *Easy Rock* cluster is characterised by a strong tonal stability. It is not surprising that $G\,major/E\,minor$ is the most used tonality in a cluster grouping authors as Eric Clapton and the britpop band Shack. However, even compositions by Justin Timberlake and Radiohead are part of this cluster, that mean they modulate using the same modulation motifs. Considering the *Mixed* cluster, three main motifs are shared by compositions of Ray Charles, Buenavista Social Club and The Cure. The *Dance* cluster is of particular interest considering how three long motifs are shared among artists like Gwen Stefani, and Cher, but also Jedi Mind Tricks, Eric Clapton and two compositions by The Beatles. Finally, we observe that the second consensus sequence associated to the *World/Soft Rock* cluster is a half-step transposition of one of the *Rock Pop & Folk* grouping.

11.4 Discussion and perspectives

In this chapter we considered a collection of symbolic sequences derived from the harmonic analysis of the chords progressions of 138 pop songs. These sequences describe the structure of each song in terms of cadential patterns, modulations and semiotic blocks. The global pairwise alignment of each type of sequences has been computed testing different algorithms and weighting matrices. In particular, we suggested a specific collection of these matrices in order to evaluate the harmonic data retrieved from the audio.

The pairings of weighting matrix and distance algorithm have been tested on several applications. First, the detection of cover tracks, where the global pairwise alignment retrieves the relationship between the original and the cover track and provides a measurement their dissimilarity. Second, respecting the classical tasks of MIR, for each clustering of the dataset obtained choosing a particular type of symbolic sequence, a distance algorithm and a weighting matrix, we computed its $1 - NN$ accuracy, the cluster precision and cluster recall in terms of retrieval of genres and artists. As a third application, we showed how the results of the alignment computed using the semiotic sequences can be improved in terms of coherence, by adding the harmonic information concerning the cadential patterns of each song. Finally, taking advantage of the analyses conducted in the applications described above, the multiple alignment of the sequences representing the changes of the tonal centres of each song has been computed using five different algorithms. A collection of reference-free methods allowed to evaluate the results obtained by these algorithms and to choose a particular multiple alignment of the sequences of the dataset. The analysis of recurrent, coherent motifs of the multiple aligned sequences highlights the transfer of musical inspiration among several artists whose compositions do not necessarily belong to the same genre, or time. To conclude, we computed the consensus sequences of the most relevant motifs shared cluster-wise

by the compositions we analysed. These paradigmatic modulation choices represent a common harmonic strategy used in the compositions *contaminated* by the same motif. The standard analyses based only on the signal content of the audio neglect this broad high-level viewpoint on the common strategies employed in songs that would result different, if compared using standard descriptors. Multiple sequence alignment of harmonic-based sequences provides a tangible evidence of the transfer of music inspiration over time.

Twelve

Musical Persistence Snapshots

In Part III we used persistent homology to compute a descriptor of music, based on the deformation of the geometry of the *Tonnetz*. This construction neglects one of the most important features of music: a composition evolves in time. This evolution allows the composer to introduce a musical idea, then describe it and finally proceed to a new scenario. Would it be possible to refine our analysis considering several configurations of the *Tonnetz* in time? What follows is a primal attempt to include this time-dependency in the *vertical* topological analysis of the deformed *Tonnetz*, in order to compare the evolution in time of two compositions.

12.1 Persistence and time varying systems

Given a continuous function on a topological space we expressed its geometrical and topological properties in terms of critical homological values of the function. In our static model a piece of music was represented by a single persistence diagram. Here, the idea is to study how the *Tonnetz* evolves when its vertices are updated by successive notes.

12.1.1 State of the art

The theory of persistent homology has been generalised to the study of time varying systems in (Cohen-Steiner et al., 2006). Intuitively, given a time-dependent continuous function $f : \mathbb{X} \times [0, 1] \to \mathbb{R}$, it is possible to represent its evolution in time as a multiset of continuous paths in the space of persistence diagrams.

In Sections 7.2 and 8.1 we described the algorithm to build the filtration of a simplicial complex given a function defined on its vertices. Once the filtration is built and the pairing between critical simplices defined, it is possible to compute the boundary matrix B (see Equation (7.2.1)), and its decomposition $B = RU$. If the function varies in time, this variation will be translated in a change of the ordering of the simplices in the matrix B. The swap of the ith and the $(i + 1)$th simplices is expressed by the product PBP, where P is the permutation matrix swapping the ith and $(i + 1)$th rows and columns of B. To update the pairing of critical simplices and the persistence diagram, it suffices to recompute the RU-decomposition of the matrix. In particular, this result can be achieved in linear time in the number of simplices.

This procedure can be interpreted in terms of a continuous function defined on a topological space, by considering the evolution of its critical homological values in

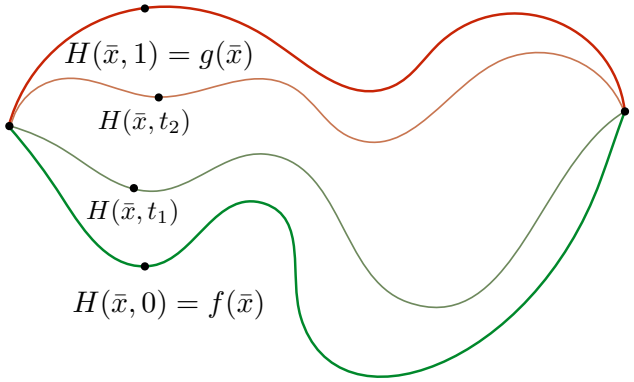


Figure 12.1: Homotopy between the functions $f, g : \mathbb{X} \to \mathbb{Y}$.The values $t \in [0, 1]$ can be interpreted as time, thus $H(x, t)$ describes the continuous deformation allowing to transform f in g.

time.

Definition 12.1.1. Let $\mathbb{X}$ and $\mathbb{Y}$ two topological spaces and $I = [0, 1] \in \mathbb{R}$. Consider two continuous functions $f, g : \mathbb{X} \to \mathbb{Y}$. The maps f and g are said *homotopic* $f \simeq g$, if there exists a continuous function $H : \mathbb{X} \times I \to \mathbb{Y}$ called a *homotopy*, such that $F(x, 0) = f(x)$ and $F(x, 1) = g(x)$, for every $x \in \mathbb{X}$.

Now, consider two tame functions $f, g : \mathbb{X} \to \mathbb{R}$. Let $H : \mathbb{X} \times I \to \mathbb{R}$ a homotopy between f and g. Assuming that $F(x, t)$ is tame for every $t \in [0, 1]$, we have a persistence diagram of dimension k, for every couple $(t, k) \in I \times \mathbb{Z}$. Let $\bar{\mathbb{R}}^3$ be the extended Euclidean space including the points at infinity. The time-dependent collection of k-persistence diagrams is called a *k-dimensional vineyard*. The 3-dimensional trajectory associated to each corner-point is called a *vine*. The different kind of paths described by the vines can be divided in three different classes. A vine is said

1. *open* if it is represented by a path composed by proper corner points for every $t \in I$;

2. *closed* if it starts and ends at an off-diagonal point;

3. *half-open* or *half-closed* if it starts (ends) on the diagonal plane and it ends (starts) at an off-diagonal point.

In Figure 12.2 on the facing page two open vines are represented as solid paths, a half-open and a half-closed vine are depicted as dashed trajectories, while the dotted path corresponds to a closed vine. If the homotopy is smooth, the vine is also smooth, the only exceptions are represented by the points in which the pairing of homology critical values change. These points are called *knees* points and come in pairs. We refer to (Cohen-Steiner et al., 2006) for the discussion of the possible configurations of vines, that shall not be used in this chapter.

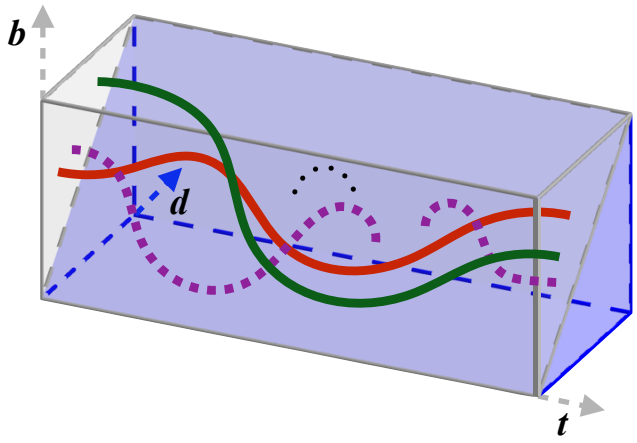


Figure 12.2: An example of vineyard. The axes represent the time t, the birth level b and the death level d of the k-homology classes of a system evolving in time. Vines are represented as continuous paths.

In general, a vineyard has a complicated geometrical structure. Statistics has been introduced in this theory in order to provide a unique mean diagram associated to the vineyard (Mileyko et al., 2011; Munch et al., 2012; Munch, 2013).

12.2 Dissimilarity of persistence time-series

Albeit vineyards represent a powerful tool, for the description of the time-varying systems, their interpretation is not intuitive. In addition, there is not a general technique that allows to compare vineyards deduced from the evolution of two different topological spaces. Furthermore, the comparison of two mean diagrams cannot provide a description of local changes in the evolution of the space that can be relevant.

Let $\mathbb{X}$ be a topological space, $f : \mathbb{X} \times [0, T] \to \mathbb{R}$ a piecewise linear function and $\mathbf{t} = \{ t_0, \dots, t_n \}$ a partition of $(n+1)$ evenly spaced points of $[0, T] \subset \mathbb{R}$. A k-persistence diagram $D_{f_i,k}$ is associated to each instant t_i. The collection of these *k-persistence snapshots* is a time series $\mathcal{D}_{f,n} = \{ D_{f_i,k} \}_{i=0}^{n} \subset D_\infty$. We name $\mathcal{D}_{f,n}$ a *k-persistence time series*.

12.2.1 Dynamic Time Warping algorithm for persistence time series

Let $\mathbf{t}_n = \{ t_1, \dots, t_n \}$ and $\mathbf{t}_m = \{ t_1, \dots, t_m \}$ two evenly spaced partition of $[0, T_f]$ and $[0, T_g]$, respectively, where $T_f, T_g \in \mathbb{R}$ and $m, n \in \mathbb{N}$. Let

$$f : \mathcal{T} \times [0, T_1] \to \mathbb{R}$$
$$g : \mathcal{T} \times [0, T_2] \to \mathbb{R}$$

be two functions, such that f_{t_i} and g_{t_j} are tame for every $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$, respectively. Consider the two persistence time series $\mathcal{D}_{f,n} = \{ D_{f_i,k} \}_{i=0}^{n}$ and $\mathcal{D}_{g,k} = \{ D_{g_i,k} \}_{i=0}^{m}$ associated to the evolution of the two functions.

There exists several methods to evaluate the dissimilarity of two time series in a metric space, in particular Dynamic Time Warping has already been used in this work in Section 3.7.1 to obtain the dissimilarity scores of multivariate time series and in Section 11.1.1 to compute the pairwise alignment of symbolic sequences. By definition, the bottleneck distance between k-persistence diagrams $d_{B,k} : D_k \times D_k \to \mathbb{R}$ satisfies the three properties that characterise a cost function for every $k \in \mathbb{Z}$. Let $x, y \in D_k$, then

- $d_{B,k}(x, y) \geqslant 0$ for every x, y and for every $k \in \mathbb{Z}$;
- $d_{B,k}(x, y) = 0$ if and only if $x = y$;
- $d_{B,k}(x, y) = d_B(y, x)$ for every $x, y \in D_k$, for every $k \in \mathbb{Z}$.

Let $\mathcal{D}_f$ and $\mathcal{D}_g$ be two time series of k-persistence diagrams of length $n, m \in \mathbb{N}$ respectively and α and β two natural numbers such that $1 \leqslant \alpha \leqslant n$ and $1 \leqslant \beta \leqslant m$. Following the notation introduced in Section 3.7.1, the DTW between two sequences of k-persistence diagrams is given by the computation of the optimal warping path γ^*:

$$\begin{aligned} DTW\left(\mathcal{D}_f, \mathcal{D}_g\right) &= d_{B_{\gamma^*}}\left(\mathcal{D}_f, \mathcal{D}_g\right) \\ &= \min\left\{ d_{B_\gamma}\left(\mathcal{D}_f, \mathcal{D}_g\right) \mid \gamma \text{ is an } (n,m) - \text{ warping path} \right\}. \end{aligned}$$

In particular, the DTW inherits the symmetry by the bottleneck distance and the tameness of f and g assures the bottleneck stability of every diagram D_{f_i} and D_{g_j}, with $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$.

Consider the prefix sequences $\mathcal{D}_{f,\alpha} = \{ D_{f_i,k} \}_{i=0}^{\alpha}$ and symmetrically $\mathcal{D}_{g,\beta} = \{ D_{g_i,k} \}_{i=0}^{\beta}$. For simplicity, the index k specifying the dimension of the persistence diagram shall be omitted if the context is clear. Let

$$A\left(\alpha, \beta\right) := DTW\left(D_{f,\alpha}, D_{g,\beta}\right),$$

be the entry of the *accumulated cost matrix*, then $A\left(n, m\right) = DTW(\mathcal{D}_f, \mathcal{D}_g)$.

Theorem 12.2.1. *Let A be the accumulated cost matrix. The identities*

1. $A\left(\alpha, 1\right) = \sum_{l=1}^{\alpha} d_B\left(Df_l, Dg_1\right)$, *for* $1 \leqslant \alpha \leqslant n$;
2. $A\left(1, \beta\right) = \sum_{l=1}^{\beta} d_B\left(Df_1, Dg_l\right)$, *for* $1 \leqslant \beta \leqslant m$;
3. $A\left(\alpha, \beta\right) = \min\left\{ A\left(\alpha-1, \beta-1\right), A\left(\alpha-1, \beta\right), A\left(\alpha, \beta-1\right) \right\} + d_B\left(Df_\alpha, Dg_\beta\right)$, *for* $1 < \alpha \leqslant n$ *and* $1 < \beta \leqslant m$.

hold. Moreover, the number of operations required for the computation of $DTW\left(D_{f,n}, D_{g,m}\right)$ is $O(nm)$.

The theorem holds for general cost functions and it is proved in (Senin, 2008). The optimal warping path with respect to the accumulated cost matrix is computed through the Algorithm 12.1.

Algorithm 12.1 Optimal warping path.

Input:
A ▷ Accumulated cost matrix.
Output:
$\gamma^* = \{ \gamma_1^*, \ldots, \gamma_l^* \}$ ▷ Optimal warping path.

1: $\gamma_l^* = (n, m)$ and $\gamma_1^* = (1, 1)$;
2: **while** $l > 1$ **do**
3: $$\gamma_{l-1}^* = \begin{cases} (1, \beta - 1), \text{if } \alpha = 1 \\ (\alpha - 1, 1), \text{if } \beta = 1 \\ \min \{ A(\alpha - 1, \beta - 1), A(\alpha - 1, \beta), A(\alpha, \beta - 1) \}, \text{otherwise} \end{cases}.$$
4: **end while**

12.3 Applications

In the following applications we use DTW to compute the dissimilarity between 0 and 1-persistence time series associated to three datasets composed by classical, pop and jazz compositions, respectively. Often, musical phrases are organised respect to the metric of the piece: modulations occur each four or eight bars in a jazz context, as the melodic line of the voice is arranged in a question and answer paradigm consisting of cycles of 2 or 4 bars in pop music. Thus, reflecting the approach followed when performing signal analysis, it is reasonable to space observations in time by taking into account the subdivision of each piece in bars. Therefore, it is also reasonable to study the properties of these features when the windowing is varied.

12.3.1 Musical interpretation

First of all, it is necessary to provide an interpretation of the music features represented by the persistence time series. In Figure 12.3 on the next page, a sequence of six 0-persistence diagrams computed considering a 8-bar windowing of *Klavierstück I* is depicted. We recall that the 0th persistent module describes the connectedness of the torus F, when it is rebuilt *via* the filtration induced by the height function on $\mathcal{T}$. First of all, note that the axes of the persistence diagrams in the figure have different limits. Consider the top-left persistence diagram, the two corner points represent the lifespan of two connected components. The first one is a cornerpoint at infinity. It reveals the connected nature of F and represents the subcomplex of minimal height retrieved by the height function. The proper cornerpoint, as we mentioned in Section 8.2 points out the presence of a minimum of the height function, which is disconnected from the first one. The remainder of the observations describes the changes in terms of death and birth-levels of such minima. In particular, the connected components correspond to disconnected subcomplexes of the fundamental domain of the *Tonnetz*. The chromatic and atonal nature of the piece is suggested by the persistence of these minima. Moreover, the increasing growth of the birth-levels of the points of the whole multiset grabs the homogeneous gain of *height* of the whole simplicial complex $\mathcal{T}$. This means that the whole chromatic scale is utilised

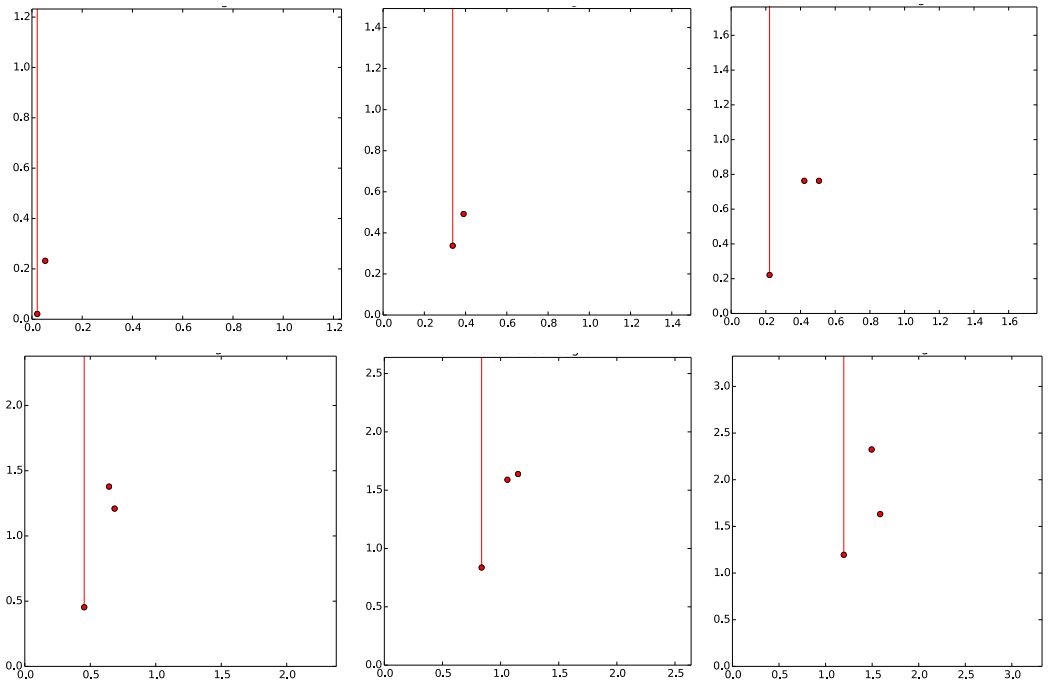


Figure 12.3: The six first observation of the 0-persistence time series. *Klavierstück I* - Schönberg. Persistence snapshots are taken each 8 bars.

in the composition in a uniform way, both in terms of pitches and duration of the notes. The relative distances among corner points represent their disparity when represented as subcomplex of the deformed *Tonnetz* $\mathcal{T}$. The variability of their configuration highlights the different *preferred directions* followed by the piece: disconnected regions of the *Tonnetz* at different heights represent pitch class or pitch-class sets that we have listened to in inverse proportion to their birth-level in the filtration.

In Figure 12.4 the persistence time series associated to the same composition, but composed by 1-persistence diagrams is shown. We recall, that the two cornerpoints at infinity correspond to the two generators of F consisting of major and minor third intervals, respectively. In the first observations the two cornerpoints at infinity are well separated and a third maximum disconnected from the others give rise to a proper cornerpoint. This homology class vanishes at the second observation suggesting a progressive compression of the whole set of pitch classes. The same idea is highlighted by the progressive reduction of the distance between the two cornerpoints at infinity, which are fused in a single point of multiplicity 2 in the third observation.

We are now ready to align two k-persistence time series and compute their dissimilarity and their optimal warping path, using the DTW algorithm.

12.3.2 Optimal persistence warping path

Given a set of k-persistence time series, the calculation of the pairwise bottleneck distance, which normally represents a computationally hard task, can be performed

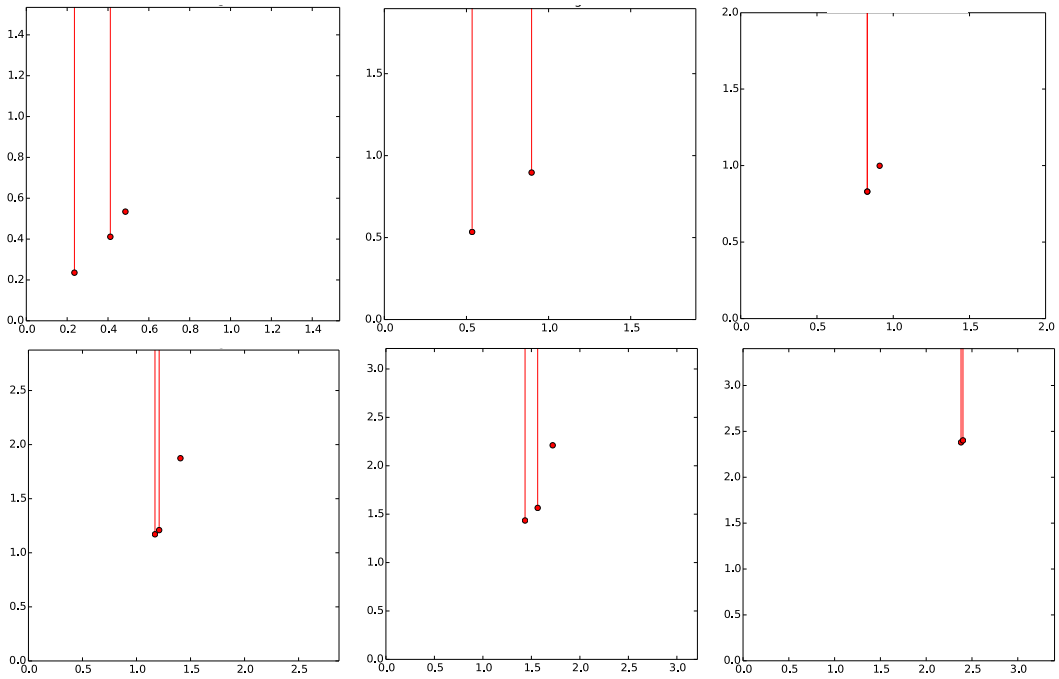
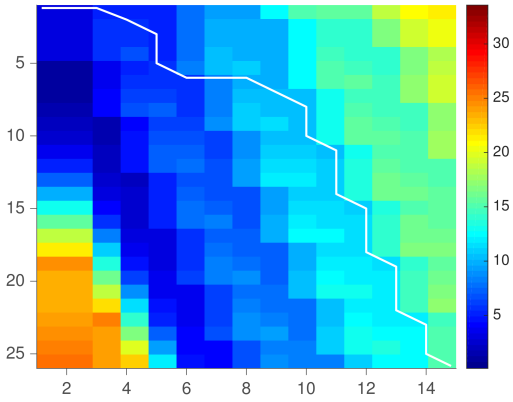


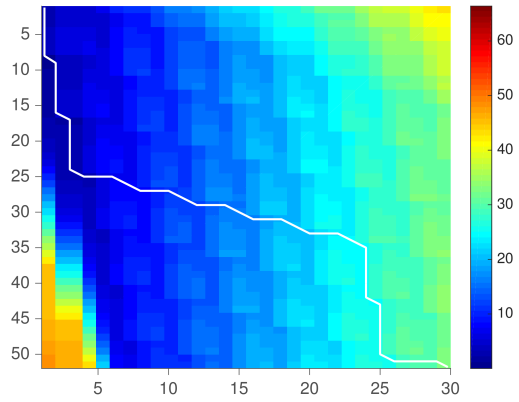
Figure 12.4: Consecutive observations of a 1-persistence time series. *Klavierstück I* - Schönberg. Persistence snapshots taken at constant relative time intervals of 8 bars.

in a reasonable amount of time, due to the low dimensionality and simple structure of F. DTW allows to compute the pairwise optimal warping path between two compositions. In Figure 12.5 on the following page two examples are depicted. The images in the first row of the figure describe the optimal warping path between the third movement of the *Sonata n.* 8 by Mozart and *Jeux d'Eau* by Ravel, for an 8 and a 4-bars windowing, respectively. The second row represents the alignment of two pop songs, namely *Genie in a Bottle* by Christina Aguilera and *Fortress around Your Heart* by Sting. Both pieces have been aligned using a 8 and a 4-bars windowing.

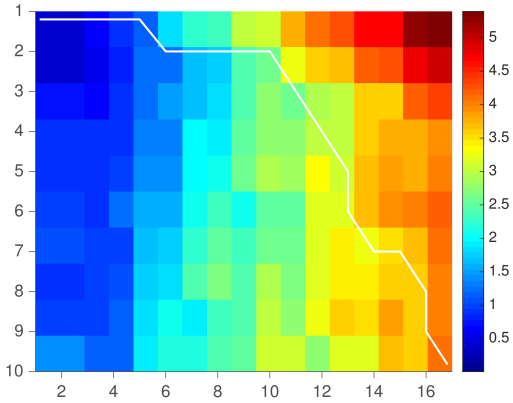
According to Theorem 12.2.1, the first point of the warping path is assumed to be the $(1, 1)$ entry of the accumulated cost matrix. This assumption corresponds to force the alignment of the first w bars of the two pieces, where $w \in \mathbb{N}$ is the size of the windows we consider. Horizontal and vertical segments of the piecewise linear path along the matrix correspond to the insertion of gaps while aligning two symbolic sequences. Thus, in musical terms, the optimal warping path represents comparable regions of the two compositions represented by similar (near with respect to the bottleneck distance) persistence diagrams. In Figure 12.6 two persistence time series associated to the compositions A and B are represented by piecewise line segments and their observations are labelled according to a 4-bars windowing. The dashed lines represent the alignment of the compositions described by the optimal warping path. The first twelve measures of A are associated to the first four bars of B. Assuming to consider an accumulated cost matrix whose columns represent the observations of A and rows the observations of B, the warping path would connect $A(1, 1)$ to $A(1, 3)$ and it is a horizontal line segment. Symmetrically, the last eight



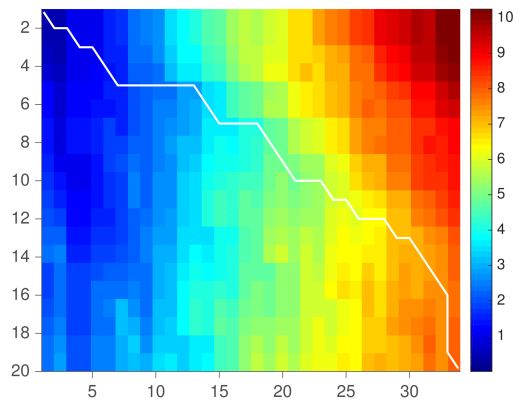
(a) Sonata n. 8 - 3, Mozart vs. Jeux d'Eau - Ravel (8 bars windowing).



(b) Sonata n. 8 - 3, Mozart vs. Jeux d'Eau - Ravel (4 bars windowing).



(c) Genie in a Bottle, Aguilera vs Fortress of Your Heart, Sting (8 bars windowing).



(d) Genie in a Bottle, Aguilera vs Fortress of Your Heart, Sting (4 bars windowing).

Figure 12.5: Accumulated cost matrices and optimal warping paths between 0-persistence time series.

bars of A are associated to the four last measures of B. In this region the warping path would be represented as a vertical line segment.

Following the musical interpretation of the 0-persistence diagrams we gave in Section 12.3.1 and the results discussed in Section 8.2, the optimal warping path suggests local regions of the two pieces in which the minima of the height function evaluated on the two deformed *Tonnetze* are organised in a similar way. Symmetrically the optimal warping path between 1-persistence time series highlights regions in which the 1-dimensional holes defined by the height function, have similar configurations in terms of height (birth-level) and connectedness (number and relevance of the cornerpoints), with respect to the structure of the *Tonnetz*. Hence, a timespan in which the compositional choices characterising the pieces are comparable in terms of their *persistent* representation.

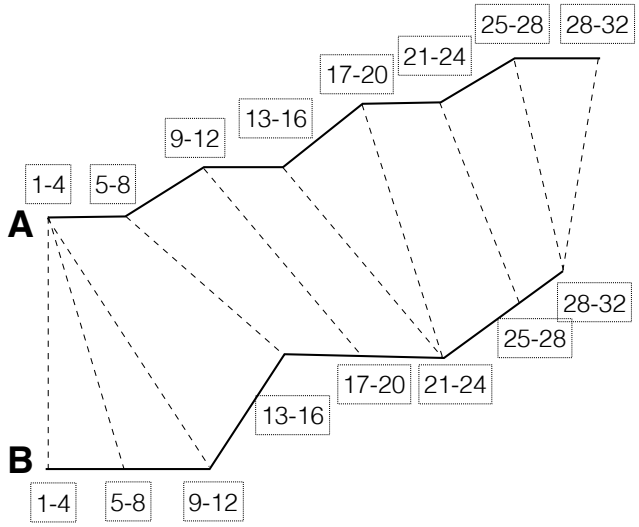


Figure 12.6: Dynamic time warping between persistence time-series associated to two compositions A and B. Observations are labelled according to a 4-bars windowing.

Composition	*Movements*	*Author*
Sonata n. 27	1,2,3	Beethoven
Arabesque		Debussy
Sonata n. 8	1,2	Mozart
Jeux d'Eau		Ravel
Klavierstük	I,II	Schönberg

Table 12.1: Summary of the compositions of the classical music dataset.

12.3.3 Dissimilarity of persistence time series

Let $\mathcal{D}_f = \{ D_{f_i,k} \}_{i=0}^{n}$ and $\mathcal{D}_g = \{ D_{g_i,k} \}_{i=0}^{m}$ be two persistence time series of k-persistence diagram, for $k \in \mathbb{Z}$. The computation of

$$DTW(\mathcal{D}_1, \mathcal{D}_2) = A(n, m)$$

allows to retrieve the dissimilarity between the two time series.

This value provides a measure of the effort needed to produce the minimal cost elastic transformation described by the optimal warping path, in order to warp a composition into the other.

In Figure 12.8 on page 200 the dissimilarity score computed by aligning the compositions belonging to three datasets are depicted. Each row of the image consists of two distance matrices, derived from an 8 (on the left) and 4-bars (on the right) windowing, respectively.

Classical alignments

Observe the first row of the figure. The pieces of the dataset are listed in Table 12.1. Proceed reading the matrix from top to bottom. Both Schönberg's compositions (we will denote them by DK11-1 and DK11-2) gave high dissimilarity score when

Label	*Ensemble*	*Style*
Caravan-js	4 Gtrs, Org, Kora, Bgtr	B.B. arr., no solo
Caravan-md	Tpt, Pf, Bgtr	Rich solos and Tensions
Fly-bz	Bgtr, Vib, Kora, Pf	B.B. arr., chromatic solos
Fly-dc	Flt, Tnr & Bar Sax, F Hn, Org, Gtr	B.B. arr., Manouche guitar
Fly-gw	Big Band	B. B. arr.
How-gr	2 Obss, 2 Gtrs, Bgtr	Manouche
How-jh	Pf, Bgtr	Chromatic solo
How-mw	Tpt, Gtr, Bgtr, Str, Pf	Embellishments, chromatic

Table 12.2: Summary of the compositions of the jazz dataset.

aligned with the tonal pieces. Consider the results of the alignment of the whole dataset with DK11-2 (first row of the matrix). The two minimal dissimilarity scores we retrieved are obtained comparing DK11-2 with the compositions by Debussy and Ravel. DK11-1 obtained smaller dissimilarity scores, however they are sufficient to segregate it from the tonal pieces we considered. The corresponding results depicted in the distance matrix on the left do not differ greatly from the one we just discussed. However the tonal traces left in DK11-1 are highlighted by the finer windowing we considered. The same consideration holds focusing on the scores realised by *Jeux d'Eau*. The surprisingly low score generated by its alignment with the second movement of the Beethoven's sonata changes considering a 4-bars windowing. In this case the composition by Ravel results segregated form the other ones we considered, while the 2nd movements of the sonata n. 27 obtain a surprisingly low dissimilarity score, when it is aligned with DK11-1. The tonal and pentatonic compositions are highlighted as similar in both representations.

Pop alignments

The dissimilarity scores computed on the dataset composed by 2 songs by Christina Aguilera and 3 pieces by Paul McCartney and Sting respectively confirm the results we obtained by the analysis of the same dataset in Section 8.2. In both diagrams the two Aguilera's pieces result well separated from the others. Sting's *Fields of Gold* and *If You Love Somebody Set Them Free* turn out to be similar to the pieces by McCartney. It is not the case for *Fortress Around Your Heart* that recollects high dissimilarity scores when aligned to the other songs of the dataset. It is interesting to notice how the two distance matrices are almost invariant respect to the change of windowing.

Jazz alignments

The classification of jazz standards is a difficult task due to the improvisational nature of this style. We considered a dataset composed by two versions of *Caravan* and three versions of *Fly Me to the Moon* and *How High the Moon*, respectively. Each interpretation is characterised by different choices in terms of ensemble and arrangements. We summarised these features in Table 12.2 denoting a big band

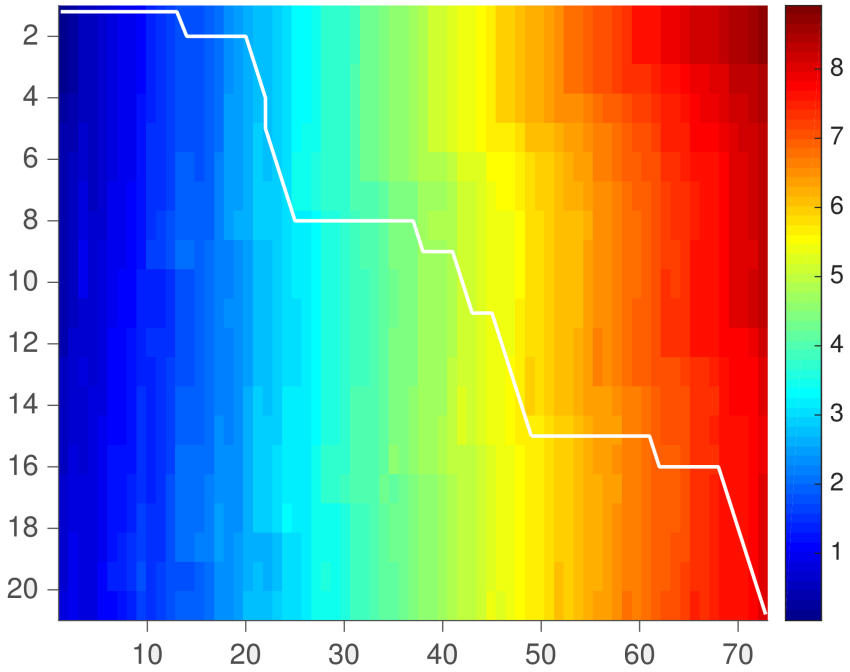


Figure 12.7: Optimal warping path between to versions of *Caravan*. The positions of the gaps correspond to the solo parts of the longer version (frames 25-50 and 51-65 respectively).

arrangement (breaks, horns fills, et cetera) as B.B. arr., pointing out the presence of solo parts, their main features, or particular stylistic choices.

The dissimilarity scores resulting by the global pairwise alignment of the persistence time series associated to these compositions are depicted in the third row of Figure 12.8. In this example, the information retrieved by the alignment is twofold: on one hand it stresses the mere melodic and harmonic similarity. On the other hand, it retrieves common stylistic choices. In both distance matrices the scores associated to the same compositions are reasonably low, highlighting their similarity in the case of *Fly Me to the Moon* and *How High the Moon*. An exception is represented by the two versions of *Caravan*. The presence of rich solos in *Caravan-md* distinguishes it neatly by the other interpretation of the standard. Notice how the optimal warping path between these two pieces depicted in Figure 12.7 tries to align them on the themes, skipping the solo parts. Hence, the evolution in time of the persistence diagrams grasps the difference between an organised thematic flow, and a freer improvisational context. Moreover, we notice how the three versions of *Fly Me to the Moon* result well separated from the three versions of *How High the Moon* only utilising a 4-bars windowing. This feature is opposite to the one characterising the analysis of the Pop dataset, stressing the different dynamics of the two styles.

12.4 Discussion and perspectives

We presented a method to adapt persistent homology to the time-dependent nature of music. We introduced the concept of persistent snapshots and presented the

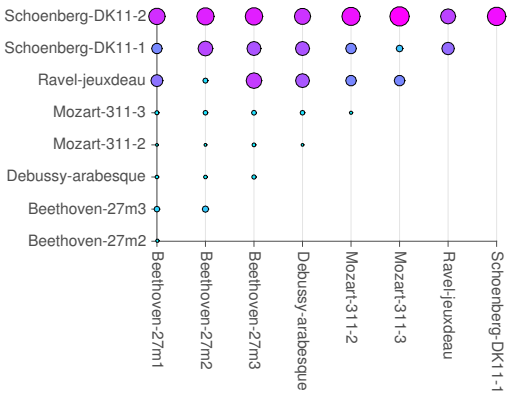
state-of-the-art methods concerning time-varying system and the computation of the evolution of persistence properties of a shape in time. The nature of the model proposed in Part III and of the functions encoding the information concerning the pitch classes and the durations of notes allowed us to consider time series, whose observations provide a persistence characterisation of collections of frames of a composition, obtained considering its subdivision into bars. We gave a musical interpretation of the evolution in time of the persistence diagrams associated to a composition and we utilised DTW to provide an alignment of persistence time series. Finally, we analyse both the optimal warping path and the alignment score of collections of classical (tonal, modal and atonal) compositions, pop songs endowed with different harmonic complexity and a collection of jazz standards played by different ensembles, with different arrangements and solo parts. The computation of the pairwise alignment scores for each dataset, revealed how in a classical music context the persistence time series classify tonal, modal and atonal compositions. The stability respect to a change of windowing of the pop compositions and hence the possibility to study them with a coarser distribution of observations has been highlighted, as the retrieval of the peculiar harmonic choices made in *Fortress Around Your Heart*. The analysis of jazz standards is more complex due to their variability in terms of improvisational styles, the generous utilisation of harmonic substitutions and the highly variable composition of the ensemble. Nevertheless, the tool we proposed is able to retrieve the similarity of different versions of the same standard, as to distinguish between the ordered structure of the theme, in opposition to the more entropic solo parts.

The natural development of this work is to extend it to the analysis of audio. The stability of the persistence diagrams assures that small variations of the function will be represented as small variations of the persistence diagrams forming the persistence time series. A chromagram can also be used to produce dynamic deformations of the *Tonnetz*, as the consonance function could be used to describe the variations in terms of tension/resolution of the degrees of the chromatic scale in relationship to a variable harmonic choice.

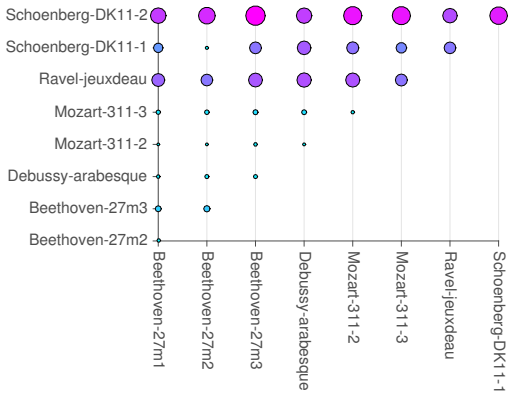
The variation of the *Tonnetz* in time can be used to generate music. The study of free or constrained trajectories of a mass on the time-dependent deformed surface induced by a composition can be used to create a new melody, according to the *preferred directions* defined by the deformation. Different melodies computed considering the nearest pitch class or pitch-class set to the mass can be classified in terms of periodicity and symmetry. The same ideas can be extended to a system of n masses moving on the surface. Some interesting starting points can be borrowed by the theory of configuration and reconfiguration spaces (Abrams and Ghrist, 2002; Ghrist and Peterson, 2007).

Persistence time series could be substituted by continuous vineyards, which are suitable for representing the variations induced by piecewise constant and piecewise linear functions. It would be interesting to study the alignment between continuous representation of the variation in time of persistence diagrams considering the minimal homotopy leading from a vineyards to another one. The musical framework we introduced is particularly suitable for this task, given the simple structure of the persistence diagrams derived from the *Tonnetz* and the possibility to provide their musical interpretation.

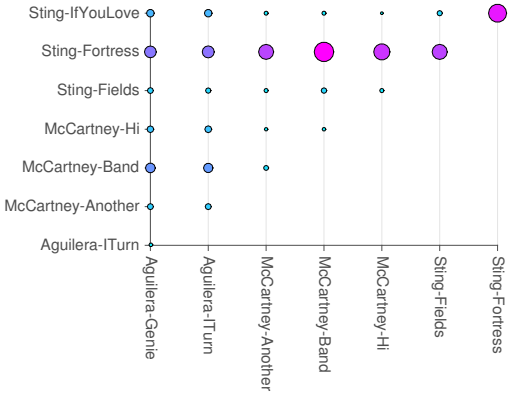
The persistence time series we introduced in this last chapter have full memory with respect to the notes that are played in time. It would be possible to introduce in the model a *gravity* function, in order to represent the *plasticity* of the listener's perception. In a pitch-duration-based model this assumption would enhance the representation of repeated musical ideas, while in a consonance-oriented model this function would reflect the decreasing of the tensional content of a long lasting or repeated harmonic/melodic choice.



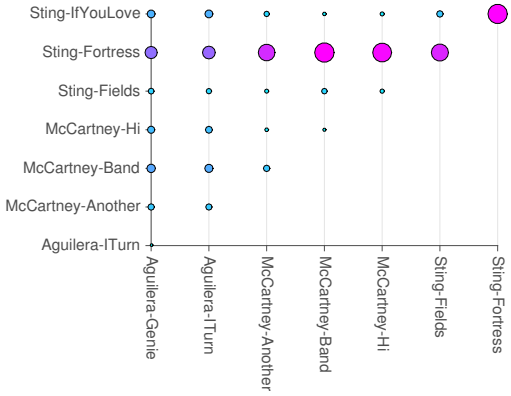
(a) Classic Music (8 bars windowing).



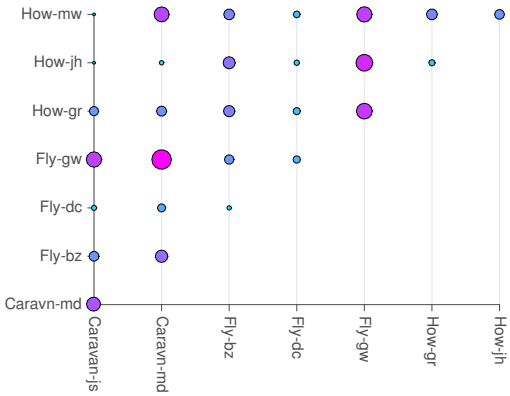
(b) Classic Music (4 bars windowing).



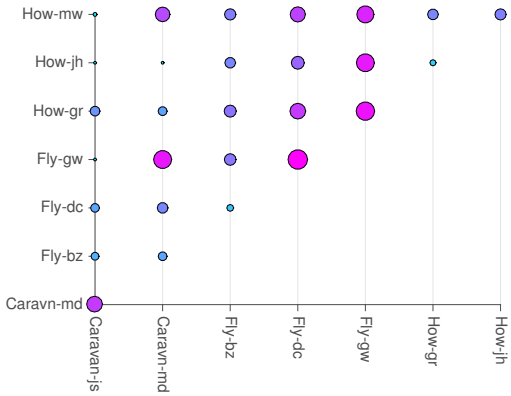
(c) Pop Music (4 bars windowing).



(d) Pop Music (2 bars windowing).



(e) Jazz Music (8 bars windowing).



(f) Jazz Music (4 bars windowing).

Figure 12.8: Alignment score of 0-persistence time series for different datasets and variable windowing. The colour and the size of the circles associated to each pair of pieces depends on their alignment score.

Part V

Conclusion and future works

Thirteen

Conclusion

The question that inspired the investigations portrayed in this work is simple: why some melodies create interesting musical soundscapes even when they are simply whistled, while others need an orchestra to be easy to understand?

To answer this question, we represented music by using topological and geometrical models, following the twofold interpretation suggested by (Kurth and Rothfarb, 1991). We tried to keep the dimensionality of our representation as low as possible, in order to guarantee a visual intuition over the entities that we described mathematically. Furthermore, the metric that is naturally defined on the spaces we considered allows to define a distance between musical objects in a natural way.

We used collections of partial permutation matrices and three dimensional paths to describe voice leadings and counterpoint. Our model surely needs improvements in order to reflect the vast information contained in a whole contrapuntal composition, and also to describe the numerous techniques that are used by composers to generate interest in the listener. As we stated in the introduction, despite some attempts have been made (Birkhoff, 1933) and it is an open line of research (Juslin and Västfjäll, 2008; Tulipano and Bergomi, 2015; Brattico and Pearce, 2013), it is not yet possible to evaluate the aesthetics of music objectively. Thus, it is not possible to speak of a truly mathematical complexity, unless considering a sort of *understandability/originality* dilemma: the action of the composer on a piece, in order to shape its resolution and tension structure in time, in order to accompany or frustrate the expectation of the listener. From this consideration follows the necessity of producing time-dependent models and study their patterns. We hope that, beyond the limitations of the model we suggest, its formal core and novel outline shall give a new perspective on the study and representation of counterpoint.

Second, we encoded part of the information contained in a musical phrase by displacing the vertices of the well-known Euler *Tonnetz*, in its simplicial complex interpretation. The assumption, in this case, does not concern the concatenation in time of motifs endowed with nuanced complexities, but the hypothesis that a core musical idea should be repeated, although slightly transformed, along a composition. In this model, a piece of music is represented as a three dimensional shape. Persistent homology, computed on the deformed surfaces derived from the *Tonnetz*, is able to grasp the main ideas used in a composition and to encode them in a simple representation. The main advantages of this approach are twofold. On one side, the topological tools we used can deal with any finite number of dimensions and have been proved to be effective in the retrieval of different musical properties. On the other side, persistence diagrams are points of a metric space. Hence, it is possible to compare them, even though the computational complexity of this distance requires

a large amount of time (interesting researches are currently investigating novel algorithms for the computation of this distance (Di Fabio and Ferri, 2015)).

When questioning the possible representations of music, it also natural to wonder if it is describable through absolute models. In this work, we do not provide a ultimate answer to this question. However, we assumed that the perception of music depends on the culture and the background of the listener. According to this observation, we introduced models whose main ingredients are the *Tonnetz* and the consonance function (Plomp and Levelt, 1965). This coupling of a musicological model and a function deduced from the fitting of experimental data, allows to consider both the symbolic structure of music and the information carried by the signal. Our consonance-based shapes are limited compared to the abstraction level provided by the standard *Tonnetz*. Nevertheless, they reflect the perceptive nature of music, providing interesting and coherent results.

Finally, the two last models we suggested are a consequence of the exploration of both the horizontal and vertical approaches. They are not to be considered as improvements of the previously investigated strategies. On the contrary, they are endowed with their own independence, and offer a novel viewpoint on the complementarity of low and high-level features of music. Moreover they take into account the dynamical nature of real-life musical applications.

Music proved to be a rich source of inspiration for the development of mathematical tools, providing a suitable framework for the novel time-series approach to the topological characterisation of time-varying systems.

If the primary goal of this research was to provide a complete formalisation of the compositional process, we are surely far from giving a broad representation of all its features. However, we hope that this work shall represent a new starting point for future researches in music analysis, music information retrieval and computational algebraic topology.

Fourteen

Future works

The results portrayed in this work can be divided in three main groups, according to the three directions we explored. The modelling and the visualisation of voice leadings, the topological description of music features and, finally, the time-oriented analysis of musical entities. Following this structure, we give a brief summary of the results discussed at the end of each chapter. Then each subsection provides a more detailed overview on their future developments.

14.1 Voice-leading modelling

14.1.1 Voice leadings as partial permutations and geodesics

The formalisation of simultaneous motions of voices provides a handy representation of a voice leading as a partial permutation matrix. The representation of voice leadings as geodesic paths in several spaces allows to simply understand their representation as a *concatenation of geodesics* and the different information retrieved by the standard spaces of music analysis.

The information carried by each partial permutation matrix has been rewritten as a five-dimensional vector, in order to represent rested voices. Thus, given a contrapuntal composition it is possible to compute its paradigmatic voice leadings and to represent them as a multiset of 5-dimensional points. Furthermore, the interpretation of the sequence of vectors as observations of a time series allows to describe the evolution of voices' motions in time. Thus, to define a dissimilarity measure describing two compositions, it is possible to apply the powerful techniques used for the computation of the *distance* between time series. Finally, a particular class of partial singular braids is used to visualise the voice leadings between chords represented as pitches and pitch classes.

Test and evaluation

The algorithm for the comparison of time series of complexity vectors has been tested on a small set of compositions. The evaluation of its accuracy in relation to tasks such as the artist retrieval and both the stylistic and temporal classification should be computed on large datasets. The same holds for the visualisation of the paradigmatic voice leading choices as a multiset of points. This representation should be evaluated on a large collection of compositions by a musicologist, which could provide a meaningful interpretation of the 3-dimensional projections of the point cloud.

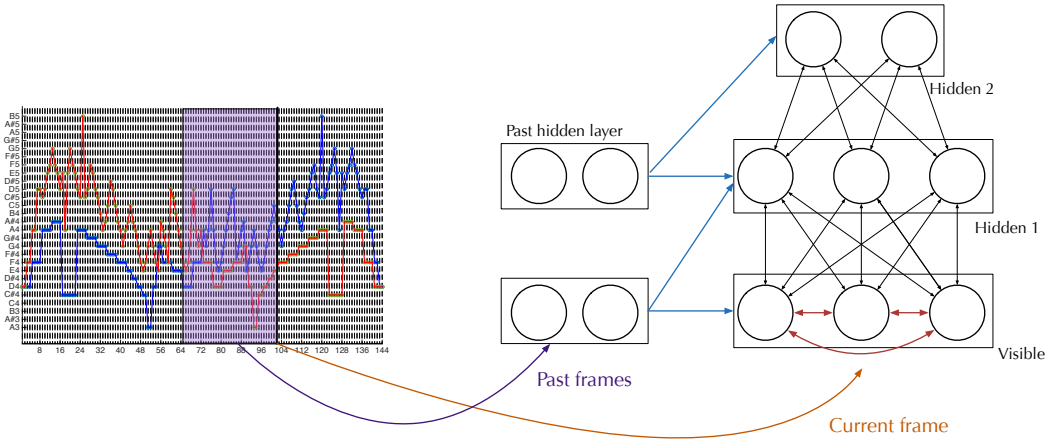


Figure 14.1: The partial permutation matrices give a low-dimensional representation of the features of each voice leading. Here, they are used to feed a harmonic conditional restricted Boltzmann machine. The lateral connections in the visible layer are used to retrieve the harmonic structure of chords. Past events are taken into account thanks to the autoregressive connections between the *current* and *past* units.

Higher order phenomena

The complexity vector we introduced takes into account only the features of a single voice leading. This analysis is extremely localised, and it is blind with respect to phenomena occurring in the concatenation of several voice leadings. For instance, the overlap of two voices (described in Section 1.2) is visible only by considering more than one voice leading at a time. In addition, the behaviour of each voice can be tracked by analysing its evolution in the concatenation of partial permutation matrices. This approach allows to measure the length of the crossings between voices and hence, a more accurate computation of the overall complexity of a polyphonic composition.

A deep learning model for orchestration

The analysis of the voice leading complexity can be used to implement the structure of generative models for automatic orchestration (Crestel, 2015). The deep neural network depicted in Figure 14.1, based on the models described in (Taylor and Hinton, 2009; Osindero and Hinton, 2008) and implemented by Léopold Crestel aims at the generation of real-time orchestrations of symbolic data (onset, pitch, duration, velocity). The network is trained by comparing piano and orchestral arrangements of a set of compositions. The number of instruments composing an orchestra (15 at the actual state of development of the model) give rise to high-dimensional, sparse representations, that can be simplified considering the representation of voice leadings through partial permutations. Furthermore, we suggested a simplistic extension of the partial permutation model in order to classify contrapuntal composition of other species than the first. In this context, the discretisation used to compute the complexity of the voice leadings shall be improved by the *(noteon/noteoff)*

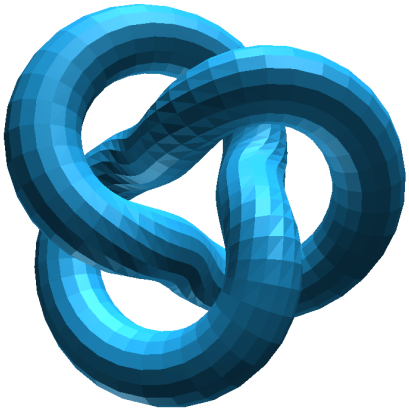


Figure 14.2: Trefoil knot. Identifying the domain and co-domain of a braid $b \in B_n$ produces a closed braid. In particular, any knot can be represented as a closed braid (Alexander, 1923).

information provided by the *MIDI* files.

14.1.2 Voice leadings and braids

A concatenation of voice leadings between n-notes chords can be represented as a piecewise geodesic path in $\mathbb{R}^n$ (ordered pitches), $\mathbb{T}^n$ (ordered pitch-classes), or $\mathbb{A}_n$ (pitch-class multisets). When $n > 3$ this path cannot be easily visualised. Braids allow to depict a voice leading between n-notes chords as a collection of n paths in $\mathbb{R}^3$. Furthermore, unisons and rests can be represented by intersections (singularity) and deletion (partiality) of the strands, respectively. In order to univocally associate a voice leading to a collection of three dimensional paths, we considered the class of piecewise, positive, partial, singular braids.

Voice leading's topological qualities

This visualisation strategy can be transformed in a representation space by taking into account the topological complexity of the braids we defined. We showed how both the motion of each voice and the intervallic leap it covers during the voice leading can be represented as the slope of the strand associated to the voice. The musical relevance of the topological invariants (Birman, 1974) of this voice-leading-oriented braids should be investigated. For instance, this problem can be tackled by considering the link generated by the closure of a braid (Alexander, 1928) (see Figure 14.2 for an example). When it is defined, it is possible to consider the closure of the multiplication (concatenation and removal of discontinuous strands) of the partial braids representing a suite of voice leadings. In our model, we assumed that every crossings between strands are positive. However it is possible to define a rule determining the sign of the crossings, for instance by taking into account the slope of the strands. In this way, the information concerning the intervallic leaps of the voices would be inherited by the structure of the link associated to a braid.

A distance between pitch and pitch-class sets

We visualised voice leadings between ordered tuples of pitches as a collection of geodesics paths in $\mathbb{R}^3$. The idea is to consider the length of the strands in order to define a distance between two particular ordered representative of a set of pitches. In particular, minimal geodesics represent a non-crossing voice leading. The introduction of crossings implies the use of longer line segments to connect the pitches. Note that this kind of distance would be different than a mere count of the crossings, since the length of the segment (or equivalently its slope) describes the relative intervallic motion of each voice. A symmetric argument can be applied to pitch-class sets, considering the helices connecting two copies of $\mathbb{R}/12\mathbb{Z}$.

14.2 Persistent music features

We interpret the *Tonnetz* as a 2-dimensional simplicial complex. Its embedding in a metric space and its structure, in which pitch classes, consonant intervals and triads are represented as 0, 1 and 2-simplices respectively, have been used in order to take advantage of a third dimension to add relevant information to its structure.

14.2.1 Pitch classes and durations

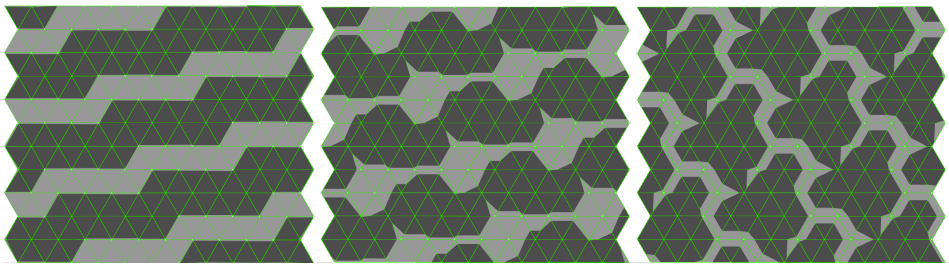
As a first application, we use the pitches and durations of a sequence of notes and chords, in order to define the height of the pitch-classes labelling the vertices of the *Tonnetz*. After introducing the theory of persistence homology, the height function defined on the vertices has been used to induce a filtration of the fundamental domain of the *Tonnetz* generated by the major and minor third intervals.

Styles classification

In spite of its simplicity, the first approach we introduced that considers a slice of the deformed *Tonnetz* in order to determine a *preferred extended shape* associated to a composition, revealed an interesting behaviour when applied to the description of different styles of classical music. For instance, the shapes retrieved from the analyses of impressionistic compositions are isomorphic. This result and the ease of the model are strong points that can lead to an intuitive representation of different compositional styles. This representation will be tested on large datasets and can be refined by introducing a variable threshold, in order to take into account the pitch classes and pitch-class set circulation (Tymoczko, 2011; Bigo, 2013).

14.2.2 A topological music fingerprint

Given a musical phrase, the computation of its persistent homology by considering the homological critical values of the height function generates two relevant persistence diagrams. The 0-persistence diagram describes the lifespan of the connected components of the shape along the filtration. The 1-persistence diagram collects the same information concerning the 1-dimensional holes of the shape. Both these representations have a neat musical interpretation. Furthermore, persistence diagrams are points of a metric space, equipped with the bottleneck distance. This notion



(a) *Time*, Hans Zimmer. (b) *Sonata n.8*, mov. 1 Mozart. (c) *Klavierstück*, I, Schönberg.

Figure 14.3: Visualisation of different compositional styles as sub-level sets of the height function (light grey area). The displacement of vertices is given by the duration and pitch classes of notes and chords.

allowed to use the 0 and 1-persistent homology representation of music, to provide a quantitative analysis of the distance between compositions. Thus, it has been possible to organise three different datasets (of classical, jazz and pop compositions respectively), according to their topological representation.

Didactic: composition

The deformed *Tonnetze* and the contours depicted by the sub-level sets of the height function highlight the differences between compositional styles. This property can be used to ease the complicated task of teaching (and learning) how to compose a piece of music. The 3-dimensional deformation of the *Tonnetz* can be used to visualise common tonal and atonal pattern, as depicted in Figure 14.3. Moreover, it can be a valuable tool for composers and students in composition, allowing them to produce an immediate visual feedback of their musical ideas and compare them to common patterns or educational examples.

Extension to audio analysis

It is possible to extract information concerning both pitch classes and duration of a sequence of notes directly from audio files (Harte and Sandler, 2005). In a chromagram, a magnitude is associated to each pitch class, representing its importance over time. The interest of this extension is twofold: on one side the stability of the persistence diagrams (computed considering tame functions) makes this representation robust to the small errors introduced by the analysis of the signal. On the other hand, the information carried by the signal is richer than the information contained in a MIDI file. The harmonic spectra of melodic and harmonic instruments influence the chromagram, but also the harmonic contribution of percussive instruments. Moreover, auditory masking phenomena (Wegel and Lane, 1924) are non-negligible. Thus, this extension would better approximate our perception of the musical information.

Validation

The results obtained by considering both the 0 and 1-dimensional persistent diagram associated to shapes generated deforming the *Tonnetz* are promising. In a classical music context tonal, modal and atonal classical pieces have been represented as separated clusters. Three different versions of a jazz standard have been grouped coherently with respect to their arrangements and pop songs belonging to different artists have been grouped according both to their *genre* and compositional style. The 1-persistence diagrams seemed to grab the similarity between melodies, but also to distinguish between the ordered structure of a theme and the more entropic solo parts. However it is necessary to test all this feature on large datasets. The extension of the model to the analysis of the audio shall be fundamental to test our model on a heterogenous and rich collection of compositions

Application to other music-oriented simplicial complexes

In this work, we considered the *Tonnetz* for its acoustical and musicological relevance. However, our constructions can be applied to every simplicial complex, graph or point cloud having a musical meaning. Clearly, one difficulty is the choice of the filtration function, as it should reflect the *musical* persistence properties one wants to consider.

Higher dimensions

In our model, we considered only three dimensions, to visualise the deformations of the *Tonnetz*. However, persistent homology is suitable for the analysis of high-dimensional data, that makes possible to study features such as the velocity, the position of a note in a bar respect to a certain metric and dynamics of a sequence of notes.

Multidimensional persistence

A further development consists in the extension of the deformed *model* to multidimensional persistence (Carlsson and Zomorodian, 2009; Cagliari et al., 2010). This theory allows to explore several filtrations of a topological space. In Figure 14.4, a bi-filtration of a triangulation of an ant has been defined considering the position of the simplices with respect to the y-axis and their discrete Gaussian curvature κ. Fixing one of the parameters, we obtain a filtration of the shape and hence, its representation in terms of persistence diagrams. Although this multidimensional model allows to build an accurate fingerprint of the shape, it is computationally hard. We guess that it would be possible to decrease the complexity of this problem, by consider the information obtained by the analysis of random filtrations. Similar diagrams obtained varying a single parameter identify the topological and geometrical property of the shape that are robust to this parameter. It would be possible to restate the evaluation of the parameters governing the multidimensional filtration as a problem of exploration/exploitation. On one side the (random) exploration of the space of filtrations. On the other side, the exploitation of the knowledge acquired by exploring the space randomly. In the future, it would be interesting to consider

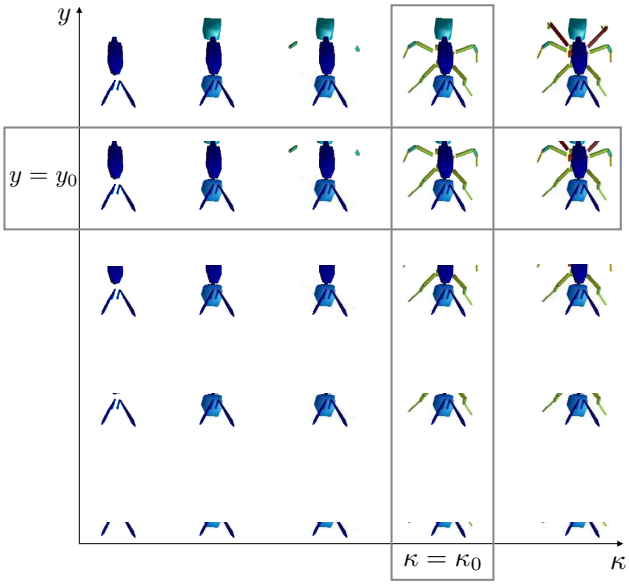


Figure 14.4: Multidimensional persistence. A 2-dimensional filtration, whose parameters are the discrete Gaussian curvature κ and the height function y. Persistent homology can be applied on each filtration obtained by fixing one of the two parameters.

this approach, inspired by the theory of reinforcement learning (Watkins and Dayan, 1992).

14.2.3 Audio feature deformed *Tonnetz*

In a second application, we modified either the labelling function associated to the vertices of the *Tonnetz* and the function used to deform its geometry. The former is restricted to the pitches corresponding to the chromatic scale built on a single octave, the latter computes the consonance (Plomp and Levelt, 1965) between a reference pitch and the pitches of the chromatic scale used to label the vertices.

The invariance of the consonance function in terms of uniform transposition of the chromatic scale and the reference notes is discussed, as well as its octave dependence. The space generated by deforming the *Tonnetz* with the consonance-based function is used to classify the 21 modal scales derived from the diatonic, melodic and harmonic minor scales. After fixing a reference pitch and a labelling of the vertices, we represented these scales as subsets of the 0-skeleton of the deformed *Tonnetz*. Thereafter, we consider the 0-persistent homology of each point cloud generated by its filtered Vietoris-Rips complex. The resulting consonance-based persistent fingerprint of modes recognises different sonorities, organising them in coherent clusters according to the disposition of their tension and resolution pitches on the *Tonnetz*. Moreover, the changes in the clustering given by the uniform transposition of the chromatic scale of an octave with respect to the reference pitch reflect the properties of the chords (first, third, fifth and seventh degrees) associated to the modal scales.

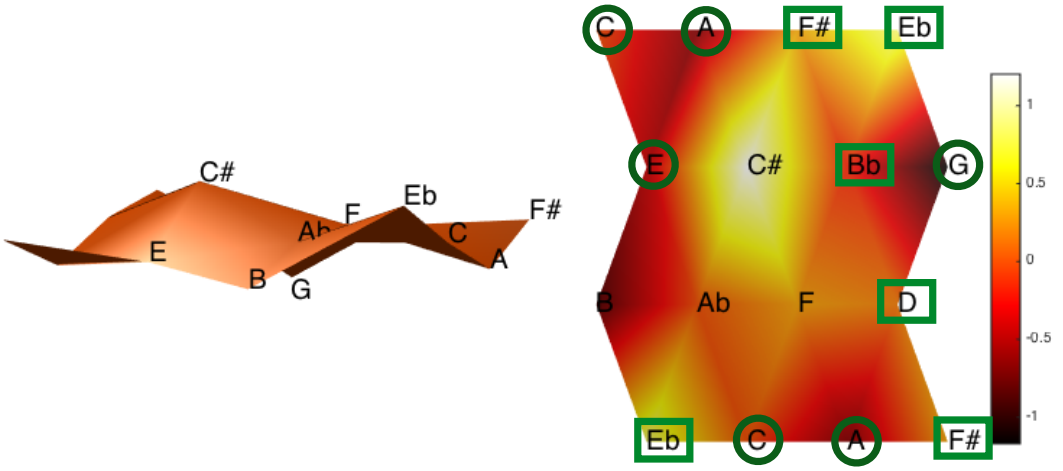


Figure 14.5: Configuration of the tensions (circles) and resolutions (squares) on the consonance-deformed *Tonnetz* obtained by considering block voicings of a major chord and the chromatic scale built an octave higher than the root note of the triad.

Melodic phrasing analysis

The space that we proposed is limited, since its geometry varies for uniform octave transposition of the chromatic scale used to label the vertices of the *Tonnetz*. Nevertheless, its topological properties are sensitive to the different sonorities we examined. Thus, a natural development of this work is the analysis of point clouds generated by other scales and, then, by more complex musical phrases. As we suggested for the model discussed above, it could be possible to describe additional features (for instance the occurrences of pitches) embedding the *Tonnetz* in a higher-dimensional space.

Learning the melodic phrasing

Every music teacher experienced how the concept of mode, sonority, or the use of *the right tension at the right moment* can be awkward for a beginner or even an experienced musician. We are surely far from explaining in a clear and unifying way these concepts. The geometric representation of consonance we suggested inherits the cultural and temporal dependence of the curve of Plomp and Levelt. Nevertheless, the visualisation of the consonance-deformed *Tonnetz* and the configurations of its vertices in terms of tension and resolution patterns could be used in a didactic context, to explain the aforementioned ideas. In Figure 14.5, the pitches (of the chromatic scale built an octave higher than the root of the chord) are highlighted to represent typical tensions and resolutions on a major triad, to obtain a bluesy sonority. Looking at this representation, one can build new patterns, that are not evident when the scales are visualised directly on an instrument. In addition, pitches connected by an edge on the *Tonnetz* or representing the vertices of a triangle are acoustically related. For instance, in the figure the tritone substitution ($G\flat$ major for C major) is represented as a triangle of tensions, juxtaposed to a triangle of resolutions (A minor triad). This representation gives an insight on harmonic

substitutions (often considered as complex rules), rather than valuable compositional or improvisational techniques.

14.2.4 Harmonic variable geometry

An extension of the consonance function to chords has been suggested and used to compute the deformation of the vertices of the *Tonnetz* labelled with the pitches of a chromatic scale. The height of each vertex is computed by considering the overall consonance of the superposition of its label and a fixed triad. In this setting, we compared the surfaces generated by six classes of triads. In particular, the values of the discrete Gaussian curvature have an interesting musical interpretation in terms of harmonic classification.

We classified the shapes obtained by deforming the *Tonnetz* with the six classes of triads, by computing their persistent homology. We compared the three different filtrations induced by the sub-level sets of the consonance function, the Vietoris-Rips complex and the discrete Gaussian curvature. The clustering of the shapes produced by the distance between their 0-persistence diagrams gave interesting preliminary results, that reflect different harmonic properties of the triads that we considered.

Chords and voicing classification

The model shall be tested on seventh chords. This analysis would provide an interesting vision of the block voicing technique, commonly used in jazz arrangements. Furthermore, we proved that in equal temperament the inversions of a chord are classified by their consonance value modulo changes of the harmonic spectrum used to compute the consonance of the voicings. The study of several families of chords and their inversions, in terms of geometrical configurations (shapes) of the deformed *Tonnetz* can lead to a classification of chords parametrisable in terms of harmonic spectra and voicing classes.

Configuration and generation

The variable geometry characterising both the modal and harmonic-oriented deformed *Tonnetz* can be used for the automatic generation of melodic lines. Figure 14.6 depicts the trajectories of several particles moving under gravity, on the surface of a deformed *Tonnetz*. Each trajectory depends on the mass and the initial condition (height, position with respect to the plane $z = 0$ and initial speed) of each particle. The retrieval of the nearest pitch along the trajectory of each particle in time allows to define a melody (in terms of pitches and durations). Furthermore, it could be interesting to study the configurations of more than one particle (Abrams and Ghrist, 2002; Ghrist and Peterson, 2007) by modelling the superpositions (and collisions) of several melodies on a given harmonic structure.

It is natural to extend this model to a dynamical framework. We can consider these trajectories while the geometry of the space updates according to a progression of bass or harmonic changes. On a given accompaniment, it would also be interesting to study the periodicity of the trajectories and introduce symmetries in order to influence the paths of the particles.

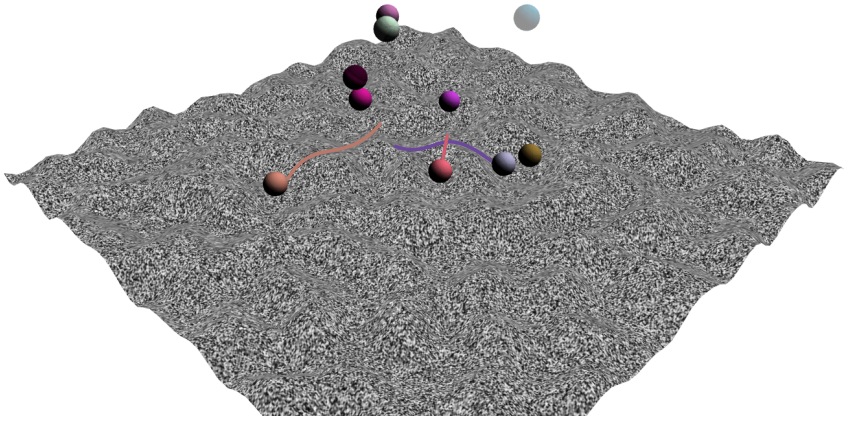


Figure 14.6: Gravity on the deformed *Tonnetz*. Masses move following the deformation of the surface. The pitches or pitch classes lying in a neighbourhood of the trajectories can be used to generate melodic lines.

14.3 Harmonic and persistence time series

The horizontal representation did not take into account the intervallic relationships between the notes constituting the domain and codomain of the voice leading. The vertical analysis neglected the ordering of notes and chords in time. In the two last applications, we recovered this ordering. On one hand, we considered time series whose observations are computed from a harmonic analysis of a composition. On the other hand, persistence time series represent the geometrical properties of the deformation of the *Tonnetz* at evenly-spaced instants.

14.3.1 Multiple (harmonic) sequences alignment

We suggested a novel method based on multiple sequences alignment, for the analysis of pop music. Albeit global pairwise alignment has already been applied to music (Pardo and Sanghi, 2005; İzmirli and Dannenberg, 2010; Martin et al., 2012), our approach consists in the exploitation of the both the signal and symbolic information and in the definition and test of a set of *ad hoc* tools for the alignment of harmonic-oriented symbolic sequences. Furthermore, multiple sequences alignment allowed us to give an encompassing analysis of the propagation of the musical inspiration among different artists, genres and time.

Abstract descriptors

Historically, the music analysis community has been devided between the symbolic (musical notations, discrete nature) and signal-oriented (continuous flux of information) research paradigms. In this signal/symbol application, the analysis of the audio allowed to extract musical descriptors, that the symbolic approach endowed with a higher-level of abstraction. In our analyses, these high-level music features (as modulations or cadential patterns) reveal surprising properties of music, that are normally neglected by the standard notion of musical genre. We suggested in the introduction of this work, that a composition is built on a collection of core

Figure 14.7: Dendrogram chasing. For each branch of the dendrogram it is possible to build a consensus sequence, that describe the similarity between the sequences of the cluster once they have been aligned.

concepts: strong musical ideas that are clearly grasped by the listeners and that allow musicians to build their own vocabulary. The high-level descriptors allow to represent the propagation of these musical concepts, retrieving them in compositions that do not belong to the same genre.

Music molecular clock

The Molecular Clock Hypothesis (MCH) consists in the reconstruction of the evolutional history of species by measuring the variations of particular structure (for instance the haemoglobin) that occur at an almost constant time rate. It would be pretentious to translate literally this assumption to music. However, the construction we proposed allows to follow the evolution of a particular motif, through the *chase in the dendrogram* depicted in Figure 14.7. The cluster in this figure has been obtained by computing the pairwise alignment of sequences of degrees (cadential patterns) among the 138 Quaero's songs we analysed, utilising the weighting matrix deduced by the spiral array and the NW algorithm. Red points corresponds to the *consensi*, or in MCH terms to the common ancestors of the songs represented in the cluster. Considering these points from the right to the left of the figure, the first consensus sequence corresponds to the synthesis of cadential solutions between the very similar *Baby's Black* and *Polythene Pam*. Then, by climbing toward the centre of the dendrogram, to broader syntheses of the cadential patterns used in the whole set of multiple aligned songs. The information carried by these *common ancestors* can be relevant either in musicological studies of particular artists, genres or periods and for music recommendation.

Computer-assisted composition

The results in terms of pairwise and multiple alignment, motif mining and the computation of the consensus sequences can be used for computer-assisted composition. Indeed, pairwise alignment provides a local description of the similarity regions of two compositions and induces a grouping of a set of songs. Multiple alignment allows to chase similarity patterns in the diagram generated by the simultaneous alignment of a set of songs and highlighted by the analysis of motifs. Finally, the consensus sequences provide a candidate capable to summarise in its structure the

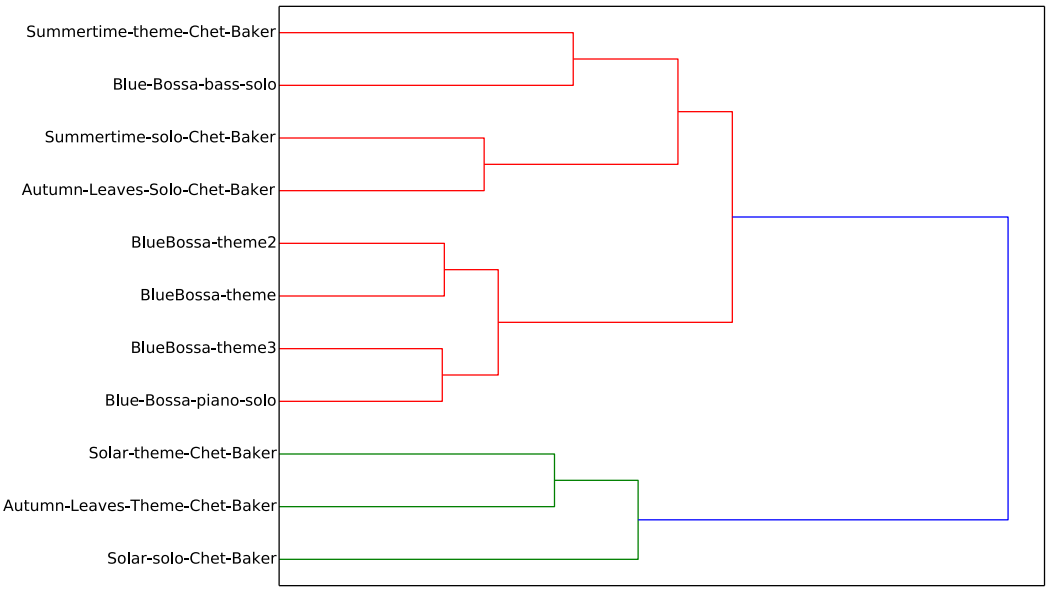


Figure 14.8: Static classification between three Chet Baker's themes and improvisations and a version of *Blue Bossa*. Two solos by the same author are grouped together, while the bass solo of *Blue Bossa* is linked to the theme of *Summertime* at a high distance.

common features of a set of previously aligned sequences. In musical terms and in our application, consensus sequences depict a possible synthesis of the coherent harmonic solutions provided by different artists, in a different time.

14.3.2 Persistence time series

The last model we proposed consists in the representation of the deformation of the *Tonnetz* in time, as a time series of persistence diagrams. We provided a musical interpretation of the sequences of diagrams and utilised the dynamic time warping algorithm with the bottleneck distance as cost function, to compute the optimal warping path between two compositions and their alignment score. These computations provide an alignment between two compositions and a dissimilarity score. We analysed three datasets collecting classical, pop and jazz compositions, by considering time series of evenly-spaced observations at 8, 4 and 2-bars respectively. The dissimilarity scores and the analysis of their variations with respect to the windowing, provide a good stylistic descriptor of music able to distinguish among tonal, atonal and modal classical compositions and sensitive to the use of different tensional paradigms in pop music.

Improviser retrieval

The jazz dataset represents a point of interest due to the improvisational nature of the compositions. In this case the information provided by the optimal warping path allows to distinguish between themes and solo parts, in two versions of the same standard. This method can be used to measure the distance between improvisational

styles. This particular development is also suggested by the preliminary result we obtained considering the static deformed *Tonnetz* depicted in Figure 14.8.

Musical concepts: granularity and propagation

In order to interpret the evolution of a system in time as a collection of observations, we defined a windowing, consisting in an even partition of the composition according to its subdivision in bars. We saw how the three datasets we considered respond differently to changes of this windowing. In musical terms, this corresponds to the pace at which musical concepts evolve in the composition. It would be interesting to compute the optimal time *granularity* necessary to describe the evolution of compositions belonging to different genres or artists.

Mean persistence diagrams

The computation of the pairwise bottleneck distances between the observations of two persistence time series has a high computational cost. A possible development of our model consists in the evaluation of the strategy introduced in (Munch, 2013) in which a unique average persistence diagram is associated to the vineyard associated to a time-varying system. In terms of classification, it would be interesting to compare the results obtained through our primary static analysis and the corresponding mean persistence diagrams. In addition, their differences could be relevant in order to compute the optimal granularity mentioned above or at least to bound its value.

Persistence time series analysis

The strategy we proposed in order to compare the evolution of the persistence properties of two time-varying spaces is a new approach. Its efficiency can be tested in many applications, such as animals tracking (Pérez-Escudero et al., 2014) and group behaviour (Topaz et al., 2015).

Memory

The dynamic *Tonnetz* has full memory of the composition: the heights of its vertices increases monotonically. This feature does not reflect our perception of music, since we cannot remember every note of a whole composition. The definition of a gravity function in opposition to the one generating the deformation of the vertices can be used to endow this space with a type of short-term memory. The same argument can be applied to the study of dynamical consonance-based deformed *Tonnetze*. In addition, we can define a variable gravitational field (or equip the vertices with variable masses) in order to diminish the effect of the gravity in correspondence of vertices representing relevant elements of a musical phrase (for example its higher, lower, first, ending and syncopated notes), we refer interested readers to (Perricone, 2000).

Part VI

Appendices

A

Modes in Modern Music and a Topological viewpoint

The aim of this chapter is twofold. On one hand, it provides both the basic references concerning modal theory and the definition of mode in a modern music context (Appendix A.1). On the other hand, it gives an intuitive, topological-oriented point of view on modal theory. A mode is interpreted as a superposition of a four-note chord and a triad. This construction follows naturally when modes are deduced from the harmonisation of a scale: the four-note chord (named the *base-chord* in the remainder of this chapter) represents the set of *resolutions* of the modal scale it defines, while a triad of *tensions* is formed by the second, fourth and sixth degree of the modal scale.

Finally, we associate to each class of base-chord an oriented planar connected graph. This association allows both to define a new family of modes and to provide a topological, qualitative description of seven classes of four-note chords.

Remark 17. What follows is a shortened, modified version[1] of (Bergomi and Portaluri, 2013).

A.1 Standard modes as superposition of chords

From a music theory viewpoint, modes are defined as *a seven-note scales created by starting on any of the seven notes of a major or a melodic minor scale* (Levine, 2011). In the following paragraphs, we apply this definition to deduce a family 21 modal scales and discuss their hidden harmonic nature.

A.1.1 Deducing the standard modes

Following the definition given in (Levine, 2011), we assume that a mode is a heptatonic scale, whose structure is inherited from a *tonal* scale. In table A.1 we list the 21 modes built on each degrees of the diatonic, melodic minor and harmonic minor scale. We include the harmonic minor scale, since *its* modes are widely used in several musical contexts. See (Bergomi and Geravini, 2012) for details and examples.

A comparison between Tables A.1 and A.2 reveals that the idea of mode is deeply related to a harmonic choice: a modal scale is recognisable if it is played together with either a reference pitch (its root), or a chord.

[1]The full text is available at `http://arxiv.org/abs/1309.0687`.

Scale	Degree	Modes	Example
Major	I	Ionian	C - D - E - F - G - A - B
	II	Dorian	D - E - F - G - A - B - C
	III	Phrygian	E - F - G - A - B - C - D
	IV	Lydian	F - G - A - B - C - D - E
	V	Mixolydian	G - A - B - C - D - E - F
	VI	Eolian	A - B - C -D - E - F - G
	VII	Locrian	B - C - D - E - F - G - A
Melodic Minor	I	Hypoionian	C - D - E♭ - F - G - A - B
	II	Dorian ♭2	D - E♭ - F - G - A - B - C
	III	Lydian Augmented	E♭ - F - G - A - B - C - D
	IV	Lydian Dominant	F - G - A - B - C - D - E♭
	V	Mixolydian ♭13	G - A - B - C - D - E♭ - F
	VI	Locrian ♯2	A - B - C -D - E♭ - F - G
	VII	Super locrian	B - C - D - E♭ - F - G - A
Harmonic Minor	I	Hypoionian ♭6	C - D - E♭ - F - G - A♭ - B
	II	Locrian ♯6	D - E♭ - F - G - A♭ - B - C
	III	Ionian augmented	E♭ - F - G - A♭ - B - C - D
	IV	Dorian ♯4	F - G - A♭ - B - C - D - E♭
	V	Phrygian dominant	G - A♭ - B - C - D - E♭ - F
	VI	Lydian ♯2	A♭ - B - C - D - E♭ - F - G
	VII	Ultra locrian	B - C - D - E♭ - F - G - A♭

Table A.1: The 21 modes derived from the major, melodic minor and harmonic minor scale. Examples have been built on the C major, melodic minor and harmonic minor scale, respectively.

A.1.2 Modes as superposition of chords

In this paragraph we want to stress the importance of the harmonic choice lying behind the modal scale.

Consider both the seventh chord and the modal scale built on the same degree of a tonal scale. It is easy to see that the pitch classes composing the chord are the first, third, fifth and seventh degrees of the modal scale. In a modern music context, it is possible to refer to these pitch classes as resolutions.

Example A.1.1. Consider F lydian. The chord associated to this mode is $Fmaj7$, then we have

$$F \text{ lydian scale} \quad F - G - A - B - C - D - E$$
$$Fmaj7 \text{ arpeggio} \quad F - A - C - E$$

Given a mode, we refer to the seventh chord built on its root as the *base-chord* of the mode. Thus, we call *tension-triad*, the triad composed by the second, the fourth and the sixth degree of the modal scale, obtained by deleting the pitch classes of the base chord.

$$F \text{ lydian scale} \quad F - G - A - B - C - D - E$$
$$\text{Base-chord arpeggio} \quad F - A - C - E$$
$$\text{Tension-traid arpeggio} \quad G - B - D$$

Scale	Degree	7^{th} Chord	Arpeggio (example)
Major	I	$maj7$	C - E - G - B
	II	-7	D - F - A - C
	III	-7	E - G - B - D
	IV	$maj7$	F - A - C - E
	V	7	G - B - D - F
	VI	-7	A - C - E - G
	VII	$-7^{\flat 5}$	B - D - F - A
Melodic Minor	I	$-maj7$	C - E♭ - G - B
	II	-7	D - F - A - C
	III	$maj7^{\sharp 5}$	E♭ - G - B - D
	IV	7	F - A - C - E♭
	V	7	G - B - D - F
	VI	$-7^{\flat 5}$	A - C - E♭ - G
	VII	$-7^{\flat 5}$	B - D - F - A
Harmonic Minor	I	$-maj7$	C - E♭ - G - B
	II	$-7^{\flat 5}$	D - F - A♭ - C
	III	$maj7^{\sharp 5}$	E♭ - G - B - D
	IV	-7	F - A♭ - C - E♭
	V	7	G - B - D - F
	VI	$maj7$	A♭ - C - E♭ - G
	VII	$\circ 7$	B - D - F - A♭

Table A.2: Seventh chord harmonisation on the major, melodic minor and harmonic minor scale. Refer to Appendix E for details on modern chord notation.

The F lydian mode is given by the superposition of a $Fmaj7$ chord and a G major triad.

Every modal scale can be decomposed uniquely in a seventh chord built on its root and a triad built on its second degree. A modal scale is recognisable, if it is played on its base-chord, so one can consider the base-chord as the set of stable notes and the tension-triad as the collection of tension notes of the mode[2]. See table A.3 for a complete description of modes in terms of base-chords and tension-triads. This decomposition arises naturally from the musical-oriented definition of modes of (Levine, 2011).

It is possible to associate a family of modes to each class of base-chord $[B]$ by varying the tension triad $[T]$. In Table A.4 modes are organised according to their base-chord classes. For instance, let $[B]$ be a major seven chord, then, it is possible to associate to $[B]$ three different classes of tension-triads $[T_i]$. For example, by considering the pitch class C as root of the modal scales and $maj7$ as chord class, we can describe the three modes associated to the base chord $Cmaj7$ as

1. Ionian: $i := (Cmaj7, D-)$;

[2]The term mode here is intended as a non necessarily ordered modal scale, played on its base chord, or at least with its root as accompaniment.

Mode	Scale	Base-chord	Tension-triad
C Ionian	C - D - E - F - G - A - B	$Cmaj7$	$D-$
D Dorian	D - E - F - G - A - B - C	$D-7$	$E-$
E Phrygian	E - F - G - A - B - C - D	$E-7$	F
F Lydian	F - G - A - B - C - D - E	$Fmaj7$	G
G Mixolydian	G - A - B - C - D - E - F	$G7$	$A-$
A Eolian	A - B - C - D - E - F - G	$A-7$	$B-\flat5$
B Locrian	B - C - D - E - F - G - A	$B-7^{\flat5}$	C
C Hypoionian	C - D - E$\flat$ - F - G - A - B	$C-maj7$	$D-$
D Dorian $\flat2$	D - E$\flat$ - F - G - A - B - C	$D-7$	$E\flat^{\sharp5}$
$E\flat$ Lydian Augmented	E$\flat$ - F - G - A - B - C - D	$E\flat maj7^{\sharp5}$	F
F Lydian Dominant	F - G - A - B - C - D - E$\flat$	$F7$	G
G Mixolydian $\flat13$	G - A - B - C - D - E$\flat$ - F	$G7$	$A-\flat5$
A Locrian $\sharp2$	A - B - C -D - E$\flat$ - F - G	$A-7^{\flat5}$	$B-\flat5$
B Super locrian	B - C - D - E$\flat$ - F - G - A	$B-7^{\flat5}$	$C-$
C Hypoionian $\flat6$	C - D - E$\flat$ - F - G - A$\flat$ - B	$C-maj7$	$D-\flat5$
D Locrian $\sharp6$	D - E$\flat$ - F - G - A$\flat$ - B - C	$D-7^{\flat5}$	$E\flat^{\sharp5}$
$E\flat$ Ionian augmented	E$\flat$ - F - G - A$\flat$ - B - C - D	$E\flat maj7^{\sharp5}$	F
F Dorian $\sharp4$	F - G - A$\flat$ - B - C - D - E$\flat$	$C-7$	G
G Phrygian dominant	G - A$\flat$ - B - C - D - E$\flat$ - F	$G7$	$A-\flat5$
$A\flat$ Lydian $\sharp2$	A$\flat$ - B - C -D - E$\flat$ - F - G	$Amaj7$	$B-\flat5$
B Ultra locrian	B - C - D - E$\flat$ - F - G - A$\flat$	$B^{\circ7}$	$C-$

Table A.3: Modes as a superposition of two chords.

2. Lydian: $l := (Cmaj7, D)$;
3. Lydian $\sharp2$: $l_{\sharp2} := (Cmaj7, D\sharp^{-\flat5})$

Every couple $([B], [T])$ can be associated uniquely to a modal scale. The scale is given by the set of notes $\{b_1, b_2, b_3, b_4, t_1, t_2, t_3\}$ where $B = \{b_1, \ldots, b_4\}$ and $T = \{t_1, t_2, t_3\}$. Following (Piston et al., 1978, Chapter 10) for the analysis of non-harmonic tones in classical harmony we are entitled to define

Definition A.1.1. Let $B = \{b_1, \ldots, b_4\}$ and $T = \{t_1, t_2, t_3\}$. Non-chord tones are pitch classes that do not belong to the base-chord; i.e.

$$t_i \in T \text{ such that } t_i \notin B.$$

Let $m = ([B], [T])$, then

1. m identifies a unique mode;
2. chord tones and non chord tones are splitted into two components, respectively $[B]$ and $[T]$
3. considering the notes belonging to $[B]$ and $[T]$ we deduce the modal scale associated to m, that can be re-ordered in a 7-uple in which the degrees of the scale are displayed from the root, to the seventh note.

Base-chord $maj7$	Modes	Example (root C)
T-M III-P V-M VII	Ionian	C - D - E - F - G - A - B
	Lydian	C - D - E - F♯ - G - A - B
	Lydian ♯2	C - D♯ - E - F♯ - G - A - B
Base-chord $maj7^{\sharp 5}$	Modes	Example (root C)
T-M III-aug V-M VII	Lydian augmented	C - D - E - F♯ - G♯ - A - B
	Ionian augmented	C - D - E - F - G♯ - A - B
Base-chord 7	Modes	Example (root C)
T-III M-V P-VII m	Mixolydian	C - D - E - F - G - A - B♭
	Mixolydian ♭13	C - D - E - F - G - A♭ - B♭
	Phrygian dominant	C - D♭ - E - F - G - A♭ - B♭
	Lydian dominant	C - D - E - F♯ - G - A - B♭
Base chord -7	Modes	Example (root C)
T-m III-P V-m VII	Dorian	C - D - E♭ - F - G - A - B♭
	Phrygian	C - D♭ - E - F - G - A♭ - B♭
	Eolian	C - D - E♭ - F - G - A♭ - B♭
	Dorian ♭2	C - D♭ - E♭ - F - G - A - B♭
	Dorian ♯4	C - D - E♭ - F♯ - G - A - B♭
Base chord $-7^{\flat 5}$	Modes	Example (root C)
T-m III-dim V-m VII	Locrian	C - D♭ - E♭ - F - G♭ - A♭ - B♭
	Locrian ♯2	C - D - E♭ - F - G♭ - A♭ - B♭
	Superlocrian	C - D♭ - E♭ - F$flat$ - G♭ - A♭ - B♭
	Locrian ♯6	C - D♭ - E♭ - F - G♭ - A - B♭
Base-chord $-maj7$	Modes	Example (root C)
T-m III-P V-M VII	Hypoionian	C - D - E♭ - F - G - A - B
	Hypoionian ♭6	C - D - E♭ - F - G - A♭ - B
Base-chord ∘7	Modes	Example (root C)
T-m III-dim V-dim VII	Ultralocrian	C - D♭ - E♭ - F - G♭ - A♭ - B♭♭

Table A.4: Modal scales associated to a fixed base-chord

Thus, it is possible to associate to a fixed base-chord an ordered modal scale for every available choice of tension-triad.

Example A.1.2. Fix a seventh chord, for instance a Cmaj7. The idea is to split tensions and resolutions according to the ordering induced by the modal scale as follows

$$C \rightarrow \square \rightarrow E \rightarrow \square \rightarrow G \rightarrow \square \rightarrow B$$

White squares are placeholders for the note of a suitable triad. As we showed in the previous paragraph, choosing a D minor triad one can find the *C ionian* scale, considering a D major triad we have a *C lydian* and with a $D\sharp$ diminished triad we obtain the *C lydian* ♯2 scale.

$$C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow A \rightarrow B$$

$$\text{C} \rightarrow \text{D} \rightarrow \text{E} \rightarrow \text{F}\sharp \rightarrow \text{G} \rightarrow \text{A} \rightarrow \text{B}$$

$$\text{C} \rightarrow \text{D}\sharp \rightarrow \text{E} \rightarrow \text{F}\sharp \rightarrow \text{G} \rightarrow \text{A} \rightarrow \text{B}$$

Remark 18. The following section lies beyond the scope of this appendix. However, it represents the first effort towards a topological interpretation of music I made with Alessandro Portaluri, thus I decided to add it to this thesis.

A.2 A geometrical representation of modes through graphs

In this section we suggest an elementary topological-oriented analysis of modes and modal scales. Fixed a seventh chord, a particular graph is used both to provide an intuitive visualisation of the possible modal choices associated to the seventh chord and to give a qualitative description of the seventh chord classes we listed in the first column of Table A.4.

A.2.1 Some mathematical preliminaries

Definition A.2.1. Two abstract (unoriented) graphs (V, E) and (V', E') are *isomorphic* if there exists a bijective map $f : V \to V'$ such that

$$\{v, w\} \in E \iff \{f(v), f(w)\} \in E'.$$

Remark 19. Analogous definitions for oriented graphs are obtained by replacing unordered pairs $\{\cdot, \cdot\}$ by ordered pairs $(\cdot, \cdot)$.

Definition A.2.2. For $n \geqslant 1$, a path on a graph G from v^1 to v^{n+1} is a sequence of vertices and edges

$$v^1 e^1 v^2 e^2 \dots v^n e^n v^{n+1}$$

where $e^1 = (v^1 v^2)$, $e^2 = (v^2 v^3), \dots, e^n = (v^n v^{n+1})$.

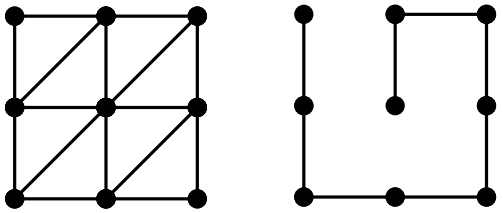
If G is oriented, we only require that $e^i = (v^i v^{i+1})$ or $e^i = (v^{i+1} v^i)$ for $i = 1, \dots, n$; that is, the edges along the path are oriented in the opposite way.

Definition A.2.3. The path is *simple* if $e^1, \dots, e^n$ are all distinct, and $v^1, \dots, v^{n+1}$ are all distincts except that possibly $v^1 = v^{n+1}$. If the simple path has $v^1 = v^{n+1}$ and $n > 0$ is called a *loop*.

A graph G is said to be *connected* if, given any two vertices v and w of G there is a path on G from v to w. A graph which is connected and without loops is called a *tree*.

Definition A.2.4. Given a graph G, a graph H is called a *subgraph* of G is the vertices of H are vertices of G and the edges of H are edges of G. Also H is called a *proper subgraph* of G if $H \neq G$.

The following definition is central in the remainder.



(a) An example of a planar grap (b) A maximal tree

Definition A.2.5. Let G be a graph, H be any maximal tree in G, S be a subset of the vertices set and k is an integer. An *admissible path γ in G with respect to S of length k* is any proper subgraph of H satisfying the two conditions:

1. each vertex $v \in S$ lies in γ;
2. the total number of vertices in γ is k.

We also observe that any graph G as a subgraph which is a tree (e.g. the empty subgraph is a tree) so that the set $\mathscr{T}$ of subgraphs of G which are trees will have maximal elements. That is, there exists at least one $T \in \mathscr{T}$ such that T is not a proper subgraph of any $T' \in \mathscr{T}$.

Lemma A.2.1. *Let G be a connected graph. A subgraph T of G is a maximal tree for G if and only if T is a tree containing all the vertices of G.*

Proof. Cfr. (Giblin, 2010, Proposition 1.11, pag.18). □

For a connected graph G there is a standard way to compute the homotopy group. In fact the following result holds:

Proposition A.2.2. *For a connected graph G with maximal tree T, $\pi_1(G)$ is a free group with basis the classes $[f_\alpha]$ corresponding to the edges e_α of $X \setminus T$.*

Proof. Cfr. (Hatcher, 2002, Proposition 1A.2 pag.84). □

A.2.2 Graphs and base-chords

Definition A.2.6. Given a base-chord $[B]$ the associated graph $\mathscr{G}([B])$ is the realisation of the abstract graph whose vertex set is given by the set of all notes forming $[B]$ and of every compatible tension-triad. The oriented edge set is represented by all possible oriented connections between each vertex, according to the order of the degrees of the scale; i.e. from the root to the seventh. (See Figure A.1).

Associating a graph to a modal scale, it is possible to arrange its degrees in the plane in infinitely many ways. However considering the orientation induced by the degrees of the scale all the oriented graphs are homeomorphic. Since homeomorphisms induce isomorphisms in homotopy, all of the homotopic classification is not affected by the convention given in definition A.2.6. We also observe that on the second, fourth and sixth degrees we have at most two choices. This is a straightforward

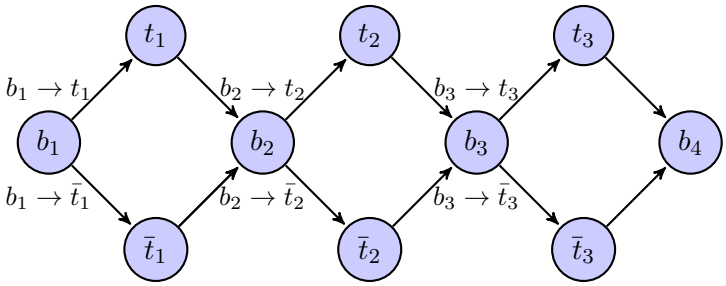


Figure A.1: A graph built assuming that the modal choices on a base-chord $B = \{b_1, b_2, b_3, b_4\}$ are given by two tension-triads $T = \{t_1, t_2, t_3\}$ and $\overline{T} = \{\bar{t}_1, \bar{t}_2, \bar{t}_3\}$

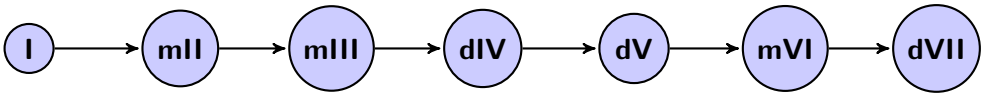


Figure A.2: The graph associated to the diminished seventh chords, $\Gamma_{\circ 7}$.

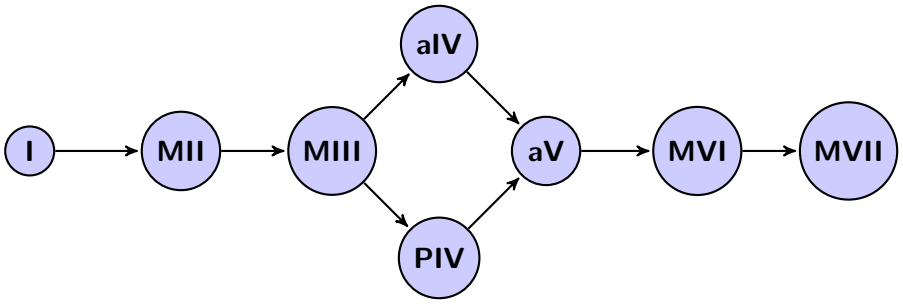


Figure A.3: The graph associated to diminished seventh chords, $\Gamma_{maj7^{\sharp 5}}$.

consequence of the constructions of the modes from the major, melodic minor and harmonic minor scales.

Following definition A.2.6, it is possible to associate a graph to each base-chord class:

$$^{\circ 7},\ maj7^{\sharp 5},\ -maj7,\ maj7,\ 7,\ -7,\ -7^{\flat 5}.$$

1. **Diminished seven:** $\Gamma_{\circ 7}$. This type of chord appears only in the harmonisation of the seventh degree of the harmonic minor scale, therefore the only available mode is the ultralocrian. Its graph (and *a fortiori* maximal tree) is represented in **??**

2. **Major seven ♯5:** $\Gamma_{maj7^{\sharp 5}}$**.** Fixing a $maj7^{\sharp 5}$ chord as *base* of the mode, we have two different possibilities: either the ionian sharp five or the lydian sharp five modal scale **??**. See **??** for an example of maximal three of $\Gamma_{maj7^{\sharp 5}}$.

3. **Minor major seven:** Γ_{-maj7}. In this case we can choose between two different modes: hypoionian and hypoionian ♭6. The graph is

4. **Major seven:** Γ_{maj7}**.** This is certainly a more common chord than the previous ones. We expect to have more possibilities, in fact a well known

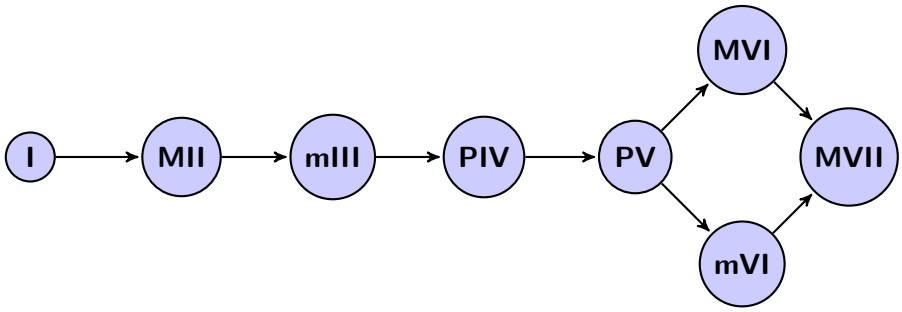


Figure A.4: The graph associated to minor major seventh chords, Γ_{-maj7}.

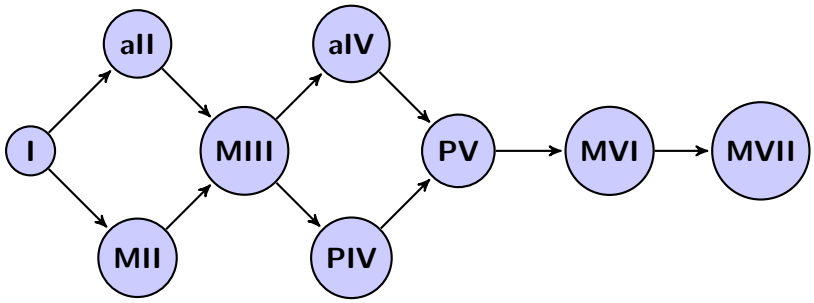


Figure A.5: The graph associated to major seven chords, Γ_{maj7}.

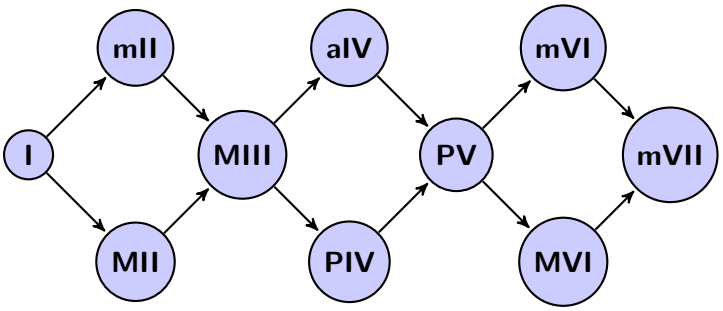


Figure A.6: The graph associated to dominant chords, Γ_7.

and simple *base-chord* surely will bear more *tension-triads* than a naturally dissonant one, as depicted in Figure A.5.

5. **Dominant:** Γ_7 **.** Dominant chords are largely used in blues and traditional jazz thanks to their capability of bearing tensions. The graph associated to this chord class is depicted in Figure A.6.

6. **Minor seven:** Γ_{-7}**.** For a minor seventh chord the only forbidden notes are the augmented second and the diminished fourth. So, we obtain a graph isomorphic to Γ_7 in Figure A.7.

7. **Minor seven** $\flat 5$**:** $\Gamma_{-7\flat 5}$**.** In this case, the root note and the diminished fifth form a tritone interval which gives a *stable* sense of dissonance to the half-diminished seventh chord, that is emphasised by the minor second which

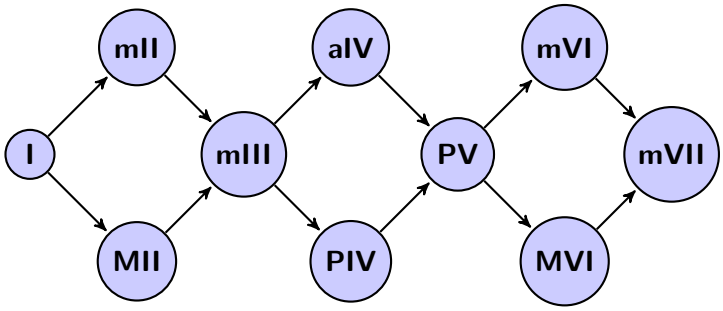


Figure A.7: The graph associated to minor seven chords, Γ_{-7}.

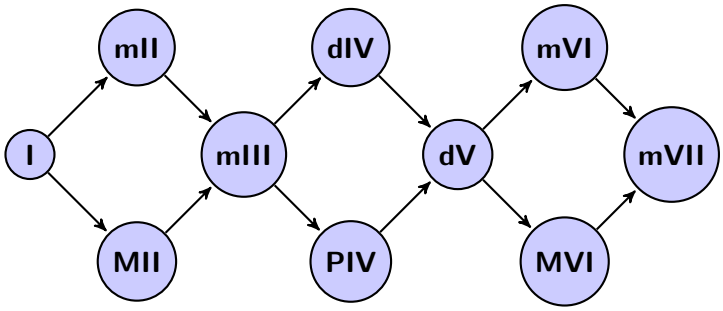


Figure A.8: The graph associated to minor seven flat five chords, $\Gamma_{-7^{\flat 5}}$.

is natural in three of the four modal solutions we find by considering this chord class, i.e. locrian, superlocrian and locrian ♯6 scales. See Figure A.8.

Remark 20. It is clear by previous discussion that even if the graphs associated to Γ_7, Γ_{-7}, $\Gamma_{-7^{\flat 5}}$ are isomorphic, they are built on different notes and hence they are quite different in the essence so the homotopy is not suitable to distinguish among them.

All of these graphs show in a clear and net way how to construct new modes from the existing ones. Given a graph G, let us consider any proper tree H (not maximal, in general!) contained in G and having 7 vertices. By taking into account definition A.2.5, we give the following:

Definition A.2.7. Let $[B]$ a base-chord and $\mathscr{G}([B])$ be the associated base-chord graph. An *admissible mode* is any admissible connected subgraph (or path in the graph) $\gamma([B])$ in $\mathscr{G}$ with respect to $[B]$ of length 7. If $\gamma([B])$ is not a mode constructed above, we refer to as *admissible special mode.*

Proposition A.2.3. *Given the base-chord class* $[B]$ *the following modes are the only admissible special modes:*

1. *if* $[\Gamma_B] = [\Gamma_{maj7}]$ *then* $\gamma([B])$ *is the path*

$$\gamma_{Ion\sharp 2} := \{I, aII, MIII, PIV, PV, MVI, MVII\}$$

2. *if* $[\Gamma_B] = [\Gamma_7]$ *then* $\gamma([B])$ *are the paths*

$$\begin{aligned}
\gamma_{Mix\flat 2} &:= \{I, mII, MIII, PIV, PV, MVI, mVII\} \\
\gamma_{Mix\flat 2\sharp 4} &:= \{I, mII, MIII, aIV, PV, MVI, mVII\} \\
\gamma_{Mix\sharp 4\flat 6} &:= \{I, MII, MIII, aIV, PV, mVI, mVII\} \\
\gamma_{Mix\flat 2\sharp 4\flat 6} &:= \{I, mII, MIII, aIV, PV, mVI, mVII\}
\end{aligned}$$

3. *if* $[\Gamma_B] = [\Gamma_{-7}]$ *then* $\gamma([B])$ *are the paths*

$$\begin{aligned}
\gamma_{Eol\flat 2} &:= \{I, mII, mIII, PIV, PV, mVI, mVII\} \\
\gamma_{Eol\sharp 4} &:= \{I, MII, mIII, aIV, PV, mVI, mVII\} \\
\gamma_{Phr\sharp 4} &:= \{I, mII, mIII, aIV, PV, mVI, mVII\}
\end{aligned}$$

4. *if* $[\Gamma_B] = [\Gamma_{-7\flat 5}]$ *then* $\gamma([B])$ *are the paths*

$$\begin{aligned}
\gamma_{Loc\sharp 2\sharp 6} &:= \{I, mII, mIII, PIV, dV, mVI, mVII\} \\
\gamma_{Sup\sharp 2} &:= \{I, MII, mIII, dIV, dV, mVI, mVII\} \\
\gamma_{Sup\sharp 6} &:= \{I, mII, mIII, dIV, dV, MVI, mVII\} \\
\gamma_{Sup\sharp 2\sharp 6} &:= \{I, MII, mIII, dIV, dV, MVI, mVII\}
\end{aligned}$$

Proof. The result readily follows by the previous graph classification. Let us consider every class of chord to prove the existence of special modes.

- $^{\circ 7}$. It is not possible to have path associated to special modes on the graph $\Gamma_{\circ 7}$ (Figure A.2), since there is only one path available which represent the ultra locrian mode.
- $maj7^{\sharp 5}$ and $-maj7$. In both $\Gamma_{maj7^{\sharp 5}}$ (Figure A.3) and Γ_{-maj7} (Figure A.4) The only available choice is on the fourth and the sixth degree of the modal scale, respectively. This fact implies that only two admissible modes can be built on such graph and they differs exactly for one note. So we can choose among two paths on the graph which are exactly the two admissible modes we used to build the graph.
- $maj7$. In Γ_{maj7} (Figure A.5) there are 2^2 available choices. The modes which generate this graph are three, so there is a special mode which represent the admissible path on Γ_{maj7} which is different from the paths representing the Ionian, Lydian and Lydian $\sharp 2$ scales. The only possible, admissible path is

$$\{I, aII, MIII, PIV, PV, MVI, MVII\}.$$

- 7. Four admissible non special modes generate Γ_7 (see Table A.4 and Figure A.6). The total number of admissible modes in this graph is $2^3 = 8$. We expect to find 4 special modes:

$$\begin{gathered}
\{I, mII, MIII, PIV, PV, MVI, mVII\} \\
\{I, mII, MIII, aIV, PV, MVI, mVII\} \\
\{I, MII, MIII, aIV, PV, mVI, mVII\} \\
\{I, mII, MIII, aIV, PV, mVI, mVII\}
\end{gathered}$$

- -7 and $-7^{\flat 5}$. This cases are similar to the previous one. Γ_{-7} (Figure A.7) is generated by 5 admissible, non special modes (table A.4), so we have 3 special modes which are

$$\{I, mII, mIII, PIV, PV, mVI, mVII\}$$
$$\{I, MII, mIII, aIV, PV, mVI, mVII\}$$
$$\{I, mII, mIII, aIV, PV, mVI, mVII\}.$$

  $\Gamma_{-7^{\flat 5}}$ (Figure A.8) is generated by four admissible non special modes (table A.4), we have the following four special modes:

$$\{I, mII, mIII, PIV, dV, mVI, mVII\}$$
$$\{I, MII, mIII, dIV, dV, mVI, mVII\}$$
$$\{I, mII, mIII, dIV, dV, MVI, mVII\}$$
$$\{I, MII, mIII, dIV, dV, MVI, mVII\}$$

□

A.2.3 A qualitative description of the base-chord classes

The aim of this section is to associate a value to each base chord, with respect to the connections of its graph, as we defined it in the previous section.

Definition A.2.8. Given a base-chord $[B]$, let $\mathscr{G}([B])$ be the associated base-chord graph. We call *topological quality* of $[B]$, i.e. $\tau([B])$, the number of generators of the fundamental group of $\mathscr{G}([B])$.

Lemma A.2.4. *Let $[B]$ be a base-chord. The integer $\tau([B])$ is well-defined.*

Proof. It is enough to observe that given any base-chord $[B]$, the associated integer $\tau([B])$ is uniquely defined. In fact by the classification given in section A.2, at each base-chord $[B]$ we can uniquely associate a planar connected graph $\mathscr{G}([B])$. As direct consequence of proposition A.2.2, the fundamental group of $\pi_1(\mathscr{G}([B]))$ is a free group having $\tau([B])$ generators. □

Proposition A.2.5. *Let $[B]$ be a base-chord, $\mathscr{G}([B])$ be a planar graph. Then the fundamental group $\pi_1(\mathscr{G}([B]))$ and the topological measure of complexity are given in the table below.*

Base-chord [B]	$\pi_1([B])$	$\tau([B])$
$\circ 7$	$\{1\}$	*0*
$maj7^{\sharp 5}$	$\mathbb{Z}$	*1*
$-maj7$	$\mathbb{Z}$	*1*
$maj7$	$\mathbb{Z}^{*2}$	*2*
7	$\mathbb{Z}^{*3}$	*3*
-7	$\mathbb{Z}^{*3}$	*3*
$-7^{\flat 5}$	$\mathbb{Z}^{*3}$	*3*

Proof. The proof follows from the classification given in section A.2 and proposition A.2.2. □

In conclusion, the topology of the graphs we constructed reflects the *degrees of freedom* offered by a standard seventh chord, from a modal viewpoint. Moreover, dominant, minor seventh and half-diminished chords are commonly substituted in a jazz context. Think either about the equivalence between dominant and minor seven chords in a blues improvisation, or the strong relation between the pitch-class sets of a dominant chord, the half-diminished built on its major third and the minor seven on its perfect fifth (for instance $\{G, B, D, F\}$, $\{B, D, F, A\}$ and $\{D, F, A, C\}$, respectively).

B

Geometric characterisation of the chord space (proof).

Theorem B.0.1. *The space of chords $\mathbb{A}_n$ is a metric space, obtained by gluing the $(n-1)$-dimensional tetrahedral bases of a right n-dimensional prism via the equivalence relation induced by a cyclic permutation of the vertices.*

Proof. Let $G = \langle T^n, \mathcal{S}_n \rangle$ be the group of isometries such that $\mathbb{A}_n = \mathbb{R}^n/G$ and $x = (x_1, \dots, x_n)$ a point in $\mathbb{R}^n$. Let $\tau_i \in T^n$ be the translation $(x_1, \dots, x_i, \dots, x_n) = (x_1, \dots, x_i + 1, \dots, x_n)$ and $\sigma_{ij} \in \mathcal{S}_n$ the permutation swapping the ith and jth coordinates of x. The group G is isomorphic to $T^n \rtimes \mathcal{S}_n$, since $T^n \cap \mathcal{S}_n = \{e\}$ and T^n is normal in G.
The proof is structured as follows:

1. Study the subgroup of isometries $F \subset G$ that fixes the hyperplanes in $\mathbb{R}^n$, then observe that the elements of this group are reflections.
2. Define the fundamental domain for the action of G and show that it is the prism endowed with the structure described in the thesis of the theorem.

1. Let C_t be the hyperplane

$$\left\{ (x_1, \dots, x_n) \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} x_i = t \right\}$$

for $t \in \mathbb{R}$ and $F \subset G$ the subgroup of isometries fixing C_t. An element of F can be written as $\sigma\tau$, where $\sigma \in \mathcal{S}_n$ and $\tau = \tau_1^{e_1} \cdots \tau_n^{e_n}$, with $\sum_{i=1}^n e_i = 0$. We prove by inducing on $m = |\{ e_i \neq 0, \text{ for } i = 1, \dots, n \}|$ that τ is an element of the group generated by conjugate elements of $\mathcal{S}_n$. Observe that m has to be at least 2, indeed $\tau = \tau_1^{e_1} \cdots \tau_n^{e_n}$. The sum of the coordinates $\sum_i x_i = t$ is invariant under τ only if the sum of its exponents $\sum_i e_i = 0$, hence $m > 1$.

- $m = 2$: $\tau = \tau_i^k \tau_j^{-k} = \left(\tau_i \tau_j^{-1}\right)^k$, $\tau_i \tau_j^{-1} = \sigma_{ij} \tau_i^{-1} \sigma_{ij} \tau_i$.
- We assume the statement true for every integer up to m. For $(m+1)$ we have

$$\tau = \tau_{i_1}^{e_1} \cdots \tau_{i_{m+1}}^{e_{m+1}} = \left(\tau_{i_1}^{e_1} \cdots \tau_{i_m}^{e_m + e_{m+1}}\right) \left(\tau_{i_m}^{-e_{m+1}} \tau_{i_{m+1}}^{e_{m+1}}\right).$$

  By induction hypothesis both $\tau_{i_1}^{e_1} \cdots \tau_{i_m}^{e_m + e_{m+1}}$ and $\tau_{i_m}^{-e_{m+1}} \tau_{i_{m+1}}^{e_{m+1}}$ can be written as products of conjugated of elements in $\mathcal{S}_n$ and hence, so is τ.

The subgroup F is generated by elements of the form $\tau\sigma_{ij}\tau^{-1}$. Thus, it is necessary to study the transformations of the form $\tau\sigma_{ij}\tau^{-1}$. We distinguish two cases:

(i) $\tau = \tau_k^m$ and $k \notin \{i, j\}$, then $\tau_k^m \sigma_{ij} \tau_k^{-m} = \sigma_{ij}$.

(ii) $\tau = \tau_k^m$ and either $k = i$ or $k = j$. Assume $k = j$, then

$$\begin{aligned}
&\tau_j^m \sigma_{ij} \tau_j^{-m} (x_1, \ldots, x_i, \ldots, x_j, \ldots, x_n) \\
&\tau_j^m \sigma_{ij} (x_1, \ldots, x_i, \ldots, x_j - m, \ldots, x_n) \\
&\tau_j^m (x_1, \ldots, x_j - m, \ldots, x_i, \ldots, x_n) \\
&(x_1, \ldots, x_j - m, \ldots, x_i + m, \ldots, x_n).
\end{aligned}$$

Hence, $\tau_j^m \sigma_{ij} \tau_j^{-m}$ corresponds to the reflection with respect to the hyperplane $x_j - x_i = m$.

2. The fundamental domain for the action of G is given by $P = C \cap D$, where

$$C = \left\{ (x_1, \ldots, x_n) \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^{n} x_i \in [0, 1] \right\}$$

and

$$D = \{ (x_1, \ldots, x_n) \in \mathbb{R}^n \mid x_i \geqslant x_{i+1} \; \forall i \in \{ 1, \ldots, n-1 \}, \, x_1 \leqslant x_n + 1 \}.$$

Indeed, every point $x \in \mathbb{R}^n$ can be translated to some C_t, for $t \in [0, 1]$, hence, every $x \in \mathbb{R}^n$ is in relationship with a point of P, through the action of elements of G. The elements of F act as mirrors with respect to the hyperplane $x_i = x_j + m$, with $c \in \mathbb{Z}$. These constitute $\binom{n}{2}$ families of parallel hyperplanes, that decompose C_t in a union of $(n-1)$-dimensional simplices. Through these reflections, each point of C_t can be associated to one and only one point of the simplex $D \cap C_t$.

The points lying in the interior of P are not in relationship. Observe that the elements of G modifies the sum $\sum_i x_i$ of an integer value. If two points $x \in C_p$ and $y \in C_q$, with $p, q \in [0, 1]$ were in relationship, it should be $p - q \in \mathbb{Z}$, then, there are only two possible cases:

(i) $p = 0$ and $q = 1$ (or vice versa). In this case the points x and y belong to the basis of the prism and not to its interior.

(ii) $p = q$. Both points lie in the simplex $D \cap C_p$, hence they cannot be in relationship. Assume the contrary, then it exists $g \in F$ such that $g(x) = y$, i. e. y can be obtained by reflecting x with respect to a hyperplane of the form $x_i = x_j + c$, which is impossible for an aforementioned argument.

Finally, we study the possible relationships between the two bases $D \cap C_0$ and $D \cap C_1$. Let g be an element of G, $V = \{v_0, \ldots, v_k\}$ a set of points in $\mathbb{R}^n$ and $x = \sum_{i=0}^{k} \lambda_i v_i$, where $\sum_i \lambda_i = 1$, then

$$g(x) = g\left(\sum_{i=0}^{k} \lambda_i v_i \right) = \sum_{i=0}^{k} \lambda_i g(v_i),$$

these equalities show that we can study the possible relationships between the vertices of the two bases, and then extend them to their convex hulls. The vertices of $D \cap C_0$ are the origin v_0 and the points

$$v_k = \left(\underbrace{\frac{k}{n}, \ldots, \frac{k}{n}}_{(n-k) \text{ times}}, \underbrace{\frac{k-n}{n}, \ldots, \frac{k-n}{n}}_{k \text{ times}} \right) \quad \text{where } k \in \{\, 1, \ldots, n-1 \,\}.$$

The vertices of the basis $D \cap C_1$ have form $u_k = v_k + \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$ for $k \in \{0, \ldots, n-1\}$, i. e. they are obtained by a translation of the vertices v_k along the height-axis of the prism P. Note that every v_k is in relationship with the point $\left(\frac{k}{n}, \ldots, \frac{k}{n}\right)$ (it suffices to apply τ_j for $j \geqslant n-k$). For the same argument, every u_k is in relation with $\left(\frac{k+1}{n}, \ldots, \frac{k+1}{n}\right)$. Hence, for $k \in \{0, \ldots, n-2\}$, u_k is in relation with v_{k+1} and u_{n-1} is in relation with v_0.

□

C

Code

C.1 Boundary matrix reduction in persistence algorithm

```
import numpy as np

def low(column):
  ones = np.nonzero(column)
  # print ones[0]
  if not ones[0].any():
    low = None
  else:
    low = int(max(ones[0]))
  return low

def persistence(boundary):
  # Initialize the reduced matrix as a copy of
     boundary
  R = boundary
  # Check if the boundary matrix has non-admissible
     entries
  nonz = R[np.nonzero(R > 1)]
  if nonz.shape[1] > 0:
    print "check your boundary matrix!"
  else:
    columns = R[:R.shape[1]].T
    for i in range(len(columns)):
      for j in range(i):
        ci = np.array(columns[i])[0].tolist()
        cj = np.array(columns[j])[0].tolist()
        low_i = low(ci)
        low_j = low(cj)
        if low_i != None and low_j != None and low_i==
           low_j:
          new_col = np.mod(np.add(ci,cj),2)
          columns[i,:] = new_col
          persistence(R)
```

```
        return columns.transpose()
```

C.2 Three Dimensional Visualization of the Deformed *Tonnetz*

The following code add some comments to the JavaScript generating the web application hosted at `http://nami-lab.com/tonnetz/examples/deformed_tonnetz_int_sound_pers.html`. This implementation depends on the *Three.js* and *MIDI.js* libraries, which are free downloadable. We refer to the online version of the code for libraries' dependencies and to see how the script is embedded in a html page.

C.2.1 Tutorial

The web application allows to deform a 15×15 *Tonnetz* in two different ways, giving to the user the possibility to play chords and melodies on the keyboard using the keys of the first and second line as a piano keyboard, according to table C.1 or playing a piece of Music among the one listed in the menu *song* of the graphic user interface (gui). It is possible to orbit in the 3D scene using the mouse to zoom and rotate the *Tonnetz* and to move the whole mesh (right click and drag).

The skeletons of the simplicial complex can be hidden using the *show* commands of the gui, to better understand the positions of the vertices as a point cloud, or the geometry of the edges which could be hidden by the triangles in some configurations.

Once the *Tonnetz* has been deformed, the function *disp_pref* allows to compute a preferred set of pitch classes filtering the vertices of the *Tonnetz* on their relative height, considering the one higher than a certain threshold, depending on the height of the maximal peak among the vertices. The preferred pitch class configuration is then show as a point cloud on the planar *Tonnetz* in $z = 0$.

Table C.1: Pitches - Key association

Key	a	w	s	e	d	f	t	g	y	h	u	j	k	o	l	p
Pitch	*C*	*C♯*	*D*	*D♯*	*E*	*F*	*F♯*	*G*	*G♯*	*A*	*A♯*	*B*	*C*	*C♯*	*C*	*D♯*

```
<!-- Soundfont settings -->
<script>
  MIDI.loadPlugin({
      soundfontUrl: "./soundfont/",
      instrument: "acoustic_grand_piano"
  });
</script>
<script>
// Three js standard objects
  var group;
  var container, stats;
  var particlesData = [];
  var camera, scene, renderer;
```

```
// Tonnetz variables
  var positions,colors;
  var pointCloud;
  var particlePositions, vertices;
  var triangles, edges_helper, edges, mesh;
  var num_of_points_per_line = 10;
  var num_of_lines = 10;
  var num_of_triangles = 2 * (num_of_lines - 1) *  (
     num_of_points_per_line - 1 );
  var num_of_edges = 3 * num_of_triangles / 2 + (
     num_of_lines - 1) + num_of_points_per_line - 1;
  var faces = [];
  var num_of_particles = num_of_points_per_line *
     num_of_lines;
  var particleCount = num_of_particles;
  var shadow, mesh_s;
// keyboard listener
  var keyboard = new THREEx.KeyboardState();
  var indices_array = [], pitches_array = [];
// GUI
  var effectController = {
      showVertices: true,
      showEdges: true,
      showTriangles: true,
      showHelper:true,
      play_pause: true,
      stop: false,
      song : "1all_the_things_you_are.mid",
      reset_ton: false
  }
// MIDI playing settings
  var delay = 0; // play one note every quarter second
  var velocity = 127; // how hard the note hits
  var note, cur_playing=[];
  var player = MIDI.Player;
  var pitch = [], message;
// Songs
  var  songsToFiles ={
    "All the Things You Are": "3
       all_the_things_you_are_piano_solo_2_tema.mid",
    ...
   };

  init();
  animate();

  function initGUI() {
```

```
var gui = new dat.GUI();


gui.add( effectController, "showVertices" ).onChange
   ( function( value ) { pointCloud.visible = value;
   } );
gui.add( effectController, "showEdges" ).onChange(
   function( value ) { edges.visible = value; } );
gui.add( effectController, "showTriangles" ).
   onChange( function( value ) { mesh.visible = value
   ; } );
gui.add( effectController, "showHelper" ).onChange(
   function( value ) { edges_helper.visible = value;
   } );
gui.add( effectController, "disp_pref" ).onChange(
   function( value ) { if (value == true) {disp_pref
   ();  mesh.visible = false;}else{mesh.visible =
   true}} );
gui.add( effectController, "play_pause" ).onChange(
   function( value ) { if (value == true) {player.
   start(); }else{player.pause()}} );
gui.add( effectController, "stop" ).onChange(
   function( value ) {{player.stop(); }} );
gui.add( effectController, 'song' ,songsToFiles ).
   onChange( function( value ) {  player.stop();
   player.loadFile("midi/" + value, player.start); }
   );
gui.add( effectController, "reset_ton" ).onChange(
   function( value ) { undeform()} );

}

function init() {

initGUI();

container = document.getElementById( 'container' );
// camera settings
camera = new THREE.PerspectiveCamera( 45, window.
   innerWidth / window.innerHeight, 1, 4000 );
camera.position.z = 10;
camera.position.x = 0;
camera.position.y = -20;
// controls
controls = new THREE.OrbitControls( camera,
   container );
```

```
// scene and lights
scene = new THREE.Scene();
scene.fog = new THREE.Fog( 0x050505, 2000, 3500 );

var light1 = new THREE.DirectionalLight( 0xffffff,
   0.5 );
light1.position.set( 100, 100, 100 );
scene.add( light1 );

var light2 = new THREE.DirectionalLight( 0xffffff,
   1.5 );
light2.position.set( 0, -1, 0 );
scene.add( light2 );
// Tonnetz vertices
group = new THREE.Group();
scene.add( group );

positions = new Float32Array( num_of_particles * 3 )
   ;
colors = new Float32Array( num_of_particles * 3 );
      // vertices material
var pMaterial = new THREE.PointCloudMaterial( {
    color: 0xffffff,
    size: 3,
    blending: THREE.AdditiveBlending,
    transparent: true,
    sizeAttenuation: false
} );

particles = new THREE.BufferGeometry();
particlePositions = new Float32Array(
   num_of_particles * 3 );
// defining the triangle of the tonnetz as
   equilateral
var k = 0;
for ( var i = 0; i < num_of_lines; i++ ) {

    for ( var j = 0; j < num_of_points_per_line; j++
        ) {
        if ( i % 2 == 0 ){
            var x = j;
            var y = Math.sqrt(3)/2*i;
            var z = 0;
            pitches_array[k] = ((j*7)+i/2)%12;
        }
        else{
            var x = j+1/2;
```

```
            var y = Math.sqrt(3)/2*i;
            var z = 0;
            pitches_array[k] = ((j*7+4)+ (i-1)/2)
               %12;
        }
        particlePositions[ k * 3     ] = x;
        particlePositions[ k * 3 + 1 ] = y;
        particlePositions[ k * 3 + 2 ] = z;
        indices_array[k] = k;
        k++;
        }
}
// add position and index attribute to vertices
particles.addAttribute( 'position', new THREE.
   DynamicBufferAttribute( particlePositions, 3 ) );
particles.addAttribute( 'index', new THREE.
   BufferAttribute( new Uint16Array( indices_array ),
    1 ) );
// add vertices to the scene
pointCloud = new THREE.PointCloud( particles,
   pMaterial );
group.add( pointCloud );
// create the 2-skeleton (edges will be added later)
triangles = new THREE.Geometry();
for (var i = 0; i < num_of_particles; i++){
    var v = new THREE.Vector3(particlePositions[ i *
        3 ], particlePositions[ i * 3 + 1 ],
       particlePositions[ i * 3 + 2 ]);
    triangles.vertices.push(v);
}
var ind = 0;
for (var j = 0; j < num_of_lines-1; j++){
  var k = j * num_of_points_per_line
  for (var i = 0; i < num_of_points_per_line-1; i++)
     {
    if (j%2 ==0){
        triangles.faces.push(new THREE.Face3(k+i+0,k
           +i+1,k+i+num_of_points_per_line));
        var v1 = [k+i+0,k+i+1,k+i+
           num_of_points_per_line];
        triangles.faces.push(new THREE.Face3(k+i+
           num_of_points_per_line,k+i+
           num_of_points_per_line+1,k+i+1));
        var v2 = [k+i+num_of_points_per_line,k+i+
           num_of_points_per_line+1,k+i+1];
    }else{
        triangles.faces.push(new THREE.Face3(k+i+0,k
```

```
            +i+1,k+i+num_of_points_per_line+1));
          var v1 = [k+i+0,k+i+1,k+i+
            num_of_points_per_line+1];
          triangles.faces.push(new THREE.Face3(k+i+
            num_of_points_per_line,k+i+
            num_of_points_per_line+1,k+i));
          var v2 = [k+i+num_of_points_per_line,k+i+
            num_of_points_per_line+1,k+i];
      }
      faces[ind] = v1;
      faces[ind + 1] = v2;
      ind += 2;
    }
  }
 for ( var i = 0; i < triangles.faces.length; i ++ ) {
    triangles.faces[ i ].color.setHex( 1* 0xffffff );
  }
  // Set the faces' material
  var material = new THREE.MeshPhongMaterial( {
    color: 0xaaaaaa, specular: 0x00ffff, shininess:
       250,
    side: THREE.DoubleSide,vertexColors: THREE.
       FaceColors
  } );
  // Compute their normals
  triangles.computeFaceNormals();
  mesh = new THREE.Mesh( triangles, material );
  // State the geoemtry will be dynamically updated
  mesh.geometry.dynamic = true;
  // Add triangles
  group.add(mesh)
  // generate the planar Tonnetz behind the deformed
     one
  edges_helper = new THREE.EdgesHelper( mesh, 0x00ff00
      );
  group.add(edges_helper);
  // edge material: the attribute wireframe allows to
     generate the edges directly from the triangles
  var ematerial = new THREE.MeshBasicMaterial( {
    color: 0x00ff00, specular: 0x00ffff, shininess:
       250,
    side: THREE.DoubleSide, wireframe:true
  });
  // Add edges
  edges = new THREE.Mesh( triangles, ematerial  );
  group.add(edges);
  // Asociation among keyboard character and sounds
```

```
var playing_char = ["a","w","s","e","d","f","t","g",
   "y","h","u","j","k","o","l","p","Ăš"];
for (var i = 0; i < playing_char.length; i++){
   cur_playing[playing_char[i]] = 0;
}
// MIDI player settings
      window.onload = function (){
  MIDI.loadPlugin(function ()
  {
      player.timeWarp = 1.0;
      player.addListener(function(data)
      {
          message = data.message;
          // If a pitch plays return it
          if (message === 144){
              pitch.push(data.note);
              }else{pitch = [];}
      });
  });
      }
      // renderer
renderer = new THREE.WebGLRenderer( { antialias:
   true, alpha:true} );
renderer.setPixelRatio( window.devicePixelRatio );
renderer.setSize( window.innerWidth, window.
   innerHeight );
renderer.gammaInput = true;
renderer.gammaOutput = true;

container.appendChild( renderer.domElement );
// fps stats
stats = new Stats();
stats.domElement.style.position = 'absolute';
stats.domElement.style.top = '0px';
container.appendChild( stats.domElement );

window.addEventListener( 'resize', onWindowResize,
   false );
}

function onWindowResize() {
  camera.aspect = window.innerWidth / window.
     innerHeight;
  camera.updateProjectionMatrix();
  renderer.setSize( window.innerWidth, window.
     innerHeight );
}
```

```
// MIDI noteOn and noteOff
function play(note) {
  MIDI.noteOn(0, note, velocity, delay);
}
 function stop(note) {
  MIDI.noteOff(0, note, delay);
}
function animate() {
  requestAnimationFrame( animate );
  stats.update();
  render();
}
// Tonnetz deformation through keyboard: play a
   pitch and update the geometry of the simplicial
   complex
function key(character, note){
  var c = 0.002;
 if( keyboard.pressed(character) ){
   for (var i = 0; i < pitches_array.length; i++){
       if (pitches_array[i]==note%12){
          particlePositions[i*3+2] += c;
          mesh.geometry.vertices[i].z += c;
       }
  }
  if( cur_playing[character] == 0){
    cur_playing[character] = 1;
    play(note);
   }
 }
else if (cur_playing[character] == 1){
  cur_playing[character] = 0;
  stop(note);
 }
}
// Bring the tonnetz to its planar shape
function undeform(){
var c = 0.002;
for (var i = 0; i < pitches_array.length; i++){
        particlePositions[i*3+2] = 0;
        mesh.geometry.vertices[i].z = 0;
     }
}
// Deformation induced by the MIDI player
function playerdef(pitch){
    c = 0.002
    if( message === 144){
         for (var i = 0; i < pitches_array.length; i
```

```
                ++){
                  if (pitches_array[i]==pitch%12){
                     particlePositions[i*3+2] += c;
                     mesh.geometry.vertices[i].z += c;
                  }
            }
      }
}

function disp_pref(){
  //get the values of the vertices on the mesh
  var sk_0 = mesh.geometry.vertices;
  //get the faces of the mesh
  var sk_2 = faces;
  var height = [];
  var h_val = [];
          var sort_indices = [];
  var pref_pitches = [];
  //read heights of pitches and create an indexed
     array
  for (var i = 0; i < 12; i++){

    height[i] = [ i , sk_0[i].z ];
    h_val[i] =  sk_0[i].z;
  }
  var fifths = [[0,'C'] ,[7,'G'] ,[2,'D'] ,[9,'A']
     ,[4,'E'] ,[11,'B'] ,[6,'F#'] ,[1,'C#'] ,[8,'Ab']
      ,[3,'Eb'] ,[10,'Bb'] ,[5,'F'] ];
  var aver = eval(h_val.join('+'))/12;
  //sort the vertices using their height and save
     the
  //corresponding permutation of the indices in the
     array
  //sort_indices
  height.sort(function(x,y){return x[1] - y[1] })

  for (var i = 0; i < height.length; i++){

    sort_indices[i] = height[i][0];
    pref_pitches[i] = fifths[height[i][0]];
  }
  //
  var soil_pref = [];
  var s = 0;
  var threshold = 2;
  var max_h = height[height.length-1][1];
  soil_pref[0] = height[height.length-1];
```

```
//select preferred pitches (depends on threshold)
for (var i = height.length-1; i > 0; i-- ){

  if (height[i][1] > max_h/threshold){

    soil_pref[s] = height[i];
    s++;
  }
}
//sort preferred pitches remembering their indices
soil_pref.sort(function(x,y){return x[1] - y[1] })
console.log(soil_pref)
var sort_soil_indices = [];
var pref_soil_pitches = [];

for (var i = 0; i < soil_pref.length; i++){

  sort_soil_indices[i] = fifths[soil_pref[i
     ][0]][0];
  pref_soil_pitches[i] = fifths[soil_pref[i
     ][0]][1];
}
// preferred pitches and their value in Z/12Z are
   displayed in console
console.log(sort_soil_indices , pref_soil_pitches)
var material = new THREE.PointCloudMaterial( {
  color: 0x000000,
  size: 5,
  transparent: false,
  sizeAttenuation: false
} );
//generate the geometry associated to preferred
   pitches
shadow =  new THREE.Geometry();
//draw it
for (var i = 0; i < sort_soil_indices.length; i++)
   {
  var pref_pitch = sort_soil_indices[i];
  for (var j = 0; j < pitches_array.length; j ++){
    if (pitches_array[j] == pref_pitch){
                var v = new THREE.Vector3(
                   particlePositions[ j * 3 ],
                   particlePositions[ j * 3 + 1
                   ],0);
      shadow.vertices.push(v);
    }
  }
```

```
    }
    //generate the point cloud of preferred pitches
       and add it to the group
    mesh_s = new THREE.PointCloud( shadow, material );
            mesh_s.geometry.dynamic = true;
            group.add(mesh_s)
  }

  function render() {
    key("a", 60);
    ...
  // Receive pitches from the player and deform the
     tonnetz
  if (message === 144){
        for (var i = 0; i<pitch.length; i++){
        playerdef(pitch[i])
        }
  }
  // Need Update declaration
    pointCloud.geometry.attributes.position.
       needsUpdate = true;
    mesh.geometry.verticesNeedUpdate = true;
    renderer.render( scene, camera );
}
</script>
```

C.3 Persistent homology computation

The following code described the computation of the persistence diagrams of the torus *Tonnetz* through the filtration induced by the height function defined on its planar covering.

```
import os
import numpy as np
from music21 import *
from dionysus import *
from os import listdir
from os.path import isfile, join
import matplotlib.pyplot as plt
import matplotlib as mpl
import csv
from mpl_toolkits.mplot3d import axes3d
import mpl_toolkits.mplot3d.axes3d as p3


class Tonnetz:
    def __init__(self, left, lowRight, upRight):
```

```
        self.left = left
        self.lowRight = lowRight
        self.upRight = upRight
        self.weights = {}
        self.triangles = []

    notes = ['C', 'C#', 'D', 'Eb', 'E', 'F', 'F#', 'G'
       , 'G#', 'A', 'Bb', 'B']


    def computeDataFrom(self, piece, considerDurations
        = True):

        def extractFromChord(c):
            values = []
            value = None
            try:
                value = c.pitchClass
            except AttributeError:
                pass # do not try others

            if value is not None:
                values.append(value)

            if values == []:
                for p in c.pitches:
                    # try to get get values from pitch
                        first, then chord
                    value = None
                    try:
                        value = p.pitchClass
                    except AttributeError:
                        break # do not try others
                    if value is not None:
                        values.append(value)
            return values


        self.weights = {}
        for i in range(12):
            self.weights[i] = 0

        flat = piece.flat.getElementsByClass([note.
           Note, chord.Chord])
        for obj in flat:
            if 'Chord' in obj.classes:
                values = extractFromChord(obj)
```

```
            else: # simulate a list
                values = [obj.pitch.pitchClass]

            for i, value in enumerate(values):
                if considerDurations:
                    self.weights[value] += obj.
                       duration.quarterLength
                else:
                    self.weights[value] += 1

    def loadDataFromTxt(self, filename):

        with open(filename, "r") as text_file:
            lines = text_file.readlines()

        self.weights = {}
        for i in range(12):
            self.weights[i] = float(lines[i].replace("
               \n", ""))

    def draw2D(self, title = '', showValues = True):
        nCols = 4
        nRows = 4
        x = []
        y = []
        z = []
        for i in range(nRows):
            for j in range(nCols):
                x.append(1 + 2 * j + i)
                y.append(nRows + 1 -  2 * i)
        self.triangles = [[0,4,1],[1,5,2],[2,6,3],
        [1,4,5],[2,5,6],[3,6,7],
        [4,8,5],[5,9,6],[6,10,7],
        [5,8,9],[6,9,10],[7,10,11]]
        triangles = np.asarray(self.triangles)
        print triangles

        plt.figure()
        plt.gca().set_aspect('equal')
        plt.triplot(x, y, triangles, "o--", color='
           blue')
        plt.title(title)
        plt.axis([0,10,0,6]) # margins
        plt.axis('off')
        matrix = [[0 for i in range(10)] for j in
           range(10)]
        noteIndex = self.left % 12 # first "previous"
```

```
            point
        for i in range(3): # this sequence depends on
            x and y creation order
            for j in range(4):
                noteIndex = (noteIndex - self.left) %
                    12
                xx = (1 + 2 * j) + i
                yy = 5 - 2 * i
                try:
                    matrix[yy][xx] = self.weights[
                        noteIndex]
                except KeyError:
                    print 'Error: Please call
                        computeDataFrom before calling
                        draw2D'
                    return
                label = self.notes[noteIndex]
                if showValues:
                    label += ' (' + str(self.weights[
                        noteIndex]) + ') ' + "[" + str
                        (4*i+j) + "]"
                plt.annotate(label,  xy=(xx, yy),
                    xytext=(8, 3), textcoords='offset
                    points', color='blue')
            noteIndex = (noteIndex + self.lowRight + 4
                * self.left) % 12

        cmap = mpl.colors.LinearSegmentedColormap.
            from_list('my_cmap', ['white','red'], 256)
        cmap._init()
        alphas = np.linspace(0, 0.8, cmap.N+3)
        cmap._lut[:,-1] = alphas

        plt.imshow(matrix, alpha=1, cmap=cmap,
            interpolation='blackman')
        plt.colorbar()
        plt.show()


    def draw3D(self, title = '', showValues = True,
        saveFileName = ""):
        nCols = 4
        nRows = 4
        x = []
        y = []
        z = []
        for i in range(nRows):
```

```
            for j in range(nCols):
                x.append(1 + 2 * j + i)
                y.append(nRows + 1 -  2 * i)
        self.triangles = [[0,4,1],[1,5,2],[2,6,3],
                          [1,4,5],[2,5,6],[3,6,7],
                          [4,8,5],[5,9,6],[6,10,7],
                          [5,8,9],[6,9,10],[7,10,11]]
        triangles = np.asarray(self.triangles)

        fig = plt.figure()
        ax = fig.gca(projection='3d')
        ax.view_init(elev=40., azim=268)

        noteIndices = []
        noteIndex = self.left % 12 # first "previous"
           point
        for i in range(3): # this sequence depends on
           x and y creation order
            for j in range(4):
                noteIndex = (noteIndex - self.left) %
                   12
                noteIndices.append(noteIndex)
                z.append(self.weights[noteIndex])
            noteIndex = (noteIndex + self.lowRight + 4
                * self.left) % 12

        if (showValues):
            [ax.text(x[i], y[i], z[i], self.notes[
               noteIndices[i]] + ":" + str("{0:.2f}".
               format(z[i]))) for i in range(12)]
        ax.plot_trisurf(x, y, z, triangles=triangles,
           cmap='Blues', linewidth=0.1, shade=True)
        if saveFileName != "":
            fig.savefig(saveFileName + '.pdf')
        else:
            plt.show()


    def getTonnetz3D(self):
        nCols = 4
        nRows = 4
        x = []
        y = []
        z = []
        for i in range(nRows):
            for j in range(nCols):
                x.append(1 + 2 * j + i)
```

```
                y.append(nRows - i)
        self.triangles =
           [[0,4,1],[1,5,2],[2,6,3],[3,7,0],
                        [1,4,5],[2,5,6],[3,6,7],[0,7,4],

                        [4,8,5],[5,9,6],[6,10,7],[7,11,4],

                        [5,8,9],[6,9,10],[7,10,11],[4,11,8],

                        [8,0,9],[9,1,10],[10,2,11],[11,3,8],

                        [9,0,1],[10,1,2],[11,2,3],[8,3,0]]

        triangles = np.asarray(self.triangles)
#         print triangles

        noteIndex = self.left % 12 # first "previous"
           point
        for i in range(3): # this sequence depends on
           x and y creation order
            for j in range(4):
                noteIndex = (noteIndex - self.left) %
                   12
                z.append(self.weights[noteIndex])
            noteIndex = (noteIndex + self.lowRight + 4
                * self.left) % 12

        return [x,y,z]


def max_vertex(s, vertices):
    values = [vertices[v] for v in s.vertices]
    if len(values) > 0:
        return max(values)
    else:
        return 0

def max_vertex_cmp(s1, s2, vertices):
    m1 = max_vertex(s1, vertices)
    m2 = max_vertex(s2, vertices)
    return cmp(m1, m2) or cmp(s1.dimension(), s2.
       dimension())

#-----------------------------------------#
#------------------ MAIN -----------------#
#-----------------------------------------#
```

```
midi_dir = "sibeliusnodrums"
distance = 1
holes_under = 1 # 0 to disable, 1 to remove z below
   1/10, 2 to remove z below 2/10...

# Name of CSV files
weights_csv_name = "weights_" + midi_dir + ".csv"
distances_csv_name = "distances_" + midi_dir + "_z.csv
   "


# Outputting weigths to CSV
header=['Name','Coords']
with open(os.path.join(weights_csv_name), 'wb') as
   csvfile:
    writer = csv.DictWriter(csvfile, fieldnames =
       header, delimiter = ';')
    writer.writeheader()
myfile = open(weights_csv_name, 'ab')
wr = csv.writer(myfile, delimiter=";")

# --- MIDI ---
dir_path = "./" + midi_dir + "/"
paths_list = [dir_path + f for f in listdir(dir_path)
   if isfile(join(dir_path, f)) ]
# --- end MIDI ---

# --- Music21 Corpus ---
#coreCorpus = corpus.CoreCorpus()
#paths_list = [path for path in coreCorpus.getPaths()]
# --- end Music21 Corpus ---

dgms = []
names = []
n = 1

print "Writing weights..."
for path in paths_list:#[0:5]:

    #   --- txt ---
    name = path.replace(dir_path, "")
    if not name.endswith(".txt"):
        continue
    print "iteration %d on element %s" % (n, name)
    path = path.replace(".txt", "")
    name = name.replace(".txt", "")
    n += 1
    #   --- end txt ---
```

```
    print
    names.append(name)
    n += 1
    tonn = Tonnetz(3,4,5)
#     tonn.computeDataFrom(piece, considerDurations =
    True)
    tonn.loadDataFromTxt(path + ".txt")
    tonn.draw3D(saveFileName = "") #path)
    row = name, tonn.getTonnetz3D()
    wr.writerow(row)

    result = row[1]
    maxZ = max(result[2])

    points = []
    vertices = []
    simplices = Filtration()
    for i in range(12):
        vertices.append(10 * float(result[2][i]) /
           maxZ)
    for i in range(24):
        if (vertices[tonn.triangles[i][0]] >
           holes_under and vertices[tonn.triangles[i
           ][1]] > holes_under and vertices[tonn.
           triangles[i][2]] > holes_under):
            simplices.append(Simplex([tonn.triangles[i
               ][0]]))
            simplices.append(Simplex([tonn.triangles[i
               ][1]]))
            simplices.append(Simplex([tonn.triangles[i
               ][2]]))
            simplices.append(Simplex([tonn.triangles[i
               ][0], tonn.triangles[i][1]]))
            simplices.append(Simplex([tonn.triangles[i
               ][1], tonn.triangles[i][2]]))
            simplices.append(Simplex([tonn.triangles[i
               ][2], tonn.triangles[i][0]]))
            simplices.append(Simplex(tonn.triangles[i
               ]))

    simplices.sort(lambda x,y: max_vertex_cmp(x,y,
       vertices))
    print "Complex in the filtration order: ", ', '.
       join((str(s) for s in simplices))
    print
    p = StaticPersistence(simplices)
```

```
    p.pair_simplices()

    # Output the persistence diagram
    smap = p.make_simplex_map(simplices)

    dgm = init_diagrams(p, simplices, lambda s: max(
       vertices[v] for v in s.vertices))
    print "Diagram:"
    print dgm
#     drawPersistenceDiagram(dgm)
    print
    dgms.append(dgm)

print "Writing distances..."
header=['Name1', 'Name2', 'Distance']
with open(os.path.join(distances_csv_name), 'wb') as
   csvfile:
    writer = csv.DictWriter(csvfile, fieldnames =
       header, delimiter = ';')
    writer.writeheader()

with open(distances_csv_name, 'ab') as csvfile:
    wr = csv.writer(csvfile, delimiter=";")

    for i in range(len(dgms)):
        for j in range(i+1, len(dgms)):
            if i % 100 == 0 and j == i+1:
                print i, "/", len(dgms)
            try:
                bott_dist = bottleneck_distance(dgms[i
                   ][distance], dgms[j][distance])
            except:
                bott_dist = 'undefined'

            result = names[i], names[j], bott_dist
            print result
            wr.writerow(result)

print "Finished."
```

C.4 Persistent time series - pairwise bottleneck distance

```
import os
import sys
```

```
import numpy as np
from music21 import *
from dionysus import *
from os import listdir
from os.path import isfile, join
import matplotlib.pyplot as plt
import matplotlib as mpl
import csv
from mpl_toolkits.mplot3d import axes3d
import mpl_toolkits.mplot3d.axes3d as p3


class Tonnetz:
    def __init__(self, left, lowRight, upRight):
        self.left = left
        self.lowRight = lowRight
        self.upRight = upRight
        self.weights = {}
        self.triangles = []
        self.numberOfFragments = 0

    notes = ['C', 'C#', 'D', 'Eb', 'E', 'F', 'F#', 'G'
       , 'G#', 'A', 'Bb', 'B']

    def computeDataFrom(self, piece, considerDurations
        = True, tsNumberOfMeasures = 1):

        def extractFromChord(c):
            values = []
            value = None
            try:
                value = c.pitchClass
            except AttributeError:
                pass # do not try others

            if value is not None:
                values.append(value)

            if values == []:
                for p in c.pitches:
                    # try to get get values from pitch
                        first, then chord
                    value = None
                    try:
                        value = p.pitchClass
                    except AttributeError:
                        break # do not try others
```

```
                    if value is not None:
                        values.append(value)
            return values


        self.weights = {}

        flat = piece.flat.getElementsByClass([note.
           Note, chord.Chord])
        fragmentNum = 0;
        self.weights[0] = {}
        for i in range(12):
            self.weights[0][i] = 0
        for obj in flat:
            fragmentNum = int(obj.offset / (4 *
               tsNumberOfMeasures))
            if (len(self.weights) <= fragmentNum):
                for i in range(len(self.weights),
                   fragmentNum + 1):
                    self.weights[i] = {}
                    for j in range(12):
                        self.weights[i][j] = self.
                           weights[i - 1][j]

            if 'Chord' in obj.classes:
                values = extractFromChord(obj)
            else: # simulate a list
                values = [obj.pitch.pitchClass]

            for i, value in enumerate(values):
                if considerDurations:
                    self.weights[fragmentNum][value]
                       += obj.duration.quarterLength
                else:
                    self.weights[fragmentNum][value]
                       += 1
        self.numberOfFragments = fragmentNum + 1


    def getTonnetz3D(self, fragmentNum = 0):
        nCols = 4
        nRows = 3
        x = []
        y = []
        z = []
        for i in range(nRows):
            for j in range(nCols):
```

```
                x.append(1 + 2 * j + i)
                y.append(nRows - i)
        self.triangles =
           [[0,4,1],[1,5,2],[2,6,3],[3,7,0],[1,4,5],
                [2,5,6],[3,6,7],[0,7,4],[4,8,5],[5,9,6],

                [6,10,7],[7,11,4],[5,8,9],[6,9,10],
                [7,10,11],[4,11,8],[8,0,9],[9,1,10],
                [10,2,11],[11,3,8],[9,0,1],[10,1,2],
                [11,2,3],[8,3,0]]
        triangles = np.asarray(self.triangles)
#         print triangles

        noteIndex = self.left % 12 # first "previous"
           point
        for i in range(3): # this sequence depends on
           x and y creation order
            for j in range(4):
                noteIndex = (noteIndex - self.left) %
                   12
                z.append(self.weights[fragmentNum][
                   noteIndex])
            noteIndex = (noteIndex + self.lowRight + 4
                * self.left) % 12

        return [x,y,z]


def drawPersistenceDiagram(dgm, name):
    for dim in range(2):
        plt.figure()
        plt.gca().set_aspect('equal')
        plt.title("Persistence Diagram")
        maximum = 0
        try:
            points = [i for i in dgm[dim]]
        except:
            continue
        x = []
        y = []
        for point in points:
            if (np.isinf(point[1])):
                x.append(point[0])
                y.append(point[0])
                plt.plot([point[0], point[0]], [point
                   [0], 100], 'r')
                maximum = max(maximum, point[0])
```

```
            else:
                x.append(point[0])
                y.append(point[1])
                maximum = max(maximum, point[0], point
                    [1])
        plt.axis([0, maximum+1, 0, maximum+1]) #
            margins
        plt.plot(x, y, 'ro')
#         plt.show()
        plt.savefig(name + '_diagram' + str(dim) + '.
            pdf')
        plt.close()


def max_vertex(s, vertices):
    values = [vertices[v] for v in s.vertices]
    if len(values) > 0:
        return max(values)
    else:
        return 0

def max_vertex_cmp(s1, s2, vertices):
    m1 = max_vertex(s1, vertices)
    m2 = max_vertex(s2, vertices)
    return cmp(m1, m2) or cmp(s1.dimension(), s2.
        dimension())

#-----------------------------------------------#
#------------------ MAIN -----------------------#
#-----------------------------------------------#

midi_dirs = ["midi_new_classic_mini", "
    midi_new_jazz_mini", "midi_new_pop_mini", "
    midi_new_pop_minimini"]
tsMeasures = 2 # Number of measures (in time series
    case)

#
    ------------------------------------------------------------------

holes_under = 0 # 0 to disable, 1 to remove z below
    1/10, 2 to remove z below 2/10...

for midi_dir in midi_dirs:
    print "Processing dir: ", midi_dir

    # --- MIDI ---
```

```
dir_path = "./" + midi_dir + "/"
paths_list = [dir_path + f for f in listdir(
   dir_path) if isfile(join(dir_path, f)) ]
# --- end MIDI ---

# --- Music21 Corpus ---
#coreCorpus = corpus.CoreCorpus()
#paths_list = [path for path in coreCorpus.
   getPaths()]
# --- end Music21 Corpus ---

dgms = {}
names = []
n = 1

for path in paths_list:#[0:5]:

#    #--- MIDI ---
    name = path.replace(dir_path, "")
    try:
        piece = converter.parse(path)
    except:
        continue
#    #--- end MIDI ---

    names.append(name)
    dgms[len(dgms)] = []
    n += 1
    tonn = Tonnetz(3,4,5)
    tonn.computeDataFrom(piece, considerDurations
       = True, tsNumberOfMeasures = tsMeasures)

    maxZ = max(tonn.getTonnetz3D(fragmentNum =
       tonn.numberOfFragments - 1)[2])

    print name, "Number of fragments: ", tonn.
       numberOfFragments
    for j in range(tonn.numberOfFragments):
        row = name, tonn.getTonnetz3D(fragmentNum
           = j)
        result = row[1]

        points = []
        vertices = []
        simplices = Filtration()
        for i in range(12):
```

```
                vertices.append(10 * float(result[2][i
                    ]) / maxZ)
            for i in range(24):
                if (vertices[tonn.triangles[i][0]] >
                    holes_under and vertices[tonn.
                    triangles[i][1]] > holes_under and
                    vertices[tonn.triangles[i][2]] >
                    holes_under):
                    simplices.append(Simplex([tonn.
                        triangles[i][0]]))
                    simplices.append(Simplex([tonn.
                        triangles[i][1]]))
                    simplices.append(Simplex([tonn.
                        triangles[i][2]]))
                    simplices.append(Simplex([tonn.
                        triangles[i][0], tonn.triangles[
                        i][1]]))
                    simplices.append(Simplex([tonn.
                        triangles[i][1], tonn.triangles[
                        i][2]]))
                    simplices.append(Simplex([tonn.
                        triangles[i][2], tonn.triangles[
                        i][0]]))
                    simplices.append(Simplex(tonn.
                        triangles[i]))

            simplices.sort(lambda x,y: max_vertex_cmp(
                x,y,vertices))
            p = StaticPersistence(simplices)
            p.pair_simplices()

            smap = p.make_simplex_map(simplices)
            dgm = init_diagrams(p, simplices, lambda s
                : max(vertices[v] for v in s.vertices))
            drawPersistenceDiagram(dgm, "./" +
                midi_dir + "/" + name + str(tsMeasures)
                + "_no_holes_frag" + str(j))
            dgms[len(dgms) - 1].append(dgm)


    distance = 0
    # Name of CSV files
    distances_csv_name = "./
        distances_no_holes_dim0_torus_ts_" + str(
        tsMeasures) + "meas/distances_" + midi_dir + "_z
        .csv"
```

```
print "Writing distances..."
header=['Name1', 'Name2', 'Distance']
with open(os.path.join(distances_csv_name), 'wb')
   as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames =
       header, delimiter = ';')
    writer.writeheader()

with open(distances_csv_name, 'ab') as csvfile:
    wr = csv.writer(csvfile, delimiter=";")

    for i in range(len(dgms)):
        print names[i], len(dgms[i])
        for j in range(i+1, len(dgms)):
            # Pairwise distances
            distances = []
            for ii in range(len(dgms[i])):
                for jj in range(len(dgms[j])):
                    try:
                        bott_dist =
                           bottleneck_distance(dgms
                           [i][ii][distance], dgms[
                           j][jj][distance])
                    except:
                        bott_dist = 'undefined'
                    distances.append(bott_dist)

            result = names[i], names[j], distances
            wr.writerow(result)

distance = 1
# Name of CSV files
distances_csv_name = "./
   distances_no_holes_dim1_torus_ts_" + str(
   tsMeasures) + "meas/distances_" + midi_dir + "_z
   .csv"

print "Writing distances..."
header=['Name1', 'Name2', 'Distance']
with open(os.path.join(distances_csv_name), 'wb')
   as csvfile:
    writer = csv.DictWriter(csvfile, fieldnames =
       header, delimiter = ';')
    writer.writeheader()

with open(distances_csv_name, 'ab') as csvfile:
    wr = csv.writer(csvfile, delimiter=";")
```

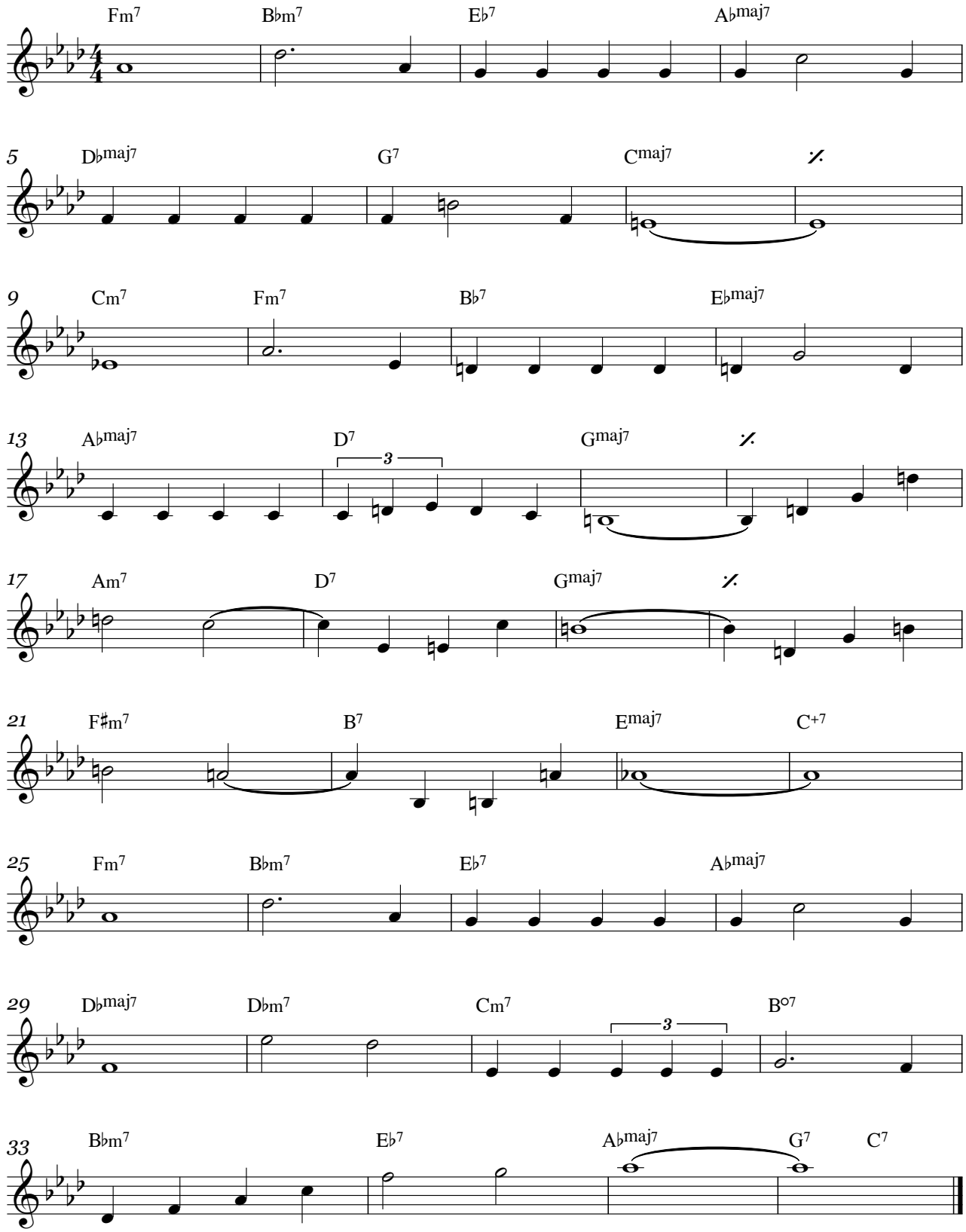
```
        for i in range(len(dgms)):
            for j in range(i+1, len(dgms)):
                # Pairwise distances
                distances = []
                for ii in range(len(dgms[i])):
                    for jj in range(len(dgms[j])):
                        try:
                            bott_dist =
                               bottleneck_distance(dgms
                               [i][ii][distance], dgms[
                               j][jj][distance])
                        except:
                            bott_dist = 'undefined'
                        distances.append(bott_dist)

                result = names[i], names[j], distances
                wr.writerow(result)

print "Finished."
```

D

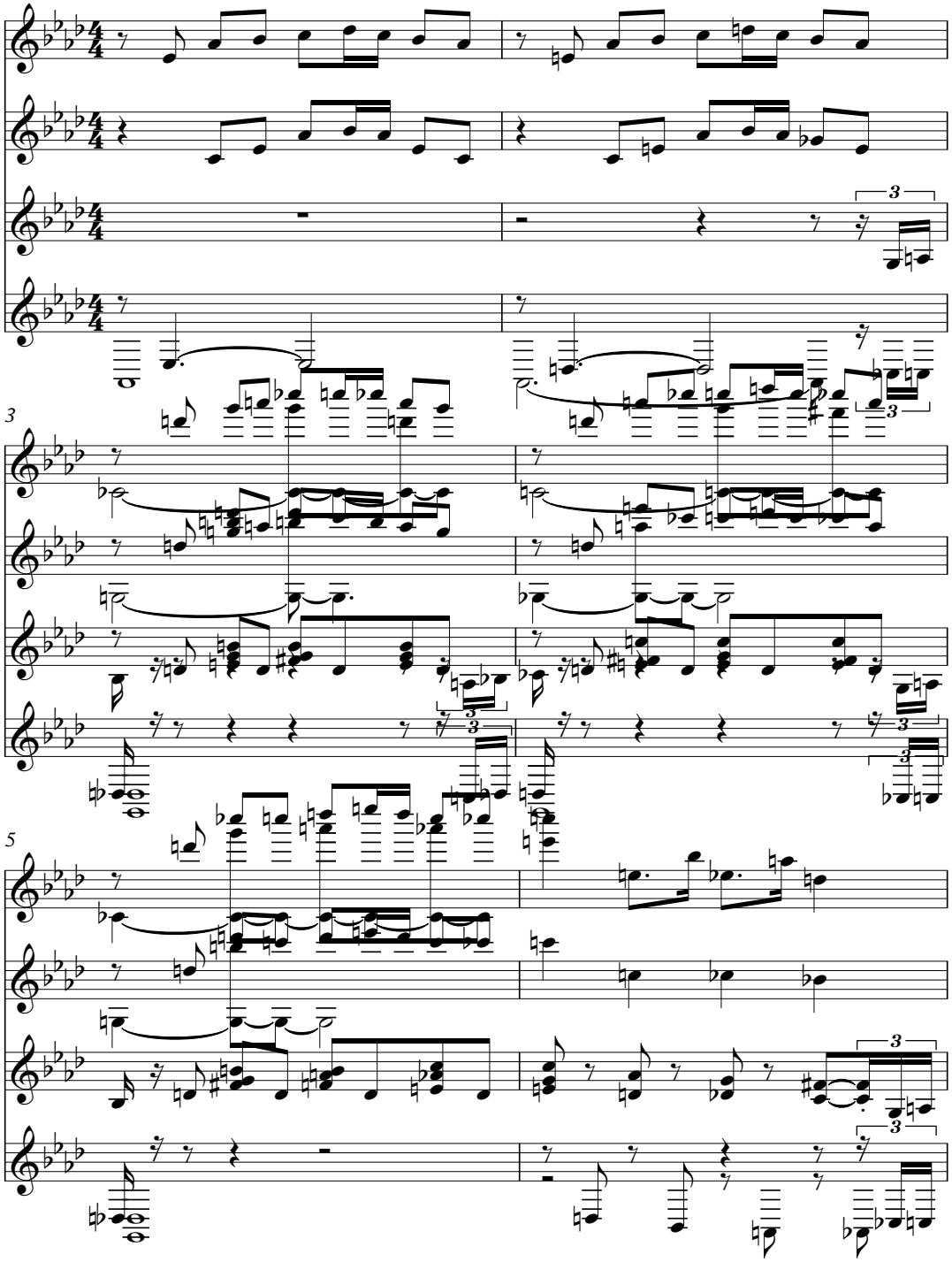
Scores

Here we report the minimal collection of scores that we believe essential for the understandability of the musical applications contained in this work. The author transcribed part of the pieces contained in this chapter, while both the two versions of *Caravan* and the fragments ([1]interlude and [3]intro) of *All the Things You Are* correspond to the MIDI files freely downloadable at `http://www.midiworld.com/` and `http://midkar.com/`.

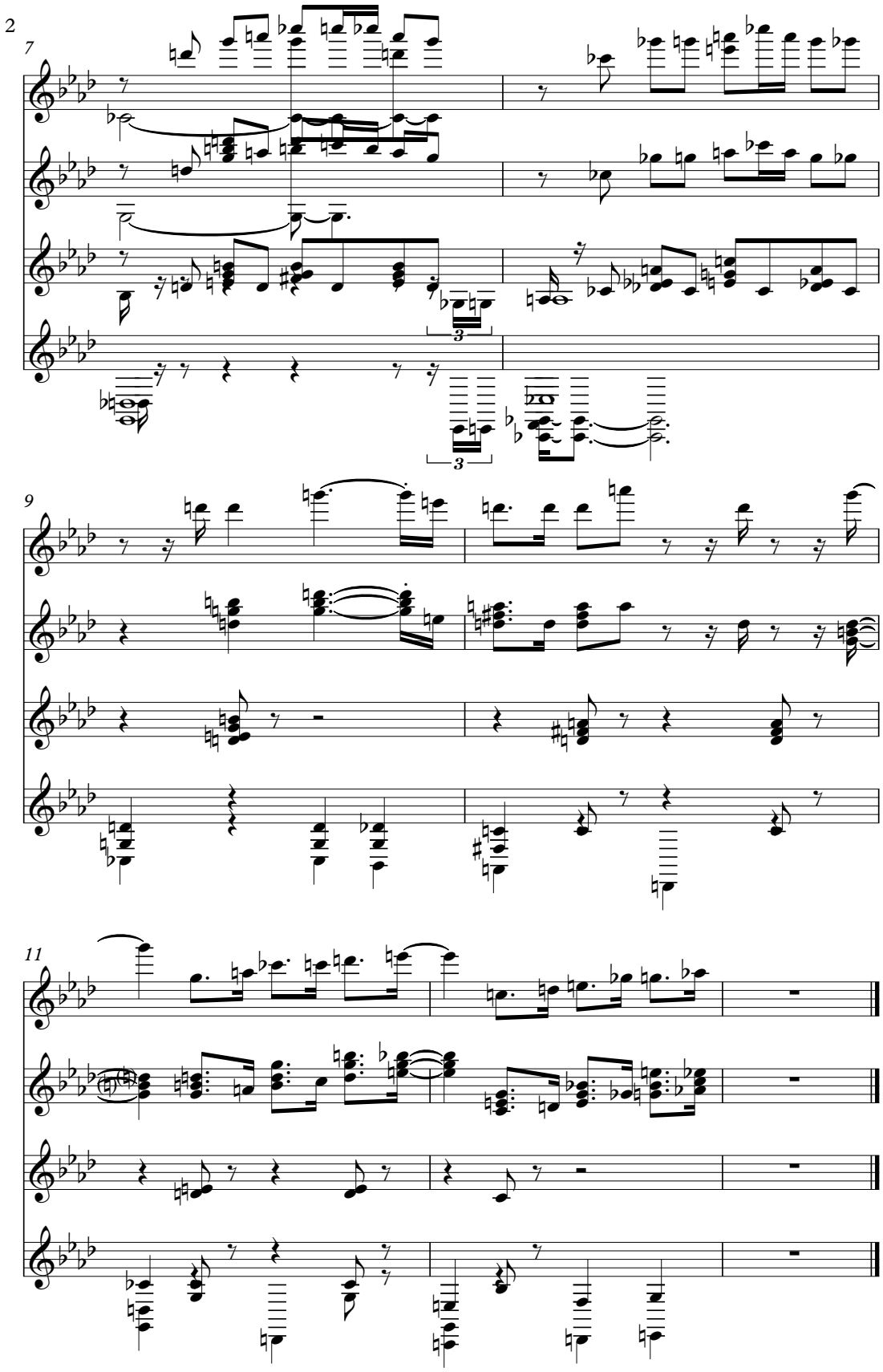
All the Things You Are - *Jerome Kern*



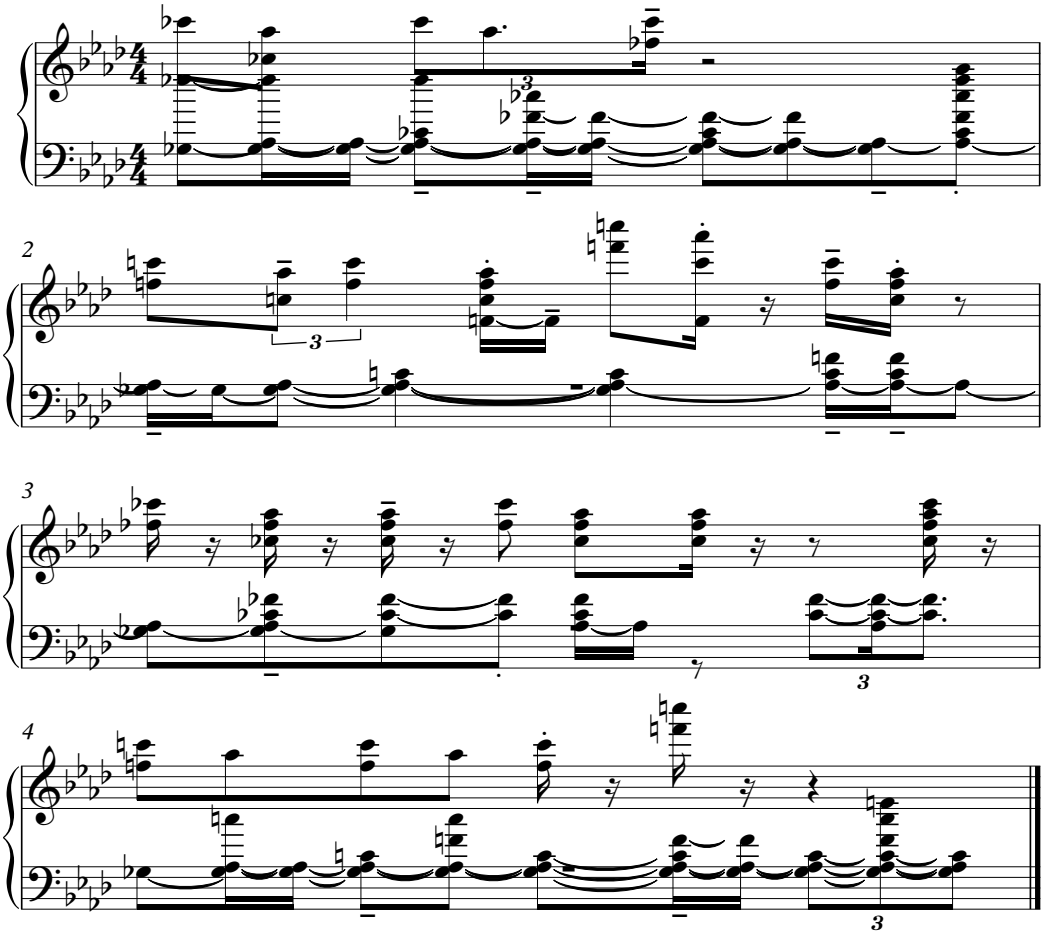
[1]interlude



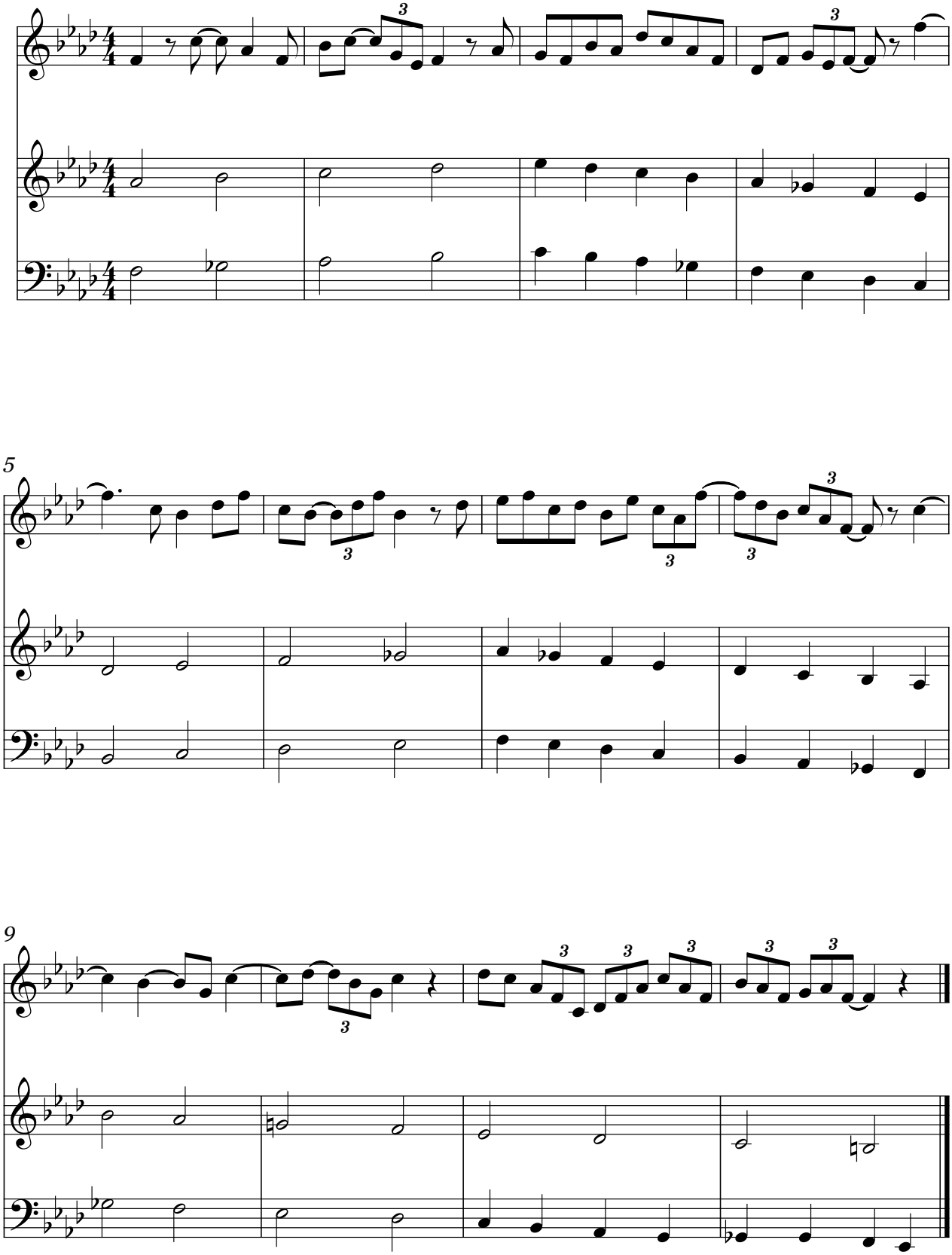
[1]interlude



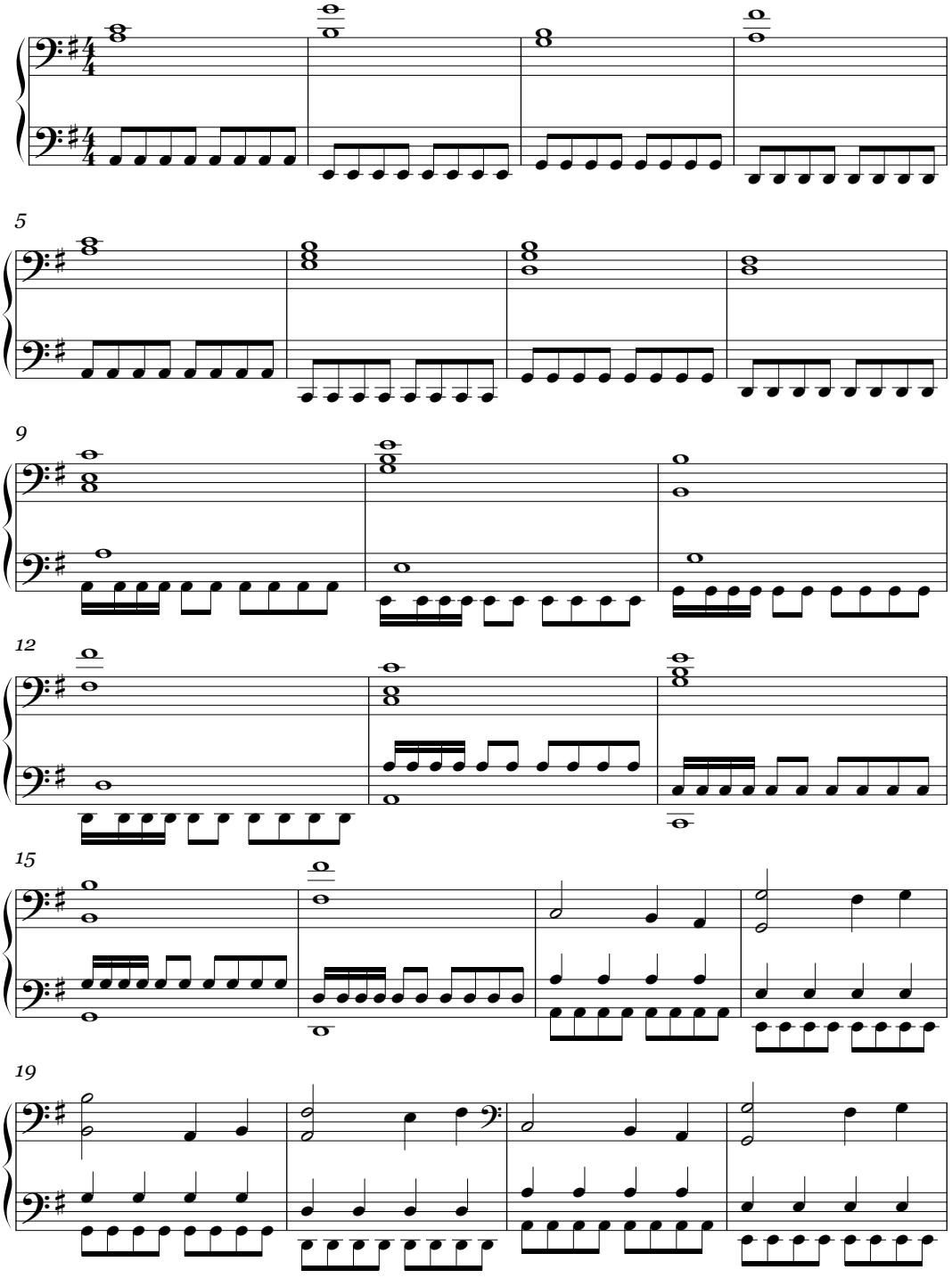
[3]intro



Interplay - *Bill Evans*

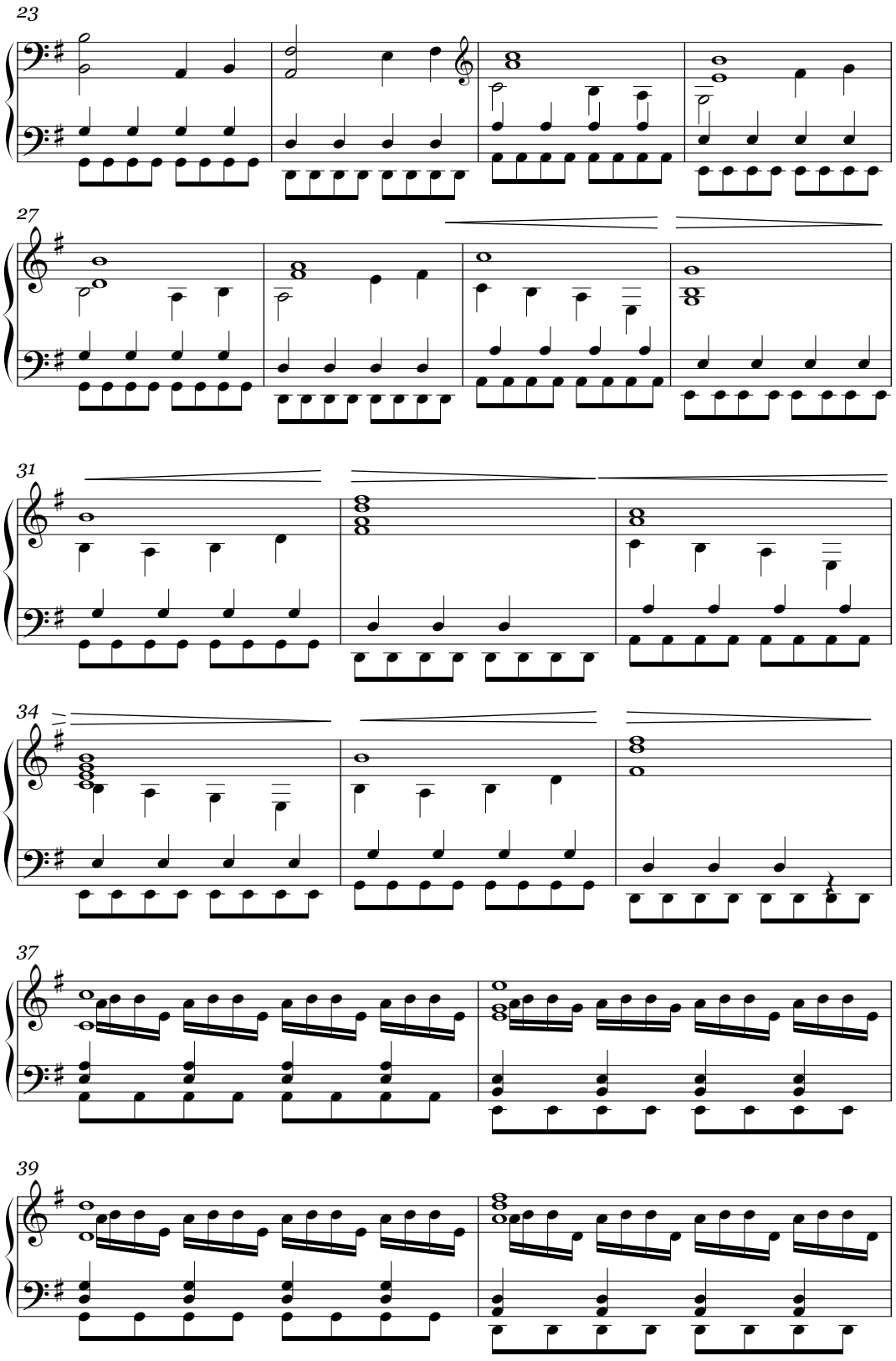


Time - *Hans Zimmer*

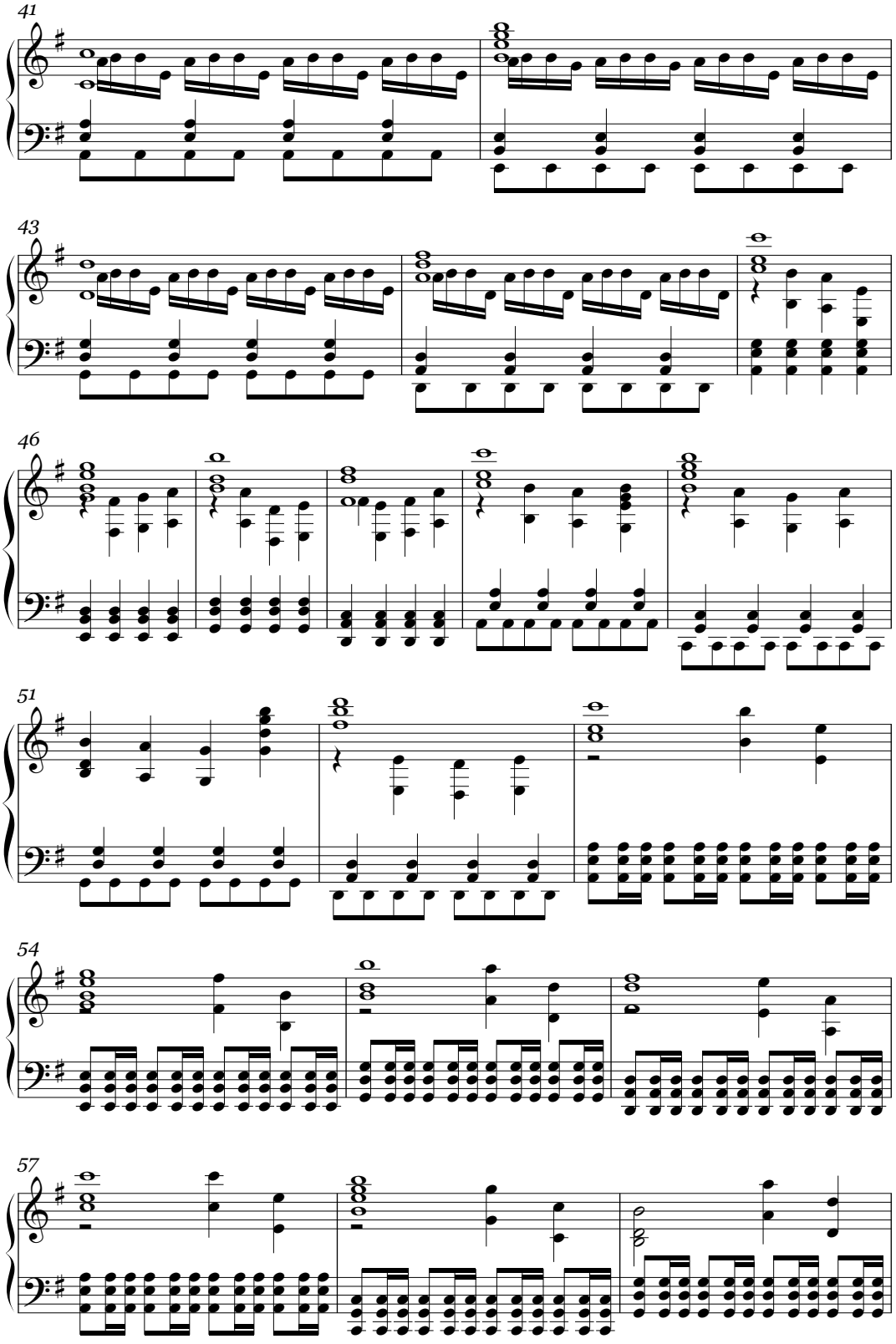


Time - *Hans Zimmer*

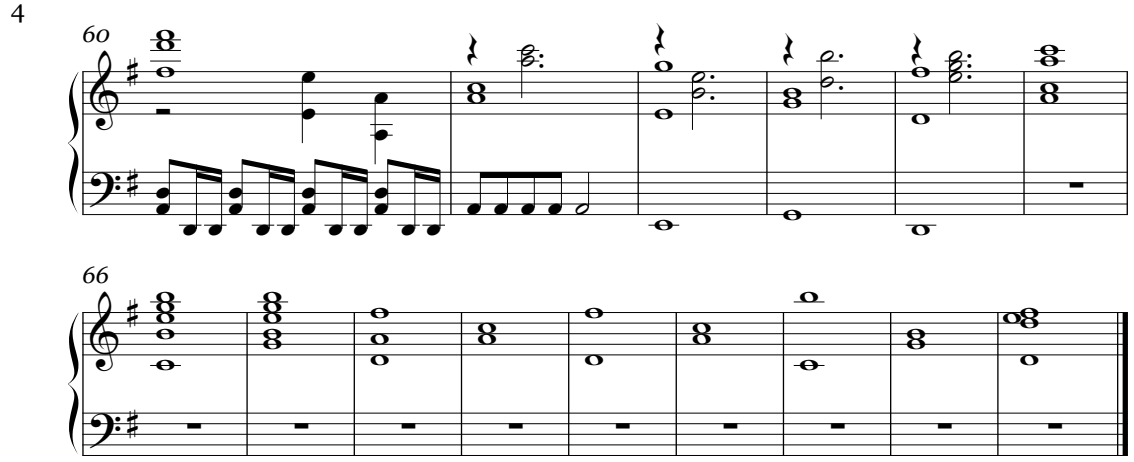
2



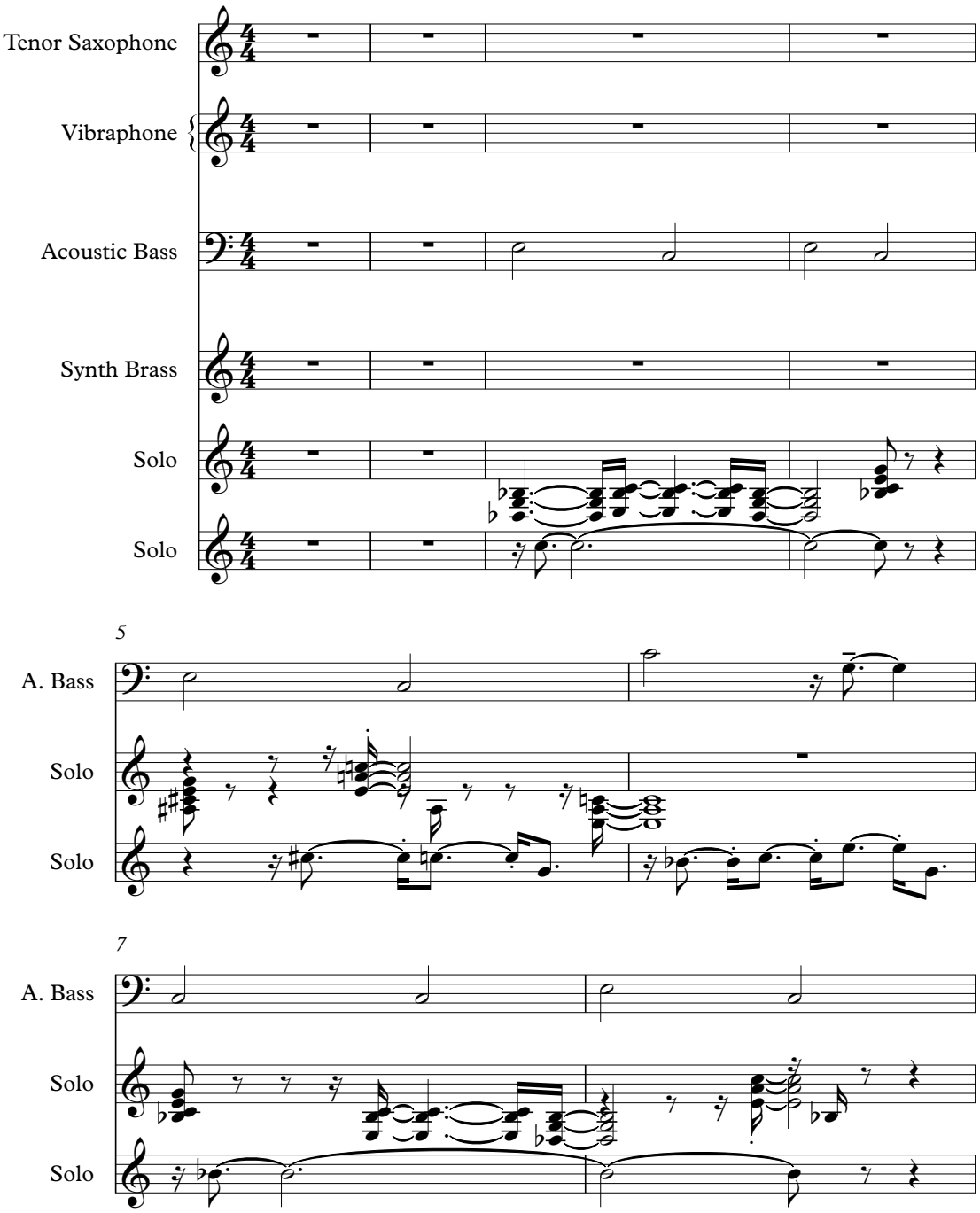
Time - *Hans Zimmer*



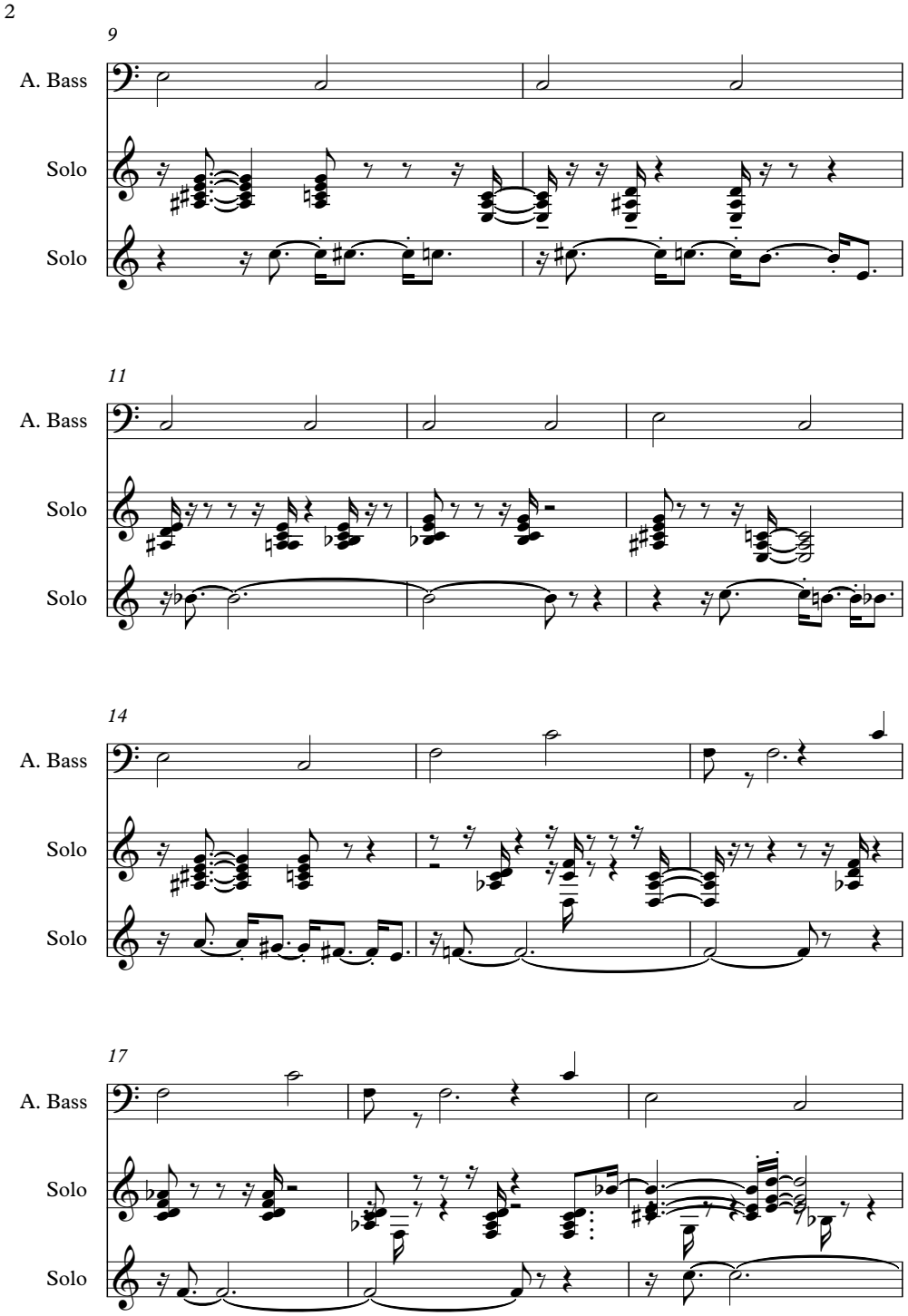
Time - *Hans Zimmer*



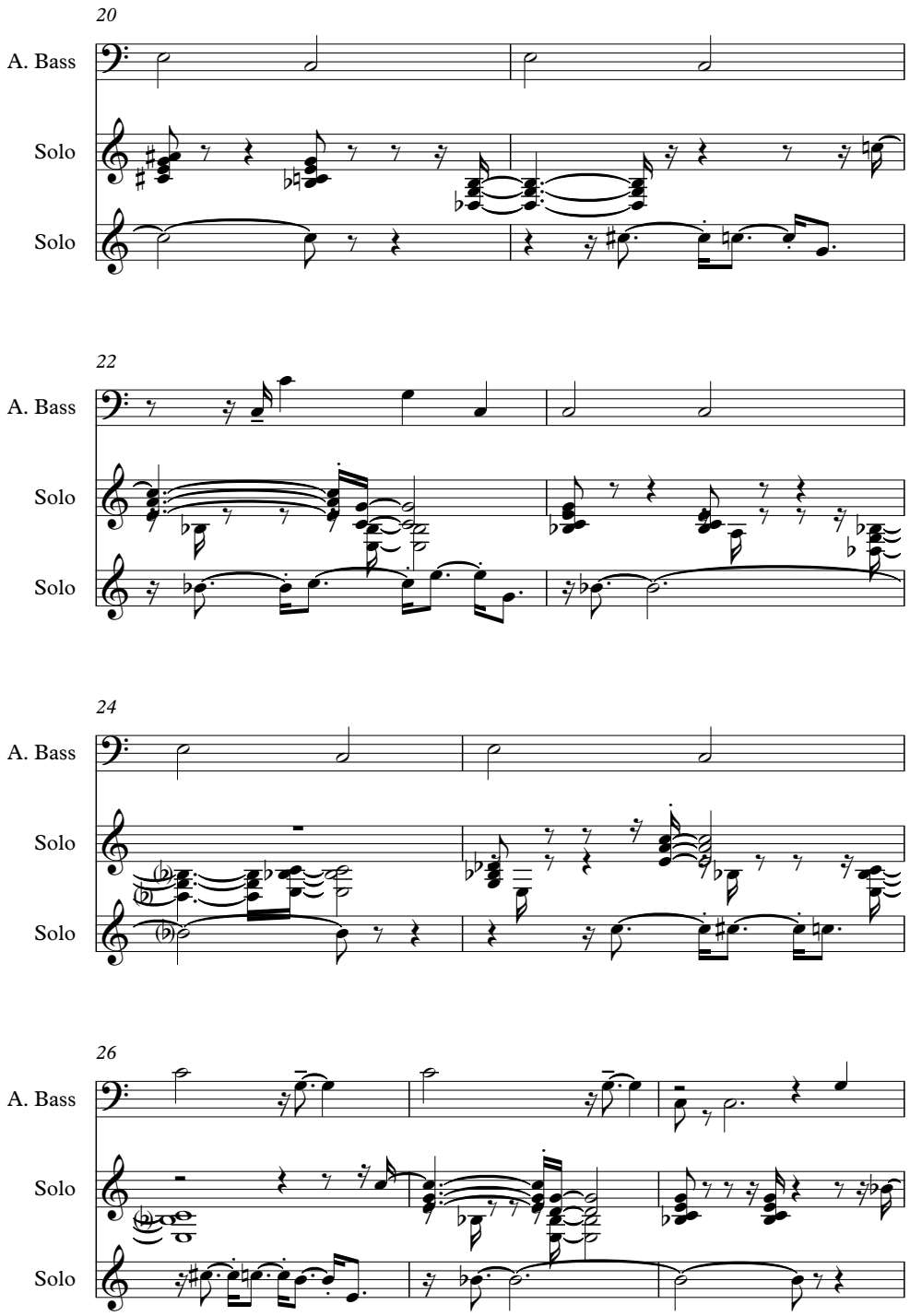
Caravan_md - bars 1 - 50



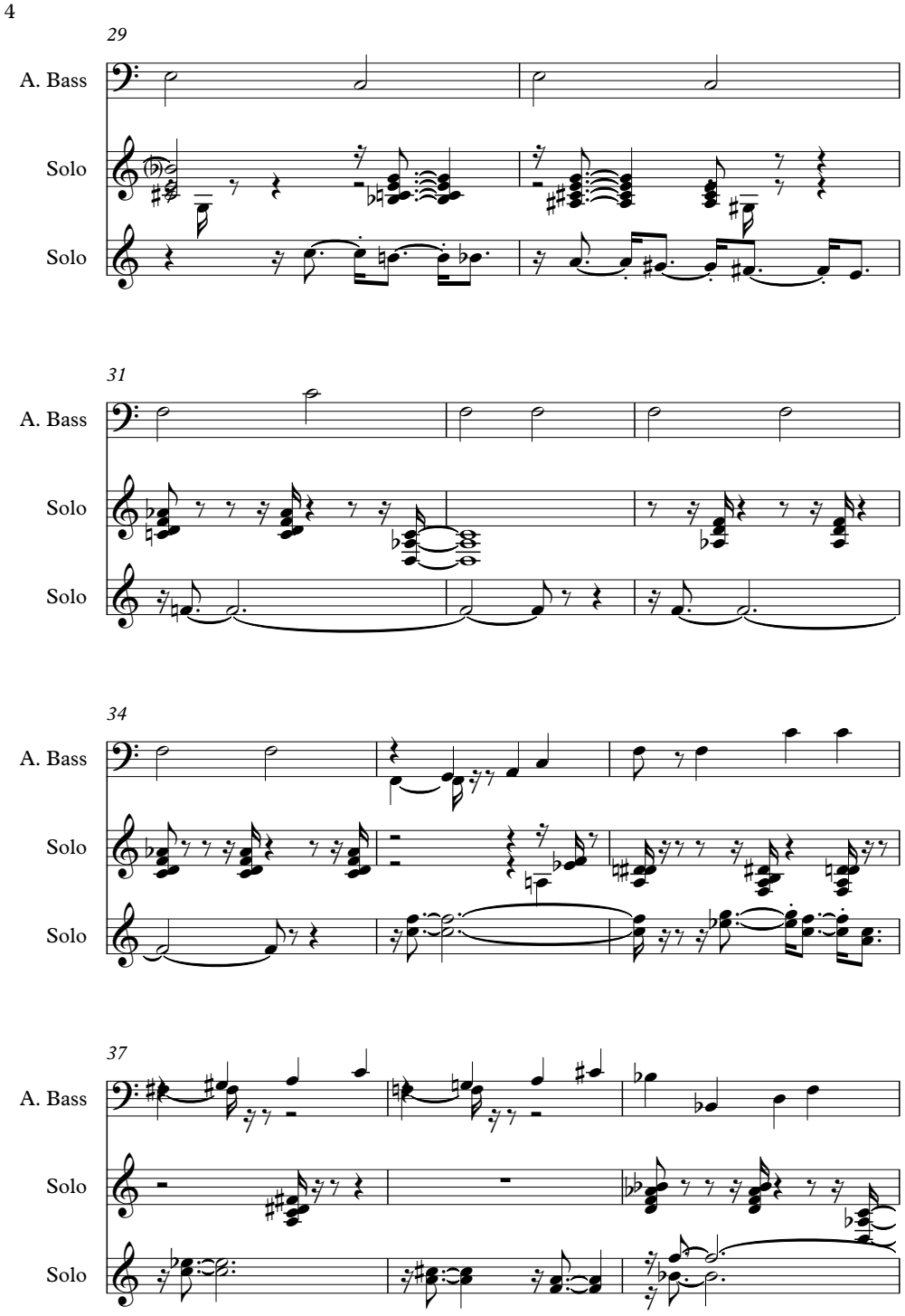
Caravan_md - bars 1 - 50



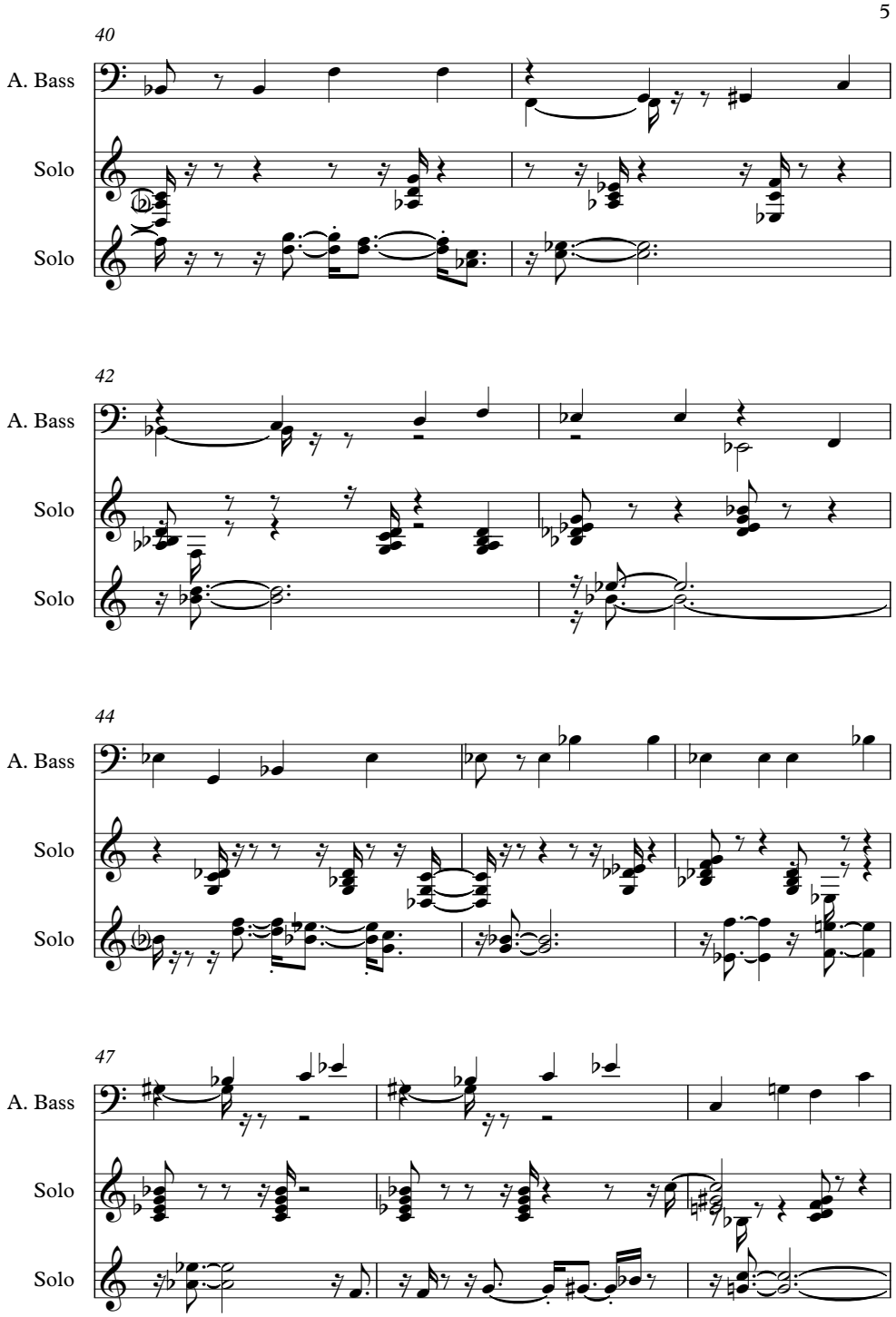
Caravan_md - bars 1 - 50



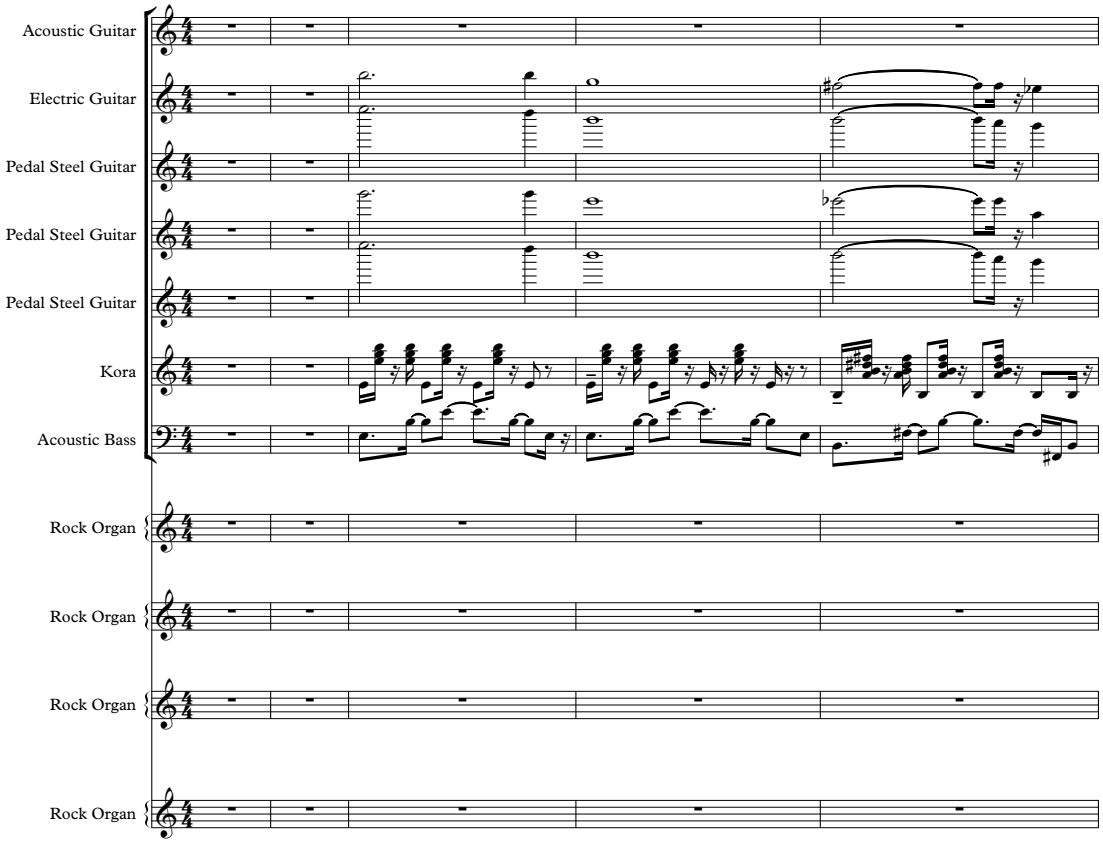
Caravan_md - bars 1 - 50

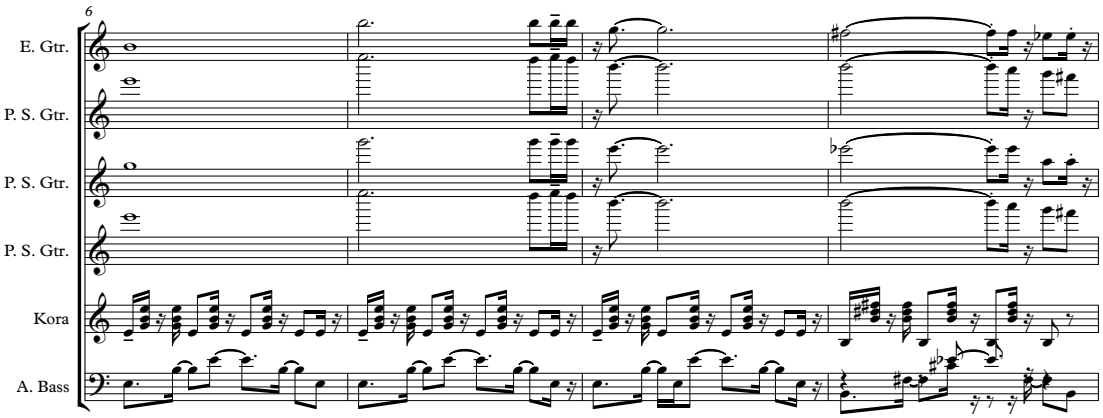


Caravan_md - bars 1 - 50

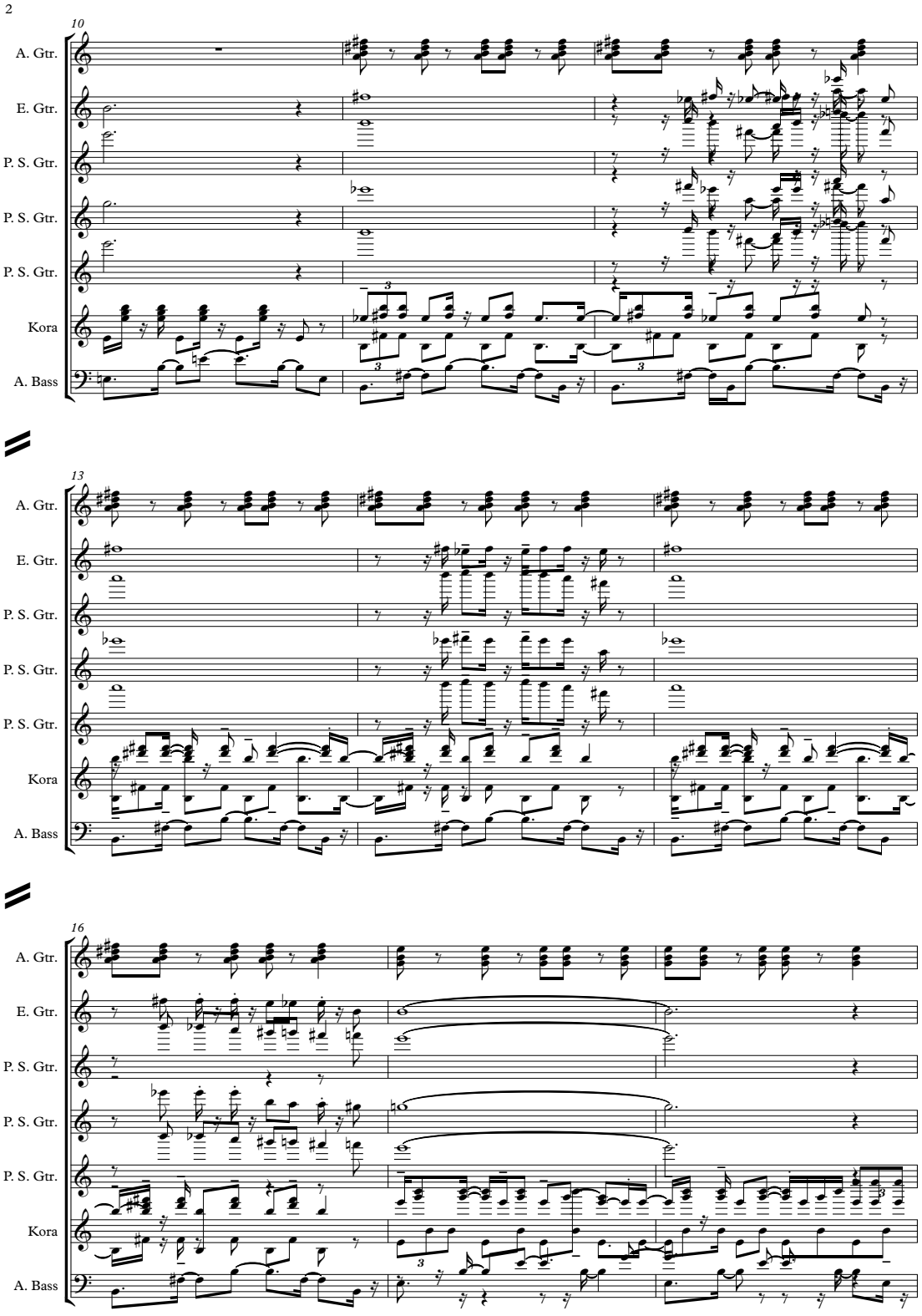


Caravan_js - bars 1 - 50

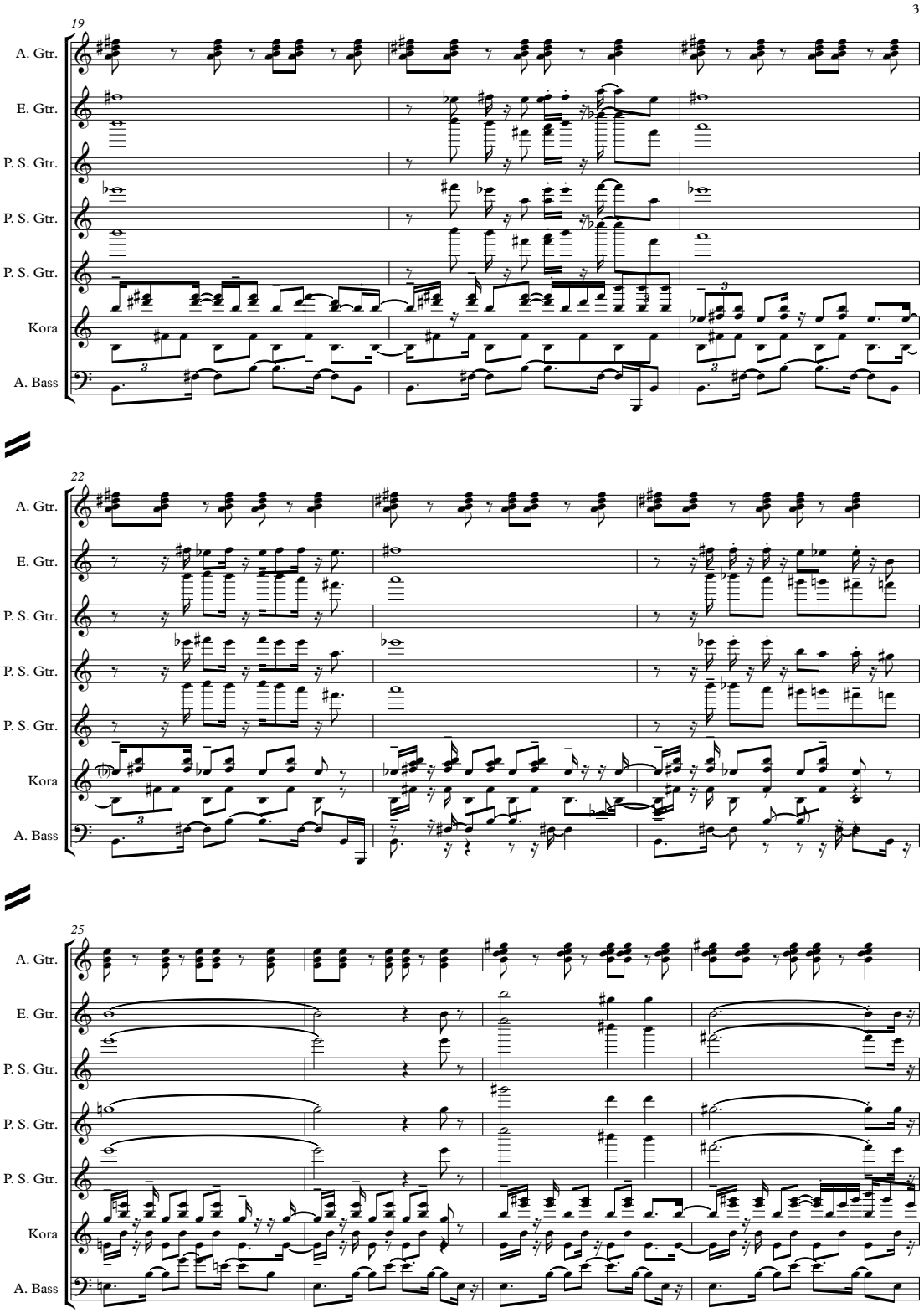




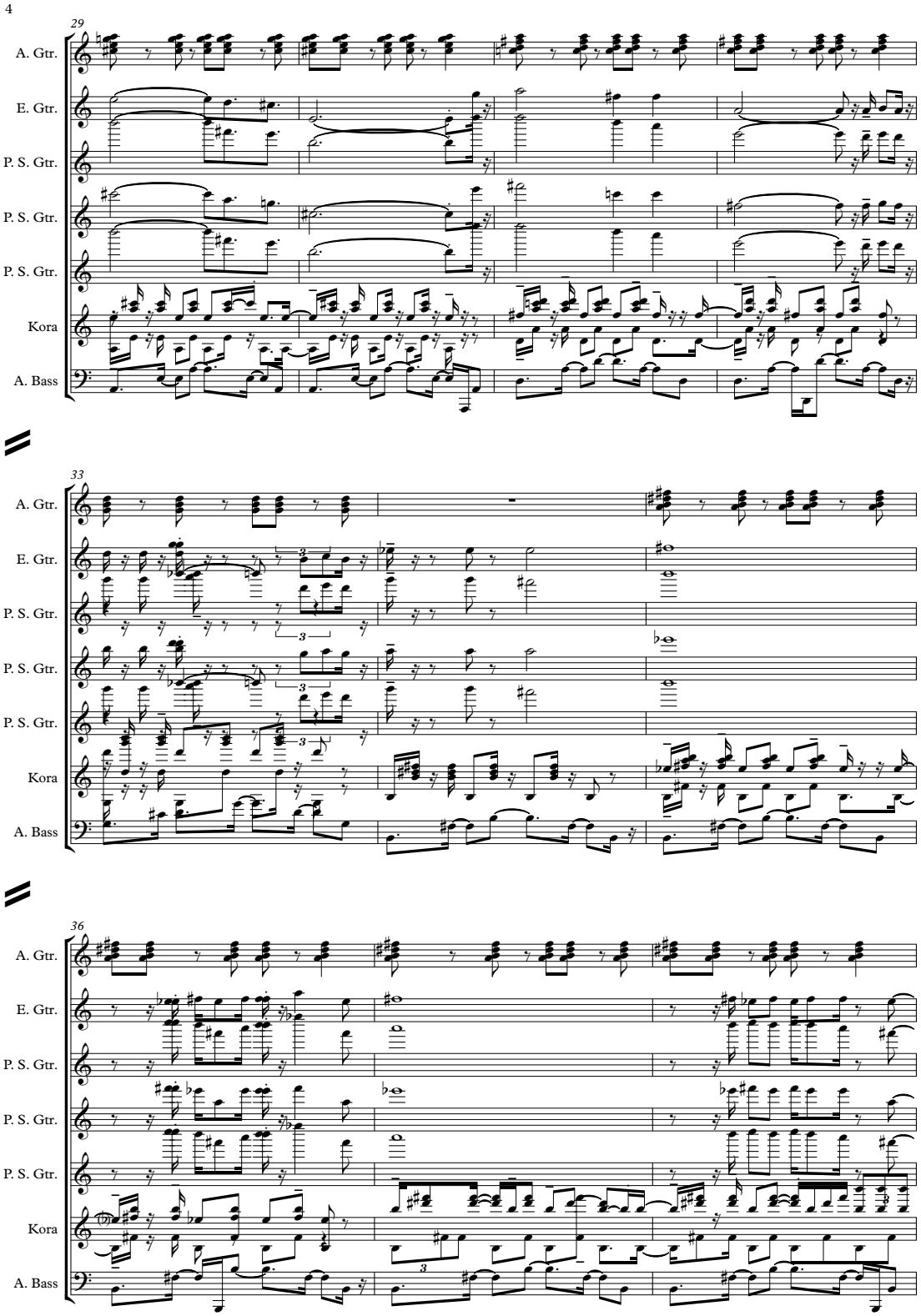
Caravan_js - bars 1 - 50



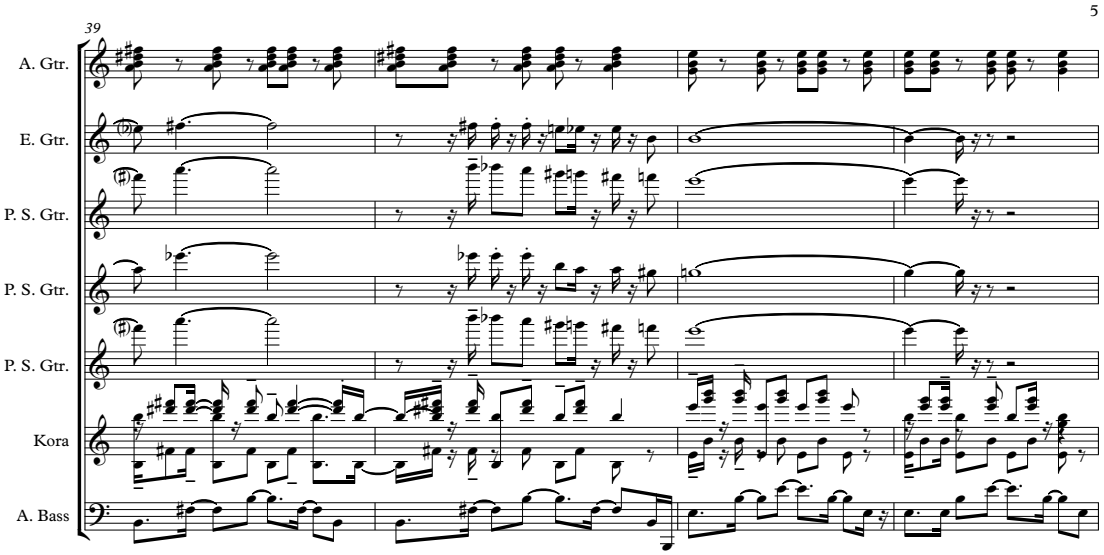
Caravan_js - bars 1 - 50

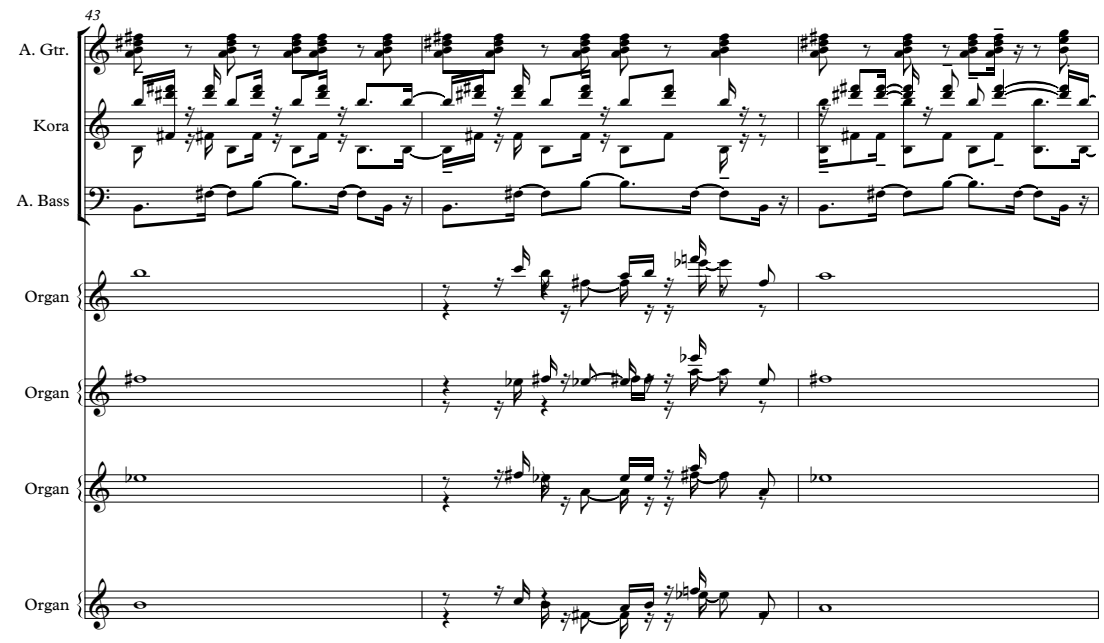


Caravan_js - bars 1 - 50

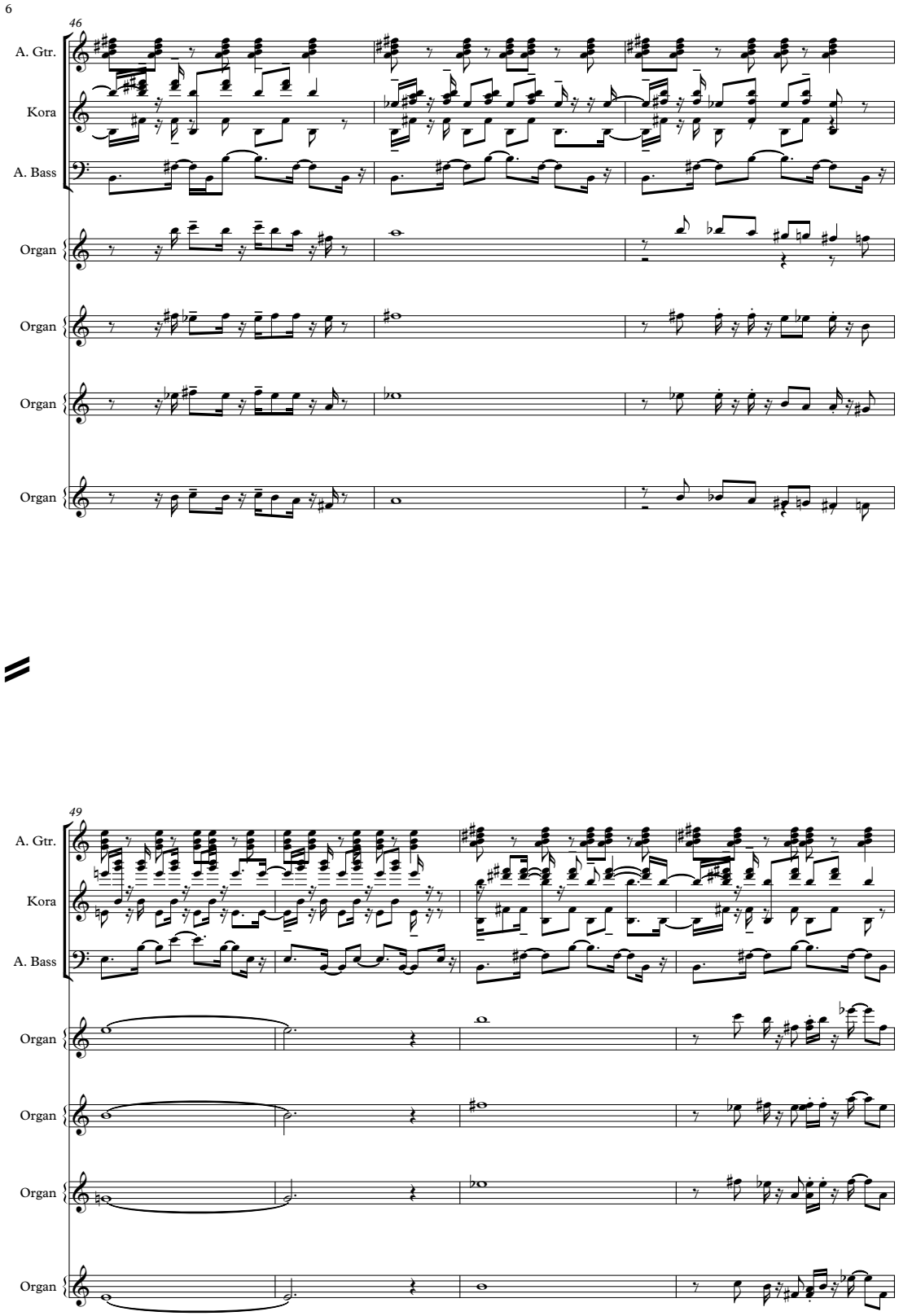


Caravan_js - bars 1 - 50





Caravan_js - bars 1 - 50



E

Modern Chord Notation

As it has been discussed in chapter 1 the lead sheet notation is widely used in modern music. We recall it consists of a melody written in classical notation supported by chords denoted as symbols. In tables E.1 and E.2 we list the most common symbols using C as root to build examples for triads and seventh chords, both including their chord tones extensions. A chord is said to be altered if a pitch not belonging diatonically to the chord is added, such as $\mathbf{C7^{\flat 9}} = (C, E, G, B\flat, D\flat)$. Where used, these chords will be described explicitly.

Table E.1: Modern triad notation.

Name	Notation	Arpeggio
Major	$\mathbf{C}$	(C, E, G)
Minor	$\mathbf{Cm}$	$(C, E\flat, G)$
Augmented	$\mathbf{Caug}$	$(C, E, G\sharp)$
Diminished	$\mathbf{Cdim}$	$(C, E\flat, G\flat)$
Suspended 2	$\mathbf{Csus2}$	(C, D, G)
Suspended 4	$\mathbf{Csus4}$	(C, F, G)
Major Added 9	$\mathbf{C^{add9}}$	(C, E, G, D)
Minor Added 9	$\mathbf{Cm^{add9}}$	(C, E, G, D)
Major Added 6	$\mathbf{C^{add6}}$	(C, E, G, A)

Table E.2: Modern seventh chords notation.

Name	Notation	Arpeggio
Major 7	$\mathbf{C^{\Delta}}$	(C, E, G, B)
Minor 7	$\mathbf{Cm7}$	$(C, E\flat, G, B\flat)$
Dominant	$\mathbf{C7}$	$(C, E, G, B\flat)$
Minor Major 7	$\mathbf{Cm^{\Delta}}$	$(C, E\flat, G, B)$
Major 7♯5	$\mathbf{C+^{\Delta}}$	$(C, E, G\sharp, B)$
Augmented Dominant	$\mathbf{C+^{7}}$	$(C, E, G\sharp, B\flat)$
Half-Diminished	$\mathbf{C^{\varnothing}}$	$(C, E\flat, G\flat, B\flat)$
Diminished	$\mathbf{C^{\circ}}$	$(C, E\flat, G\flat, B\flat\flat)$
Six	$\mathbf{C^{6}}$	$(C, E, G, B\flat, A)$
Nine	$\mathbf{C^{9}}$	$(C, E, G, B\flat, D)$
Six-Nine	$\mathbf{C^{6/9}}$	$(C, E, G, B\flat, A, D)$
Minor Nine	$\mathbf{Cm^{9}}$	$(C, E\flat, G, B\flat, D)$

List of Figures

List of Tables

List of algorithms

Bibliography

Abrams, A. and Ghrist, R. (2002). Finding topology in a factory: configuration spaces. *American Mathematical Monthly*, pages 140–150.

Adcock, A., Rubin, D., and Carlsson, G. (2014). Classification of hepatic lesions using the matching metric. *Computer vision and image understanding*, 121:36–42.

Ahola, V., Aittokallio, T., Vihinen, M., and Uusipaikka, E. (2006). A statistical score for assessing the quality of multiple sequence alignments. *BMC bioinformatics*, 7(1):484.

Aldwell, E., Schachter, C., and Cadwallader, A. (2010). *Harmony and voice leading*. Cengage Learning.

Alexander, J. W. (1923). A lemma on systems of knotted curves. *Proceedings of the National Academy of Sciences of the United States of America*, 9(3):93.

Alexander, J. W. (1928). Topological invariants of knots and links. *Transactions of the American Mathematical Society*, 30(2):275–306.

Andreatta, M. (2003). *Méthodes algébriques en musique et musicologie du XXe siecle: aspects théoriques, analytiques et compositionnels*. PhD thesis, École des Hautes Etudes en Sciences Sociales.

Apel, W. (1958). *Gregorian chant*, volume 601. Indiana University Press.

Armougom, F., Moretti, S., Keduas, V., and Notredame, C. (2006). The iRMSD: a local measure of sequence alignment accuracy using structural information. *Bioinformatics*, 22(14):e35–e39.

Aucouturier, J.-J., Pachet, F., and Sandler, M. (2005). "The way it Sounds": timbre models for analysis and retrieval of music signals. *Multimedia, IEEE Transactions on*, 7(6):1028–1035.

Bailey, T. L., Williams, N., Misleh, C., and Li, W. W. (2006). MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic acids research*, 34(suppl 2):W369–W373.

Barona, M. E. A. (2014). The fender rhodes.

Basil, S. (1963). *Exegetic homilies*, volume 46. Catholic Univ of Amer Pr.

Bergomi, M. G. (2015). (Talk). Dynamics in Modern Music Analysis. XXIst Oporto Meeting on Geometry, Topology and Physics. Applications of Topology.

Bergomi, M. G. and Andreatta, M. (2015). Math'n pop versus math'n folk? a computational (ethno) musicological approach. *Folk Music Analysis*.

Bergomi, M. G., Andreatta, M., and Fabbri, F. (2015). Hey Maths! Modèles formels et computationnels au service des Beatles. *Volume! (preprint)*.

Bergomi, M. G. and Geravini, S. (2012). *I Modi delle Scale*. Casa Musicale Eco.

Bergomi, M. G., Jadanza, R. D., and Portaluri, A. (2014a). Modelli geometrici e dinamici per spazi musicali. In Ferrara, F., Giacardi, L. M., and Mosca, M., editors, *Conferenze e Seminari dell'Associazione Subalpina Mathesis 2013–2014*, chapter Le Conferenze, pages 179–196. Kim Williams Books, Torino, Italy.

Bergomi, M. G., Jadanza, R. D., and Portaluri, A. (2014b). Una geometrizzazione dello spazio degli accordi. *Ithaca*, (3):33–46.

Bergomi, M. G. and Portaluri, A. (2013). Modes in modern music from a topological viewpoint. *arXiv preprint arXiv:1309.0687*.

Berndt, D. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *AAAI-94 workshop on knowledge discovery in databases*, pages 229–248.

Bigo, L. (2013). *Spatial Computing for Symbolic Musical Representations*. PhD thesis.

Bigo, L., Andreatta, M., Giavitto, J.-L., Michel, O., and Spicher, A. (2013). Computation and visualization of musical structures in chord-based simplicial complexes. In *Mathematics and Computation in Music*, pages 38–51. Springer.

Bimbot, F., Deruty, E., Sargent, G., and Vincent, E. (2012). Semiotic structure labeling of music pieces: concepts, methods and annotation conventions. In *13th International Society for Music Information Retrieval Conference (ISMIR)*.

Birkhoff, G. D. (1933). *Aesthetic measure*. Cambridge, Mass.

Birman, J. S. (1974). *Braids, links, and mapping class groups*. Number 82. Princeton University Press.

Birman, J. S. (1993). New points of view in knot theory. *Bulletin of the American Mathematical Society*, 28(2):253–287.

Boland, M. and Link, J. (2012). *Elliott Carter Studies*. Cambridge University Press.

Brattico, E. and Pearce, M. (2013). The neuroaesthetics of music. *Psychology of Aesthetics, Creativity, and the Arts*, 7(1):48.

Brinkmann, R. (1969). *Arnold Schönberg, drei Klavierstücke Op. 11: Studien zur frühen Atonalität bei Schönberg*. Franz Steiner Verlag.

Bugge, E. P., Juncher, K. L., Mathiesen, B. S., and Simonsen, J. G. (2011). Using sequence alignment and voting to improve optical music recognition from multiple recognizers. In *12th International Society for Music Information Retrieval Conference*, pages 405–410.

Buteau, C. and Mazzola, G. (2000). From contour similarity to motivic topologies. *Musicae Scientiae*, 4(2):125–149.

Cagliari, F., Di Fabio, B., and Ferri, M. (2010). One-dimensional reduction of multidimensional persistent homology. *Proceedings of the American Mathematical Society*, 138(8):3003–3017.

Callender, C., Quinn, I., and Tymoczko, D. (2008). Generalized Voice-Leading Spaces. *Science*, 320:346–348.

Carlsson, G. and Zomorodian, A. (2009). The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93.

Carlsson, G., Zomorodian, A., Collins, A., and Guibas, L. J. (2005). Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(02):149–187.

Casey, M., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., and Slaney, M. (2008). Content-based music information retrieval: current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696.

Cerri, A., Fabio, B. D., Ferri, M., Frosini, P., and Landi, C. (2013). Betti numbers in multidimensional persistent homology are stable functions. *Mathematical Methods in the Applied Sciences*, 36(12):1543–1557.

Chanan, M. (1994). *Musica practica: The social practice of Western music from Gregorian chant to postmodernism*. Verso.

Chazal, F., Cohen-Steiner, D., Guibas, L. J., Mémoli, F., and Oudot, S. Y. (2009). Gromov-Hausdorff Stable Signatures for Shapes using Persistence. In *Computer Graphics Forum*, volume 28, pages 1393–1403. Wiley Online Library.

Chen, L. and Ng, R. (2004). On the marriage of Lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 792–803. VLDB Endowment.

Chew, E. (2002). The spiral array: An algorithm for determining key boundaries. In *Music and artificial intelligence*, pages 18–31. Springer.

Chung, M. K., Bubenik, P., and Kim, P. T. (2009). Persistence diagrams of cortical surface data. In *Information Processing in Medical Imaging*, pages 386–397. Springer.

Cohen-Steiner, D., Edelsbrunner, H., and Morozov, D. (2006). Vines and vineyards by updating persistence in linear time. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 119–126. ACM.

Cohen-Steiner, D. and Morvan, J.-M. (2003). Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 312–321. ACM.

Cohn, R. (2011). *Audacious Euphony: Chromatic Harmony and the Triad's Second Nature*. Oxford University Press.

Crestel, L. (2015). Deep symbolic learning of multiple temporal granularities for musical orchestration.

d'Amico, M., Ferri, M., and Stanganelli, I. (2004). Qualitative Asymmetry Measure for Melanoma Detection. In *ISBI*, pages 1155–1158.

d'Amico, M., Frosini, P., and Landi, C. (2006). Using matching distance in size theory: A survey. *International Journal of Imaging Systems and Technology*, 16(5):154–161.

d'Arezzo, G., Colette, M.-N., and Jolivet, J.-C. (1993). *Micrologus*. Éd. IPMC.

Das, G., Gunopulos, D., and Mannila, H. (1997). Finding Similar Time Series. In *Principles of data mining and knowledge discovery: First European Symposium, PKDD'97, June 24-27*, volume 1263, pages 88–100, Trondheim, Norway. Springer Verlag.

De Silva, V. and Ghrist, R. (2007). Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358.

Di Fabio, B. and Ferri, M. (2015). Comparing persistence diagrams through complex vectors. *arXiv preprint arXiv:1505.01335*.

Di Fabio, B. and Frosini, P. (2013). Filtrations induced by continuous functions. *Topology and its Applications*, 160(12):1413–1422.

Di Fabio, B. and Landi, C. (2011). A Mayer–Vietoris formula for persistent homology with an application to shape recognition in the presence of occlusions. *Foundations of Computational Mathematics*, 11(5):499–527.

Do, C. B., Mahabhashyam, M. S., Brudno, M., and Batzoglou, S. (2005). ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome research*, 15(2):330–340.

Douthett, J. and Steinbach, P. (1998). Parsimonious graphs: A study in parsimony, contextual transformations, and modes of limited transposition. *Journal of Music Theory*, pages 241–263.

Dowling, W. J. (1972). Recognition of melodic transformations: Inversion, retrograde, and retrograde inversion. *Perception & Psychophysics*, 12(5):417–421.

Dudeque, N. (2005). *Music theory and analysis in the writings of Arnold Schoenberg (1874-1951)*. Ashgate Publishing, Ltd.

Easdown, D. and Lavers, T. (2004). The inverse braid monoid. *Advances in Mathematics*, 186(2):438–455.

East, J. (2007). Braids and partial permutations. *Advances in Mathematics*, 213(1):440–461.

East, J. (2010). Singular braids and partial permutations. *preprint*.

Edelsbrunner, H. and Harer, J. (2008). Persistent homology-a survey. *Contemporary mathematics*, 453:257–282.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2002). Topological persistence and simplification. *Discrete and Computational Geometry*, 28(4):511–533.

Edgar, R. C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797.

Ellis, D. P. and Weller, A. V. (2010). The 2010 LABROSA chord recognition system. *MIREX 2010*.

Erickson, R. and Palisca, C. V. (1995). *Musica Enchiriadis: And, Scolica Enchiriadis*. Yale University Press.

Esling, P. and Agon, C. (2012). Time series data mining. *ACM Computing Surveys*, 45(1).

Esling, P. and Bergomi, M. G. (2015). Multiple sequence alignment and the musical molecular clock hypothesis. *ACM Trans Intell Syst Technol (submitted)*.

Euler, L. (1739a). *Tentamen novae theoriae musicae ex certissimis harmoniae principiis dilucide expositae*. ex typographia Academiae scientiarum.

Euler, L. (1739b). *Tentamen novae theoriae musicae ex certissismis harmoniae principiis dilucide expositae*. Saint Petersburg Academy. p. 147.

Euler, L. (1774). De harmoniae veris principiis per speculum musicum repraesentatis. *Opera Omnia*, 3(1):568–586.

Euler, M. (1766). Conjecture sur la raison de quelques dissonances generalement recues dans la musique.

Everett, W. (2000). *Expression in pop-rock music: a collection of critical and analytical essays*, volume 2. Taylor & Francis.

Fenn, R. and Keyman, E. (2000). Extended braids and links. *Knots in Hellas*, 98:229–251.

Ferri, M., Frosini, P., and Landi, C. (2011). Stable Shape Comparison by Persistent Homology.

Fletcher, H. (1940). Auditory patterns. *Reviews of modern physics*, 12(1):47.

Folgieri, R., Bergomi, M. G., and Castellani, S. (2014). EEG-Based Brain-Computer Interface for Emotional Involvement in Games Through Music. In *Digital Da Vinci*, pages 205–236. Springer.

Foote, J. and Uchihashi, S. (2001). The beat spectrum: A new approach to rhythm analysis. In *null*, page 224. IEEE.

Forman, R. (1998). Witten–Morse theory for cell complexes. *Topology*, 37(5):945–979.

Forman, R. (2002). A user's guide to discrete Morse theory. *Sém. Lothar. Combin*, 48:35pp.

Frosini, P. (1992). Measuring shapes by size functions. In *Intelligent Robots and Computer Vision X: Algorithms and Techniques*, pages 122–133. International Society for Optics and Photonics.

Frosini, P. and Landi, C. (2001). Size functions and formal series. *Applicable Algebra in Engineering, Communication and Computing*, 12(4):327–349.

Galilei, G. (1638). *Discorsi e dimostrazioni matematiche, intorno à due nuove scienze*.

Galilei, V. (1569). *Il fronimo*. Forni.

Ghrist, R. (2008). Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75.

Ghrist, R. and Peterson, V. (2007). The geometry and topology of reconfiguration. *Advances in applied mathematics*, 38(3):302–323.

Giblin, P. (2010). *Graphs, surfaces and homology*. Cambridge University Press.

Govc, D. (2013). On the definition of homological critical value. *arXiv:1301.6817*.

Hansen, V. L. (1989). *Braids and coverings: selected topics*, volume 18. Cambridge University Press.

Harte, C. and Sandler, M. (2005). Automatic chord identifcation using a quantised chromagram. In *Audio Engineering Society Convention 118*. Audio Engineering Society.

Hatcher, A. (2002). *Algebraic topology*. Cambridge University Press.

Helmholtz, H. v. (1877). *Die Lehre von den Tonempfindungen als physiologische Grundlage für die Theorie der Musik*. Vieweg, Braunschweig.

Hertz, G. Z. and Stormo, G. D. (1999). Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7):563–577.

Horn, R. A. and Johnson, C. R. (1991). *Topics in matrix analysis*. Cambridge University Press, Cambridge.

Hughes, J. R. (2015). Using Fundamental Groups and Groupoids of Chord Spaces to Model Voice Leading. In *Mathematics and Computation in Music*, pages 267–278. Springer.

İzmirli, Ö. and Dannenberg, R. B. (2010). Understanding Features and Distance Functions for Music Sequence Alignment. In *ISMIR*, pages 411–416.

Jensen, K. (2007). Multiple scale music segmentation using rhythm, timbre, and harmony. *EURASIP Journal on Applied Signal Processing*, 2007(1):159–159.

Johnson, M. (2009). *Pop Music Theory*. Lulu. com.

Juslin, P. N. and Västfjäll, D. (2008). Emotional responses to music: The need to consider underlying mechanisms. *Behavioral and brain sciences*, 31(05):559–575.

Katoh, K., Misawa, K., Kuma, K.-i., and Miyata, T. (2002). MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research*, 30(14):3059–3066.

King, H., Knudson, K., and Mramor, N. (2005). Generating discrete Morse functions from point data. *Experimental Mathematics*, 14(4):435–444.

Knees, P., Schedl, M., and Widmer, G. (2005). Multiple Lyrics Alignment: Automatic Retrieval of Song Lyrics. In *ISMIR*, pages 564–569.

Kozlov, D. (2007). *Combinatorial algebraic topology*, volume 21. Springer Science & Business Media.

Kurth, E. and Rothfarb, L. A. (1991). *Ernst Kurth: selected writings*. Number 2. Cambridge University Press.

Langfelder, P., Zhang, B., and Horvath, S. (2008). Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. *Bioinformatics*, 24(5):719–720.

Larson, S. (2012). *Musical Forces: Motion, Metaphor, and Meaning in Music*. Indiana University Press.

Lassmann, T. and Sonnhammer, E. L. (2005). Automatic assessment of alignment quality. *Nucleic acids research*, 33(22):7120–7128.

Lee, K. M., Skoe, E., Kraus, N., and Ashley, R. (2009). Selective subcortical enhancement of musical intervals in musicians. *The Journal of Neuroscience*, 29(18):5832–5840.

Lester, J. (1994). *Compositional theory in the eighteenth century*. Harvard University Press.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Levine, M. (2011). *The jazz theory book*. " O'Reilly Media, Inc.".

Lewin, D. (2007). *Generalized musical intervals and transformations*. Oxford University Press.

Li, T. L., Chan, A. B., and Chun, A. (2010). Automatic musical pattern feature extraction using convolutional neural network. In *Proc. Int. Conf. Data Mining and Applications*.

Martin, A. P. and Palumbi, S. R. (1993). Body size, metabolic rate, generation time, and the molecular clock. *Proceedings of the National Academy of Sciences*, 90(9):4087–4091.

Martin, B., Brown, D. G., Hanna, P., and Ferraro, P. (2012). Blast for Audio Sequences Alignment: a Fast Scalable Cover Identification. In *13th International Society for Music Information Retrieval Conference*, pages pages–529.

Martinez, W. L., Martinez, A., and Solka, J. (2010). *Exploratory data analysis with MATLAB*. CRC Press.

Matityaho, B. and Furst, M. (1995). Neural network based model for classification of music type. In *Electrical and Electronics Engineers in Israel, 1995., Eighteenth Convention of*, pages 4–3. IEEE.

Mauch, M. (2010). *Automatic chord transcription from audio using computational models of musical context*. PhD thesis, School of Electronic Engineering and Computer Science Queen Mary, University of London.

Mauch, M., Noland, K., and Dixon, S. (2009). Using Musical Structure to Enhance Automatic Chord Transcription. In *ISMIR*, pages 231–236.

Mazzola, G. and Andreatta, M. (2006). From a categorical point of view: K-nets as limit denotators. *Perspectives of New Music*, pages 88–113.

Mazzola, G. et al. (2002). *The topos of music*. Birkhäuser, Basel.

Mileyko, Y., Mukherjee, S., and Harer, J. (2011). Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007.

Milnor, J. W. (1963). *Morse theory*. Number 51. Princeton university press.

Munch, E. (2013). *Applications of persistent homology to time varying systems*. PhD thesis, Duke University.

Munch, E., Shapiro, M., and Harer, J. (2012). Failure filtrations for fenced sensor networks. *The International Journal of Robotics Research*, 31(9):1044–1056.

Munkres, J. R. (1984). *Elements of algebraic topology*, volume 2. Addison-Wesley Reading.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Notley, M. A. (2007). *Lateness and Brahms: music and culture in the twilight of Viennese liberalism*. Oxford University Press, USA.

Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology*, 302(1):205–217.

Ogdon, W. (1981). HOW TONALITY FUNCTIONS IN SCHOENBERG OPUS-11, NUMBER-1. *Journal of the Arnold Schoenberg Institute*, 5(2):169–181.

Osindero, S. and Hinton, G. E. (2008). Modeling image patches with a directed hierarchy of Markov random fields. In *Advances in neural information processing systems*, pages 1121–1128.

OSullivan, O., Zehnder, M., Higgins, D., Bucher, P., Grosdidier, A., and Notredame, C. (2003). APDB: a novel measure for benchmarking sequence alignment methods without reference alignments. *Bioinformatics*, 19(suppl 1):i215–i221.

Ott, N. (2009). *Visualization of Hierarchical Clustering: Graph Types and Software Tools*. GRIN Verlag.

Pardo, B. and Sanghi, M. (2005). Polyphonic Musical Sequence Alignment for Database Search. Citeseer.

Pass, J. (1987). *Joe Pass guitar chords*. Alfred Musicr.

Pearsall, E. (2012). *Twentieth-century music theory and practice*. Routledge.

Pérez-Escudero, A., Vicente-Page, J., Hinz, R. C., Arganda, S., and de Polavieja, G. G. (2014). idTracker: tracking individuals in a group by automatic identification of unmarked animals. *Nature methods*, 11(7):743–748.

Perricone, J. (2000). *Melody in songwriting: tools and techniques for writing hit songs*. Hal Leonard Corporation.

Piston, W. (1947). *Counterpoint*. WW Norton & Company.

Piston, W., De Voto, M., and Jannery, A. (1978). *Harmony*. Gollancz, London.

Plomp, R. and Levelt, W. J. (1965). Tonal consonance and critical bandwidth. *Journal of the Acoustical Society of America*, 38(4):548–560.

Plomp, R. and Steeneken, H. (1968). Interference between two simple tones. *Journal of the Acoustical Society of America*, 43(4):883–884.

Popoff, A., Andreatta, M., and Ehresmann, A. (2015). A Categorical Generalization of Klumpenhouwer Networks. In *Mathematics and Computation in Music*, pages 303–314. Springer.

Prout, E. (2012). *The orchestra: orchestral techniques and combinations*. Courier Dover Publications.

Rankin, S. (1993). Winchester Polyphony. The Early Theory and Practice of Organum. *Music in the Medieval English Liturgy*, pages 59–100.

Russo, W. (1997). *Jazz composition and orchestration*. University of Chicago Press.

Sankoff, D. (1972). Matching sequences under deletion/insertion constraints. *Proceedings of the National Academy of Sciences*, 69(1):4–6.

Senin, P. (2008). Dynamic time warping algorithm review. *University of Hawaii*.

Sethares, W. (2004). *Tuning, Timbre, Spectrum Scale*. Springer, New York.

Six, J. and Cornelis, O. (2012). A Robust Audio Fingerprinter Based on Pitch Class Histograms Applications for Ethnic Music Archives. In *Proceedings of the Folk Music Analysis conference (FMA 2012)*.

Slavich, L. (2009–2010). Strutture algebriche e topologiche nella musica del ventesimo secolo. Master's thesis, University of Pisa.

Smaragdis, P. and Brown, J. C. (2003). Non-negative matrix factorization for polyphonic music transcription. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 177–180. IEEE.

Sturm, B. L. (2013a). Classification accuracy is not enough. *Journal of Intelligent Information Systems*, 41(3):371–406.

Sturm, B. L. (2013b). The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. *arXiv preprint arXiv:1306.1461*.

Sturm, B. L. (2014). A survey of evaluation in music genre recognition. In *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, pages 29–66. Springer.

Sussman, R. and Abene, M. (2012). *Jazz composition and arranging in the digital age*. Oxford University Press.

Taruskin, R. (2009). *Music in the Nineteenth Century: The Oxford History of Western Music*. Oxford University Press.

Taylor, G. W. and Hinton, G. E. (2009). Factored conditional restricted Boltzmann machines for modeling motion style. In *Proceedings of the 26th annual international conference on machine learning*, pages 1025–1032. ACM.

Tenney, J. (1988). *A History of 'Consonance' and 'Dissonance'*. Excelsior Music Publishing Company, New York.

Thompson, J. D., Gibson, T., Higgins, D. G., et al. (2002). Multiple sequence alignment using ClustalW and ClustalX. *Current protocols in bioinformatics*, pages 2–3.

Thompson, J. D., Linard, B., Lecompte, O., and Poch, O. (2011). A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PloS one*, 6(3):e18093.

Thompson, J. D., Plewniak, F., Ripp, R., Thierry, J.-C., and Poch, O. (2001). Towards a reliable objective function for multiple sequence alignments. *Journal of molecular biology*, 314(4):937–951.

Thurston, W. P. (2002). The Geometry and Topology of Three-Manifolds. Electronic version 1.1, website: `http://library.msri.org/nonmsri/gt3m`.

Topaz, C. M., Ziegelmeier, L., and Halverson, T. (2015). Topological data analysis of biological aggregation models.

Trezise, S. (2003). *The Cambridge Companion to Debussy*. Cambridge University Press.

Tulipano, L. and Bergomi, M. G. (2015). Meaning, music and emotions: a neural activity analysis. In *NEA Science*, pages 105–108.

Tymoczko, D. (2006). The geometry of musical chords. *Science*, (313):72–74.

Tymoczko, D. (2008). Scale theory, serial theory and voice leading. *Music Analysis*, 27(1):1–49.

Tymoczko, D. (2011). *A geometry of music: harmony and counterpoint in the extended common practice*. Oxford University Press.

Tymoczko, D. (2012). The Generalized Tonnetz. *Journal of Music Theory*, 56(1):1–52.

von Appen, R., Doehring, A., Helms, D., and Moore, A. F. (2015). *Song Interpretation in 21st-Century Pop Music*. Ashgate Publishing, Ltd.

Wang, A. et al. (2003). An Industrial Strength Audio Search Algorithm. In *ISMIR*, pages 7–13.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8(3-4):279–292.

Wegel, R. and Lane, C. (1924). The auditory masking of one pure tone by another and its probable relation to the dynamics of the inner ear. *Physical review*, 23(2):266.

Wei, M. (2008). *Jazz Piano Handbook: Essential Jazz Piano Skills for All Musicians*. Alfred Music Publishing.

William, B. (1984). Harmony in Radical European Music. *Society of Music Theory*.

Yoshitaka, A. and Ichikawa, T. (1999). A survey on content-based retrieval for multimedia databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):81–93.

Zabka, M. (2009). Generalized Tonnetz and Well-Formed GTS: A Scale Theory Inspired by the Neo-Riemannians? *Mathematics and Computation in Music*, page 286.

Zwicker, E. (1961). Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *The Journal of the Acoustical Society of America*, 33(2):248–248.

Zwicker, E. and Terhardt, E. (1980). Analytical expressions for critical-band rate and critical bandwidth as a function of frequency. *The Journal of the Acoustical Society of America*, 68(5):1523–1525.