Representation of Musical Structures and Processes in Simplicial Chord Spaces

Louis Bigo, Daniele Ghisi, Antoine Spicher, Moreno Andreatta

Louis Bigo,[*] Daniele Ghisi,[†] Antoine Spicher,[**] and Moreno Andreatta[†]
[*]University of the Basque Country UPV/EHU
Department of Computer Science and Artificial Intelligence
Paseo de Manuel Lardizabal, 1
San-Sebastian, 20018 Spain
louis.bigo@ehu.es
[†]Institut de Recherche et Coordination Acoustique/Musique
Lab CNRS-IRCAM-UPMC
1 Place Igor-Stravinsky
75004 Paris, France
{daniele.ghisi, moreno.andreatta}@ircam.fr
[**]Université Paris-Est
Laboratoire d'Algorithmique, Complexité et Logique
61 avenue du Général de Gaulle
94010 Créteil, France
antoine.spicher@u-pec.fr

Representation of Musical Structures and Processes in Simplicial Chord Spaces

Abstract: In this article, we present a set of musical transformations based on the representations of chord spaces derived from the *Tonnetz*. These chord spaces are formalized as simplicial complexes. A musical composition is represented in such a space by a trajectory. Spatial transformations are applied on these trajectories and induce a transformation of the original composition. These concepts are implemented in two applications, the software HexaChord and the Max object `bach.tonnetz`, dedicated to music analysis and composition, respectively.

Spatial structures are commonly used by music theorists to represent sets of symbolic objects such as notes, chords, and rhythms. If geometry and topology play a major role in the representation of the musical objects, an algebraic formalization is necessary to grasp the combinatorial potential of all these structures. The study of the combinatorial properties of these geometrical and topological spaces is also a source of inspiration for new approaches in music theory, analysis, and composition. Geometrical and topological spaces are conceived as "support spaces" in the representation of given musical objects. This is the case for the *Tonnetz*, a support space for representing neo-Riemannian transformations (Cohn 2012). Other examples of topological spaces used for musical analysis include work by Dmitri Tymoczko (2011) on "orbifolds," which can be used to describe voice-leading progressions, and Elaine Chew's work with spiral arrays, which can be used to track key boundaries (Chew 2002), although we will not delve further into these representations in this article.

Our focus in this article is on the harmonic dimension, and we use a set of chord spaces to describe and execute certain musical transformations in the pitch domain. These spaces, which are generalizations of the *Tonnetz*, are formalized as simplicial complexes, a topological concept that has proved to be useful in many music-theoretical and analytical situations (Bigo et al. 2013). In this work, we focus on the use of simplicial complexes in the representation of musical structures and process, with particular emphasis on musical transformations. After briefly introducing the concepts of simplicial complexes, simplicial collections, and structural inclusion in the next section, in the subsequent section we discuss the main support space of our approach (the *Tonnetz* and its generalized versions). Generalized *Tonnetze* are then described as simplicial complexes in which trajectories represent musical sequences. This is followed by a discussion of transformation of a sequence in a generic chord

Computer Music Journal, 39:3, pp. 9–24, Fall 2015
doi:10.1162/COMJ_a_00312

space, and we discuss two musical cases corresponding to the isomorphism between two support spaces and the automorphism within a given support space. Finally, we present two environments that are based on the geometric approach described in the article. The first environment, called HexaChord, is experimental software dedicated to computer-aided music analysis. The second one is the Max object `bach.tonnetz`, which focuses primarily on musical notation and new strategies for computer-aided music composition (Agostini and Ghisi 2015).

Technical Background

In this section, we present several spatial tools used in the course of this article. First, we describe simplicial complexes and collections, which allow us to formalize chord spaces. Then, we describe structural inclusions, which provide a formalization of transformations within chord spaces.

Simplicial Complexes

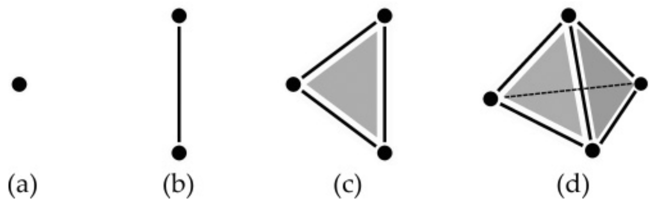
Given a set V of elements, a *simplicial complex* $\mathcal{K}$ defined on V is a collection of nonempty finite subsets of V, called *simplices* and denoted $\sigma \in \mathcal{K}$, verifying the *closure condition*:

> for any simplex $\sigma \in \mathcal{K}$, every nonempty subset $\sigma' \subset \sigma$ is also an element of $\mathcal{K}$, i.e., $\sigma' \in \mathcal{K}$.

The definition of the closure condition is equivalent to saying that there is an *incidence relation* between two simplices σ' and σ, which will be written as $\sigma' \prec \sigma$. Every simplex σ of $\mathcal{K}$ is characterized by its dimension, dim() such that $\dim(\sigma) = \mathrm{card}(\sigma) - 1$, where the function card() gives the cardinality of σ. A simplex of dimension n is called an n-simplex. Vertices can be used to represent 0-simplices, edges for 1-simplices, triangles for 2-simplices, tetrahedra for 3-simplices, etc.

The closure condition implies that an edge is incident to two vertices, a triangle is incident to three edges, etc. In general, every n-simplex is incident to $n+1$ $(n-1)$-simplices. A proper subset of a simplicial complex $\mathcal{K}$ that is also a simplicial complex is called a *subcomplex* of $\mathcal{K}$.

Figure 1. A 0-simplex (a), a 1-simplex (b), a 2-simplex (c), and a 3-simplex (d).



For the sake of simplicity, we will often consider that the term "simplex" designates the subcomplex containing a simplex and all its incident simplices of lower dimensions. Figure 1 illustrates examples of n-simplices for $n \in \{0, 1, 2, 3\}$.

A simplicial d-complex is a simplicial complex where the highest dimension of any simplex is d. Graphs are special cases of simplicial complexes where the highest dimension of any simplex is equal to 1. Figure 2a illustrates a simplicial 3-complex. For any natural integer n, the *n-skeleton* of a simplicial complex $\mathcal{K}$ is defined by the subcomplex $\mathscr{S}_n(\mathcal{K})$ of this complex formed by its simplices of dimension n or less. Figure 2a illustrates a complex and its 1-skeleton (see Figure 2b).

Simplicial Collections

A *simplicial collection* $\mathcal{K}$ is a simplicial complex in which every simplex is labeled by an arbitrary value, as illustrated in Figure 2c. The term "collection" comes from the notion of *topological collections* used in the MGS programming language (Giavitto and Michel 2001), which has strongly inspired this work.

More formally, a simplicial collection is a function that associates values from an arbitrary set with the simplices of a simplicial complex. The notation $\mathcal{K}(\sigma)$ enables one to address the label associated with the cell σ in the collection $\mathcal{K}$. We denote as $|\mathcal{K}|$ the *support* of the collection $\mathcal{K}$, which is the simplicial complex without labels. A collection $\mathcal{K}'$ is a *subcollection* of $\mathcal{K}$ if $|\mathcal{K}'| \subset |\mathcal{K}|$ and $\mathcal{K}'(\sigma) = \mathcal{K}(\sigma)$ for every σ of $\mathcal{K}'$. When there is no danger of ambiguity, the notation $|\cdot|$ can be omitted. Similarly, we will often use the term "complex" to designate a simplicial collection, that is, the simplicial complex and its labels.

Figure 2. A simplicial complex $\mathcal{K}$ (a), the 1-skeleton $\mathscr{S}_1(\mathcal{K})$ (b), and a simplicial collection whose support is a subcomplex of $\mathcal{K}$ (c).

Figure 3. Two possible structural inclusions of one simplicial complex in another.

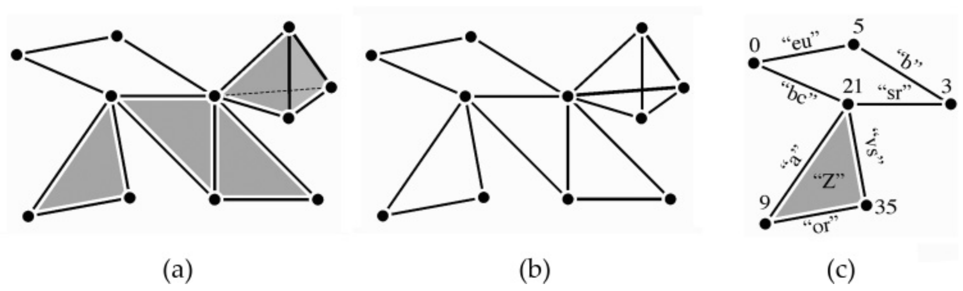


Figure 2

Structural Inclusions

In this work, we will be interested in the ways one complex can be embedded into another (or into itself). In order to deal with this notion, we introduce the concepts of *morphism* and *structural inclusion*.

Let $\mathcal{K}$ and $\mathcal{K}'$ be two simplicial complexes. A function $\phi : \mathcal{K} \rightarrow \mathcal{K}'$ is a *morphism of simplicial complexes* if for every cell σ and σ' of $\mathcal{K}$:

$$\sigma \prec \sigma' \Rightarrow \phi(\sigma) \prec \phi(\sigma'),$$
$$dim_{\mathcal{K}'}(\phi(\sigma)) = dim_{\mathcal{K}}(\sigma).$$

These two conditions preserve, respectively, the neighborhood between simplices and their dimension. In other words, a morphism of complexes is a function in which structure is preserved. Two complexes are said to be *isomorphic* if they have exactly the same structure (i.e., ϕ is bijective).

A morphism between the support complexes of two simplicial collections provides a way to modify values labeling the simplices. Let $\mathcal{K}$ and $\mathcal{K}'$ be two simplicial collections and $\phi : |\mathcal{K}| \rightarrow |\mathcal{K}'|$ a morphism of complexes from the support complex of $\mathcal{K}$ into the support complex of $\mathcal{K}'$. We denote as $\mathcal{K}^{\phi}$ the simplicial collection having the support complex $|\mathcal{K}|$ such that for every simplex σ of $|\mathcal{K}|$:

$$\mathcal{K}^{\phi}(\sigma) = \mathcal{K}'(\phi(\sigma)). \tag{1}$$

Structural inclusion enables us to formulate how one complex can be embedded into a second one. A structural inclusion of a complex $\mathcal{K}$ in complex $\mathcal{K}'$ is an injective morphism from $\mathcal{K}$ into $\mathcal{K}'$. A morphism of complexes is *injective* if $\forall \sigma, \sigma' \in \mathcal{K}, \phi(\sigma) = \phi(\sigma') \Rightarrow \sigma = \sigma'$. Injectivity enables one to distinguish in $\mathcal{K}'$ a subcomplex that has the same structure as $\mathcal{K}$. We thus say that $\mathcal{K}$ is structurally included in $\mathcal{K}'$. Figure 3 illustrates two different ways in which a complex may be structurally included in another.

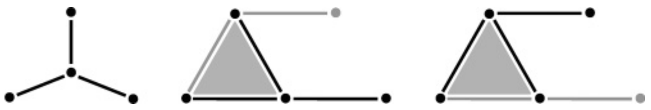


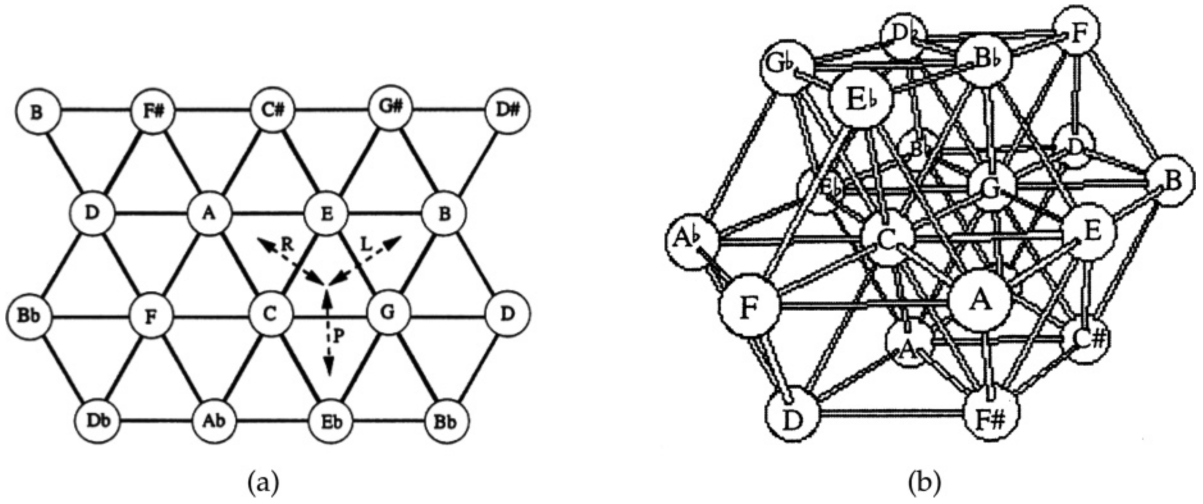
Figure 3

Finally, every automorphism in a complex defines a structural inclusion into itself. The set of automorphisms of a complex represents its *structural symmetries*.

Musical Representations

In this section, we present two well-known music-theoretical constructions: the *Tonnetz* and the catalog of T/I classes. The first notion refers to a very particular organization of pitches, which will be generalized by using the second notion, providing a classification of musical chords with respect to an equivalence relation. (We will use the mathematical term "up to transposition/inversion relation" to indicate the underlying group-theoretical basis of this equivalence relation, without entering into more technical aspects related to algebraic combinatorics and group actions.)

Figure 4. The neo-Riemannian Tonnetz *and the three neo-Riemannian operations* P, L, *and* R *(a), and a three-dimensional derivation of the* Tonnetz *by Edward Gollin (1998) enabling the representation of dominant seventh and half-diminished chords by tetrahedra (b).*



The *Tonnetz*

One of the strongest motivations for this work is to better understand some mathematical properties of the *Tonnetz*. This construction represents a widely used tool in music theory, analysis, and composition, and it refers to a symbolic organization of pitches in the Euclidean space following infinite axes associated with particular musical intervals. It was independently investigated by Leonhard Euler (1739) and Jean-Philippe Rameau (1750) for acoustical purposes with the aid of similar two-dimensional representations, organizing pitches in just intonation along an axis of pure fifths and an axis of pure major thirds. This representation was rediscovered later by musicologists Arthur von Oettingen and Hugo Riemann, and this enabled the music-theoretical community to generalize the application domain of this construction and to adapt it to the case of twelve-tone equal temperament. More recently, music theorists have shown a strong interest in this model, in particular for the representation of typical post-romantic chord progressions (Cohn 2012) or neo-Riemannian transformations, according to the transformational paradigm proposed by David Lewin in the late 1980s (see Lewin 2007). This model has also been used in musical composition, from contemporary music experiments (Chouvel 2009) to stylistic practices more characteristic of popular music (Bigo and Andreatta 2014).

The neo-Riemannian *Tonnetz* (see Figure 4a) is a graph in which pitches are organized along the intervals of the fifth (horizontal axis), and the major and minor thirds (diagonal axes). This representation has the interesting property of revealing major and minor triads as triangles. The three arrows illustrate the neo-Riemannian operations *P* (parallel), *L* (leading-tone), and *R* (relative), which describe the different transitions between pairs of triads having two common notes. Music theorists have investigated different derivations of the *Tonnetz* by adding other axes, allowing the representation of chords having more than three elements. For instance, Figure 4b illustrates a *generalized Tonnetz* represented by a three-dimensional *Tonnetz* space (Gollin 1998). This model adds new interval axes to the two-dimensional representation of the traditional *Tonnetz* in such a way that dominant seventh and half-diminished chords are represented by tetrahedra pointing upwards (for the dominant seventh chord) or downwards (for the half-diminished chord). Three-dimensional models are well suited to studying progressions of four-note chords. Note that the generating axes do not necessarily need to label chromatic intervals. One may build diatonic versions of the *Tonnetz* by associating axes with intervals that are diatonic instead of chromatic (*diatonic Tonnetze*). In this way, vertices and shapes only represent notes and chords belonging to a single tonality.

Our work is based on the assumption of twelve-tone equal temperament and octave reduction (i.e., we are dealing with *pitch classes*, without consideration of octaves). For example, the notes C♯3, C♯4, and D♭4 are all considered to be representatives of the same pitch class. In particular, what we call a *Tonnetz* refers more specifically to the pitch-class *Tonnetz*. In this context, the graph in Figure 4a infinitely repeats the twelve pitch classes along its axes. An important consequence of this representation is that every pitch class is represented in multiple locations. However, the methods presented here could be applied in more general contexts (e.g., just intonation, octave distinction). For example, the Max object `bach.tonnetz`, presented in the final section of this article, does not respect the hypothesis of octave identification, which is commonly and tacitly admitted in most *Tonnetz*-based analyses.

Generalized *Tonnetze* and T/I Classes

By considering the role played by particular chords (minor and major chords as triangles in the neo-Riemannian *Tonnetz*, and dominant seventh and half-diminished chords as tetrahedra in the three-dimensional *Tonnetz*), one can argue that the starting point for the construction of a *Tonnetz* is more related to the choice of a set of chords than to a set of interval axes. This observation has been our starting point for studying generalized versions of the *Tonnetz* with the help of the notion of an equivalence relation up to a given musical transformation. In the two earlier examples, the chords represented in a *Tonnetz* are all equivalent up to transposition and inversion (i.e., they belong to the same T/I class). More algebraically, saying that two chords belong to the same equivalence class corresponds tacitly to the assumption that there exists a group structure that induces the equivalence relation. Two chords A and B are said to belong to the same T/I class if there is an element of the group generated by transpositions and inversions that transforms A into B (and, conversely, although via a different transformation, B into A, thanks to the symmetry property of the equivalence relation). This group is known as the *dihedral group* $\mathbb{D}_N$, and the previous sentence can be reformulated, as we suggested previously, in terms of equivalence classes (or "orbits") induced by an underlying group transformation (more precisely a group action, as defined, for example, by Andreatta and Agon 2003).

This fact is surely true for all the generalized *Tonnetze* proposed in our approach, but the reader should pay attention to the fact that some equivalence relations can be "non-paradigmatic," meaning that it is not possible to easily find a group structure in such a way that two chords belonging to the same equivalence class are equivalent up to a transformation of the underlying group. This is, for example, the case of the music-theoretical relationship known as the Z-relation, for which the possible group-induced equivalence is still an open problem (see Mandereau et al. 2011). A T/I class is usually identified by the intervallic structure that is shared by all the chords of the class, eventually by considering some cyclic permutations or retrograde versions of its elements. For instance, major and minor chords correspond to the intervallic structures $[4, 3, 5]$ and $[3, 4, 5]$, respectively, where the second structure is clearly obtained by a retrograde reading of the elements of the first structure (starting from the second element). They therefore both share the intervallic structure $[3, 4, 5]$, because the row of intervals between pitch classes they result from is composed of a minor third (three semitones), a major third (four semitones), and a perfect fourth (five semitones). Intervals are not ordered in the same direction for major ($[4, 3, 5]$) and minor chords ($[3, 4, 5]$). With the same reasoning, dominant seventh and half-diminished chords are identified by the intervallic structure $[2, 3, 3, 4]$. Note that the elements of the intervallic structure add up to the number of steps N dividing the octave, e.g., $N = 12$ in the chromatic system and $N = 7$ in the diatonic system.

T/I classes can be easily calculated, which for $N = 12$ leads to the catalog of 224 pitch-class sets (Forte 1973), also known as "Forte classes." In the diatonic system ($N = 7$), which divides the octave into seven parts (of unequal size), there exist 18 such classes. Following this line of thought, we now show how it is possible to automatically build

Figure 5. Representing notes and chords with simplices (a). The complex (b) corresponds to the C-major chord and all 2-note chords and individual notes included within it.

Figure 6. A region of the complex $\mathcal{K}[3,4,5]$ composed of major and minor chords (a) and the complex $\mathcal{K}[2,3,3,4]$, composed of dominant seventh and half-diminished chords (b).

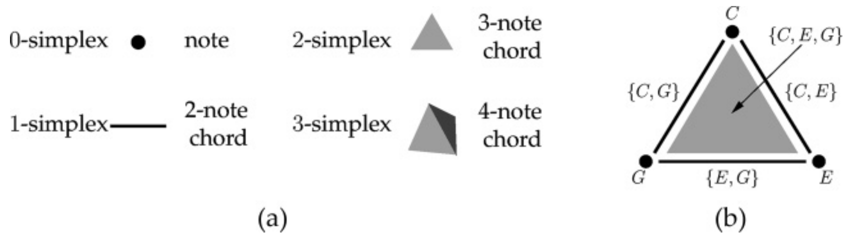


Figure 5

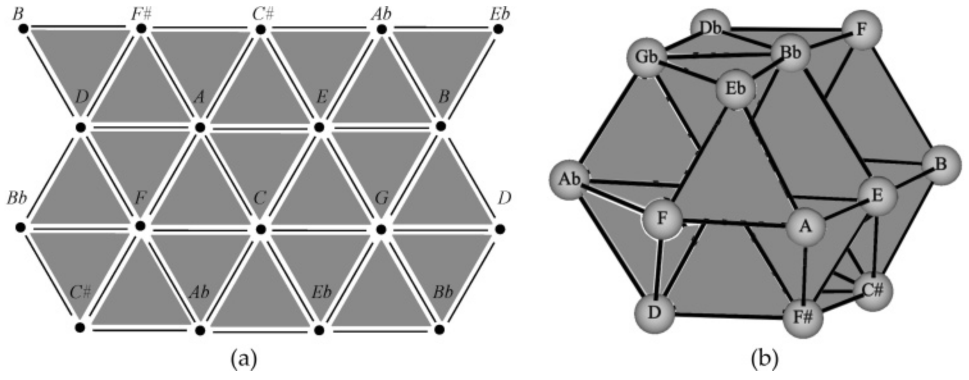


Figure 6

the generalized *Tonnetze* associated with the 224 T/I chromatic classes and 18 T/I diatonic classes.

Chord Complexes and Trajectories

In this section we introduce the representation of a musical sequence by a *trajectory* in a chord space. Chord spaces are inspired by the *Tonnetz* and formalized as simplicial complexes. Trajectories are sequences of regions of these complexes.

Chord-Based Complexes

In the following, we call a set of pitch classes a *chord*. This means that we make abstractions of some parameters, such as duration and octave position of the notes.

Generalized Tonnetze *as Simplicial Collections*

We use a method presented by the first author to represent chords as simplices (see Bigo 2013). In this article, the term "*n*-note chord" refers to a chord containing *n* pitch classes (not *n* notes in the usual sense), which can be represented by a $(n-1)$-simplex. A 0-simplex represents a single pitch class, a 1-simplex represents a 2-note chord (or interval), a 2-simplex represents a 3-note chord (or triad), etc. (see Figure 5a). Figure 5b illustrates a simplicial collection representing the C-major chord. It includes seven simplices representing each subchord of C-major (including three individual pitch classes, which are 0-simplexes, three intervals, which are represented as 1-simplexes, and a 2-simplex representing the entire chord as a triangle).

We represent a generalized *Tonnetz* as a simplicial collection composed of *n*-simplices representing the chords of a given T/I class. In the following, we denote as $\mathcal{K}[a_1,\ldots,a_i]$ the complex associated with the T/I class identified by the intervallic structure $[a_1,\ldots,a_i]$. Figure 6 illustrates regions of the complexes $\mathcal{K}[3,4,5]$ and $\mathcal{K}[2,3,3,4]$. They correspond to the two graphs of Figure 4, in which 2-simplices and 3-simplices have been integrated.

In other words, the neo-Riemannian *Tonnetz* and the three-dimensional *Tonnetz* correspond to the 1-skeletons of $\mathcal{K}[3,4,5]$ and $\mathcal{K}[2,3,3,4]$, respectively. As previously mentioned, dominant seventh and half-diminished chords are represented by tetrahedra pointing up or down, respectively, although the tetrahedra are not fully visible in this representation. The reader is referred to the skeleton in Figure 2 to see all the pitch classes.

Chord Complex Construction

The construction of the chord complex $\mathcal{K}[a_1,\ldots,a_i]$ proceeds as follows. First, an n-note chord belonging to the class identified by the intervallic structure $[a_1,\ldots,a_i]$ is chosen and represented by an $(n-1)$-simplex (for example, the C-major chord illustrated in Figure 5b for the class $[3,4,5]$). The simplex is then embedded in an equilateral manner in the $(n-1)$-dimensional Euclidean space. For a 3-note chord, this space is the Euclidean plane, and the chord is embedded as an equilateral triangle. The directions given to the 1-simplices (i.e., edges) define axes associated with particular intervals, as in the *Tonnetz*. Then, simplices are naturally replicated along these axes, in such a way that the chords represented respect the transpositions induced by the intervals along these axes. The transposition is chromatic or diatonic, depending on the T/I class. Note that the simplices of a complex associated with a diatonic T/I class only represent pitch classes and chords belonging to a single tonality (i.e., key).

A consequence of this generic method of construction is that two complexes associated with chord classes of the same size are isomorphic. For example, the two complexes $\mathcal{K}[3,4,5]$ (see Figure 6a) and $\mathcal{K}[2,3,7]$ (see Figure 7) are both two-dimensional infinite triangular tessellations.

Representation of a Musical Sequence in a Chord Complex

In this section we present the representation of a musical sequence by a trajectory in a chord complex.

A trajectory is a sequence of regions of the complex. Note that it is not necessarily continuous

Figure 7. A region of the complex $\mathcal{K}[2,3,7]$ isomorphic to the complex $\mathcal{K}[3,4,5]$ illustrated in Figure 6a.

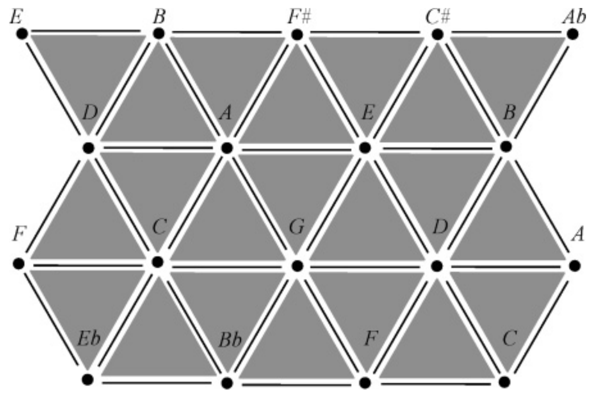


Figure 7

Figure 8. Segmentation of a sequence based on the set of pitch classes played. The duration unit corresponds here to the quarter note.

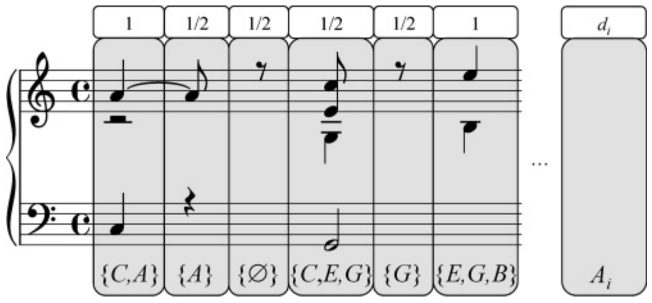
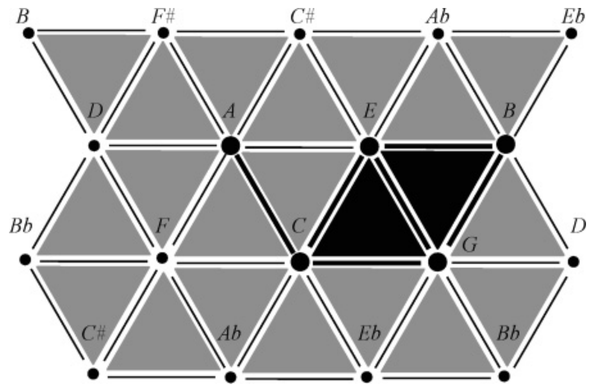


Figure 8

in space, which means that successive regions do not need to be spatial neighbors. Each successive region represents a temporal segment of a musical composition. In our work, we use a very simple segmentation method, based on the appearance and disappearance of pitch classes. Each time a pitch class enters or leaves the set of played notes, the current segment stops and a new one begins. This principle is illustrated in Figure 8. Each segment is characterized by its duration relative to the other segments. Thus, we reduce a musical sequence P to a sequence of pitch-class sets, each labeled by a relative duration. So we have $P = [(A_0,d_0),\ldots,(A_N,d_N)]$, where A_i is the set of *active* pitch classes with a duration d_i.

As previously noted, each pitch class is represented in multiple locations in the pitch-class *Tonnetz*. In the same way, an n-note chord is

Figure 9. Trace (in black) of a trajectory representing the sequence illustrated in Figure 8 in $\mathcal{K}[3,4,5]$.



represented by multiple $(n-1)$-simplices in a chord complex.

A trajectory in a chord complex $\mathcal{K}$ is a sequence of subcollections of $\mathcal{K}$, which are all labeled by a duration. Let $T_{\mathcal{K}} = [(\mathcal{K}_0, d_0), \ldots, (\mathcal{K}_N, d_N)]$ be a trajectory in $\mathcal{K}$ and $P = [(A_0, d'_0), \ldots, (A_N, d'_N)]$ a musical sequence. We say that the trajectory $T_{\mathcal{K}}$ represents a musical sequence P if for every i, $d_i = d'_i$ and $\mathcal{K}_i$ is a simplicial subcollection such that:

$$\forall \sigma \in |\mathcal{K}_i|, \quad \mathcal{K}(\sigma) \subseteq A_i \tag{2}$$

In other words, the subcollection $\mathcal{K}_i$ only represents the set of pitch classes present in the i^{th} segment of the sequence. The *trace* of a trajectory $T_{\mathcal{K}}$ is the subcollection $\mathcal{T} \subseteq \mathcal{K}$ constituted by the simplices included in $T_{\mathcal{K}}$:

$$\mathcal{T} = \bigcup_{(\mathcal{K}_i, d_i) \in T_{\mathcal{K}}} \mathcal{K}_i \tag{3}$$

Figure 9 illustrates the trace of a sequence representing the sequence illustrated in Figure 8 in the chord complex $\mathcal{K}[3,4,5]$. Because each chord is represented in multiple locations in the complex $\mathcal{K}$, the definition enables a large number of different trajectories to represent a given sequence P within the specific complex. To automatically attribute a trajectory to a sequence, we use an algorithm based on two main criteria:

1. Chords must be represented as compact subcomplexes.
2. Chord transitions must correspond to small movements.

For a complete description of the algorithm, please refer to the first author's doctoral work (Bigo 2013).

Transformations of Trajectories

Several musical transformations can be interpreted as spatial operations on trajectories. The list of musically relevant operations includes translations, rotations, and embeddings of the trajectory in a new support space. Some of these transformations correspond to well-known musical operations, as in the case of translations (which are equivalent to musical transpositions) and rotations (which can be interpreted as inversions). The case of the embedding of a given trajectory in a new support space is interesting, because it does not correspond to any currently familiar musical transformation. The musical interpretation of this geometric transformation is therefore an open problem in computational musicology.

Transformation of a Sequence

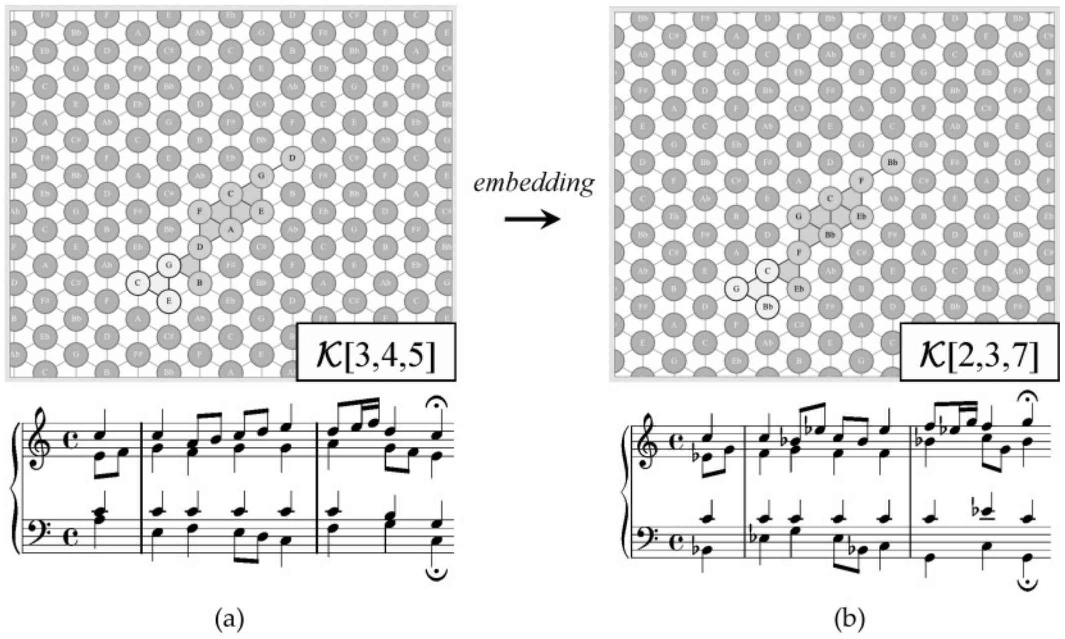
Let P be a musical sequence, $\mathcal{K}$ and $\mathcal{K}'$ two chord complexes, and $\mathcal{T} \subset \mathcal{K}$ the trace of a trajectory $T_{\mathcal{K}}$ that represents the sequence P in $\mathcal{K}$. Let ϕ be a structural inclusion of $|\mathcal{T}|$ in $|\mathcal{K}'|$. The morphism ϕ allows a relabeling of the simplices of $\mathcal{K}$ to shape a trace $\mathcal{T}^{\phi}$ in $\mathcal{K}'$. This modification of labels then induces a transformation $\mathcal{T}_{\phi}$ of the sequence P into a different sequence P' defined by:

$$\mathcal{T}_{\phi}(A_i, d_i) = \\ (\{n \in \mathbb{Z}_N \mid \exists\, \sigma \in \mathscr{S}_0(\mathcal{K}_i), \mathcal{T}^{\phi}(\sigma) = n\}, d_i),$$

where $\mathcal{K}_i$ represents A_i in $T_{\mathcal{K}}$.

The notation $\mathcal{T}_{\phi}$ stresses the fact that the transformation depends only on the function ϕ (not on the sequence P). We observe that the transformation $\mathcal{T}_{\phi}$ can be applied just on the vertices to determine the new set of pitch classes. To produce a new sequence P' from the new segments given by $\mathcal{T}_{\phi}$, it is necessary to provide the octave information for each transformed pitch class. In the next examples, we choose the octave of the transformed pitch class

Figure 10. The first measures of the chorale "Ach lieben Christen, seid getrost," by J.S. Bach (BWV 256), represented by a trajectory in $\mathcal{K}[3,4,5]$ *(a), and the transformation of the sequence resulting from the embedding of the trajectory in* $\mathcal{K}[2,3,7]$ *(b).*



in a way that the distance from the original pitch is minimized. Then a transformation affects pitch classes with only a minimal alteration of the pitch register in which the new sequence is evolving. Note that this implicitly makes use of the equivalence between the toroidal structure of the *Tonnetz* and its multiple representations as selected regions of the infinite bidimensional space, according to the number of octaves we want to take into account. Furthermore, as this work concentrates on pitch transformations, segment durations are left unchanged.

Isomorphism between Two Support Spaces

Let $\mathcal{K}_1$ and $\mathcal{K}_2$ be two chord complexes, and ϕ a structural inclusion of $|\mathcal{K}_1|$ in $|\mathcal{K}_2|$. It is easy to see that the function ϕ can be applied to any trajectory in $\mathcal{K}_1$. This kind of transformation can be understood intuitively as embedding a trajectory coming from one complex into another one. In particular, every isomorphism between two complexes $\mathcal{K}_1$ and $\mathcal{K}_2$ enables a given trajectory built in one of the complexes to be embedded into the second.

As mentioned in the section "Chord Complexes and Trajectories," the property of having the same dimension is sufficient to establish an isomorphism between two chord complexes $\mathcal{K}[a_1, \ldots, a_i]$. For example, $\mathcal{K}[3,4,5]$ and $\mathcal{K}[2,3,7]$ are isomorphic, because they both result from the infinite repetition of 2-simplices along axes in three directions. A natural consequence is that any trajectory built in one of these complexes can be embedded into the other.

Figure 10 illustrates this transformation with the first measures of the chorale "Ach lieben Christen, seid getrost," by J.S. Bach (BWV 256). The trajectory in Figure 10a represents the sequence in $\mathcal{K}[3,4,5]$. In this complex, triangles represent major and minor chords. In Figure 10b the same trajectory is embedded into $\mathcal{K}[2,3,7]$, in which triangles represent "incomplete minor seventh chords" (i.e., minor seventh chords without the fifth, and minor or dominant seventh chords without the third). These chords have the interesting property of including typical intervals of the pentatonic scale, which gives a particular color to the transformed sequence. The result of the transformation is available online in MIDI format as supplementary content on the MIT Press Web site (http://mitpress.mit.edu/10.1162/COMJ_a_00312).

Embedding a trajectory into a chord complex, when the trajectory was built in another chord complex, gives a musical sequence a new harmonic color, with conservation of its characteristic shape. In particular, embedding a trajectory into a diatonic complex has the obvious consequence of giving the transformed sequence the tonality characterizing the complex.

Automorphism in a Support Space

When the chord complex $\mathcal{K}$ includes structural symmetries, the associated automorphisms define isometries that can be applied to any trajectory of $\mathcal{K}$. By definition, a complex built from a T/I class is structurally included into itself at least N times (where N is the number of pitches into which the octave is divided). Indeed, the construction method described earlier ensures that for any pitch-class set represented in the complex, its $N-1$ transpositions are represented as well. For a T/I class including 3-note chords, the symmetries of the corresponding complex (which is a triangular tessellation) can be associated intuitively with the possible "simplex-to-simplex" superpositions of two copies of the complex, after any translations or rotations. The numerous symmetries in T/I chord complexes enable a large number of distinct transformations for a given trajectory. Some of these transformations can intuitively be interpreted as discrete translations or rotations of the trajectory. Some musical transformations produced by automorphisms in chord complexes are available at the MIT Press URL listed previously.

Discrete Translations

Let $\mathcal{K}$ be a complex and σ_1 and σ_2 two 0-simplices of $\mathcal{K}$. The translation ϕ, which transforms σ_1 into σ_2, is characterized by the interval class j that transforms the pitch classes associated with these vertices by the quantity $j = \mathcal{K}(\sigma_2) - \mathcal{K}(\sigma_1)$. We observe that for any vertex σ labeled by the pitch class n, the transformed vertex $\phi(\sigma)$ will be labeled by the pitch class $n+j$. We thus have for any sequence P:

$$\mathcal{T}_\phi(A_i, d_i) = (\{(n+j) \bmod N \mid n \in A_i\}, d_i) \quad (4)$$

The application of a translation on a trajectory in a complex associated with a T/I class corresponds to a transposition. In the case of $N = 12$, the translation of a trajectory corresponds to a chromatic transposition. On the other hand, a translation processed in a diatonic complex (in which case $N = 7$) produces a diatonic (or modal) transposition. If the sequence belongs to a tonality, a translation in the associated diatonic complex will achieve a change of mode (e.g., a major-to-minor transformation).

Discrete Point Reflections

A discrete point reflection in a complex has the consequence of transforming intervals into their inversions. Indeed, every direction is associated with a particular interval, and the point reflection reverses directions. The pitch class m, labeling the center vertex of the point reflection, is unchanged by the transformation. The interval separating m from a pitch class is inverted to produce the new pitch class. As with translations, a pitch class is transformed according to its value, not according to the position of its simplex in the complex. We thus have for a given sequence P:

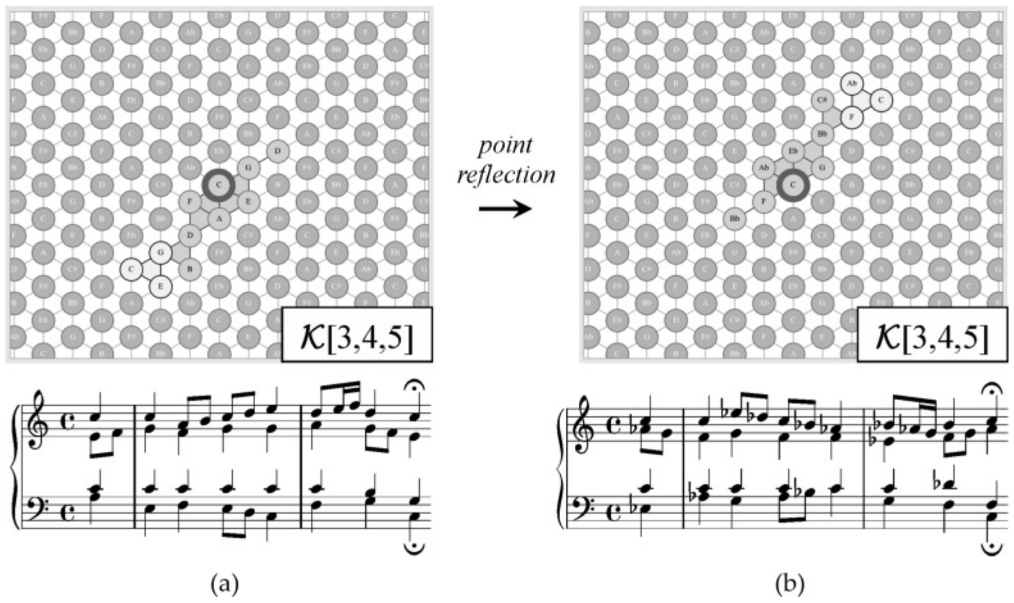
$$\mathcal{T}_\phi(A_i, d_i) = (\{(m-n) \bmod N \mid n \in A_i\}, d_i) \quad (5)$$

Figure 11 illustrates a point reflection applied on a trajectory in $\mathcal{K}[3,4,5]$. The center of the point reflection is a vertex labeled by the pitch class C. The result of a point reflection is a pitch-class inversion. In chromatic and diatonic complexes, these inversions are, respectively, chromatic and modal.

Other Transformations

As discussed earlier, the two preceding transformations have the property of producing pitch classes entirely determined by their original values, not by the positions of their vertices. Moreover, these two spatial transformations result in well-known musical operations (transpositions for translations and inversions for point reflections). On the other hand, some other automorphisms cannot be simply

Figure 11. A trajectory representing the first measures of the Bach chorale BWV 256 in $\mathcal{K}[3,4,5]$ (a). A point reflection is applied on the trajectory and produces a new sequence (b).



specified as transformations of pitch classes. For example, line symmetries or rotations can transform vertices labeled with the same pitch class into vertices labeled with different pitch classes. The same generally applies when embedding a trajectory into a new chord complex. These transformations do not have any conventional musical interpretation, to the best of our knowledge, and they result in new musical operations. By leaving the realm of music analysis, this clearly shows the potential of these tools in music composition, particularly when these tools are integrated into a computer-aided composition environment, as described in the next section.

Computer Programs

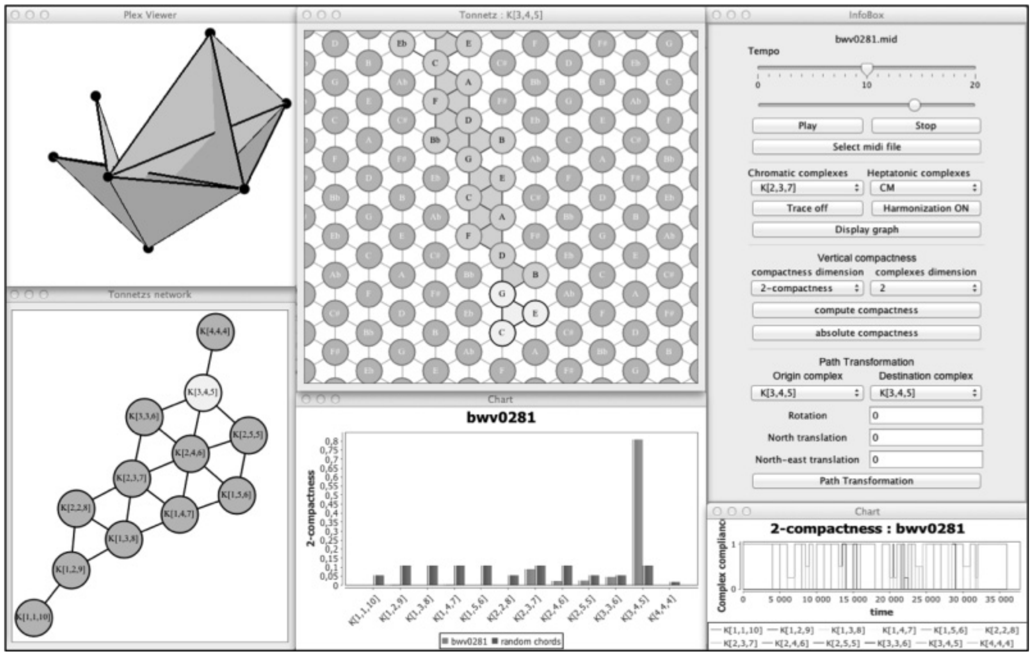
In this section, we present two software applications enabling one to work with the notions presented in the previous sections: HexaChord, which is mostly devoted to music analysis, and the Max object `bach.tonnetz`, which is dedicated to computer-aided music composition.

HexaChord

HexaChord (at www.lacl.fr/~lbigo/hexachord) is a computer-aided music analysis environment based on the spatial representations discussed in this article. The software provides a three-dimensional visualization of the complex representing any set of chords. Chord complexes related to a T/I class that groups 3-note chords in diatonic and chromatic scales are represented as infinite triangular tessellations (as in Figures 10 and 11). HexaChord also includes other musical representation spaces, such as the chromatic circle, the circle of fifths, and a voice-leading space that spatially organizes chord intervallic structures of size 3, depending on their voice-leading proximity (Tymoczko 2011). Figure 12 illustrates the graphical user interface of the software.

Musical pieces are imported as MIDI files. An algorithm described by the first author is used to compute a trajectory for any pair of musical pieces and chord complexes (Bigo 2013). The different visualization spaces are updated in real time when a file is played. The musical sequence is then represented as a path that evolves in complexes

Figure 12. Graphical user interface of HexaChord.



bwv0281.mid
Tempo: 10
Chromatic complexes: K[2,3,7]
Heptatonic complexes: CM
Vertical compactness
compactness dimension: 2-compactness
complexes dimension: 2
Path Transformation
Origin complex: K[3,4,5]
Destination complex: K[3,4,5]
Rotation: 0
North translation: 0
North-east translation: 0

chosen by the user. The spatial harmonization functionality highlights at any instant of the piece the pitch class (or classes) not played but having the highest number of neighborhood relationships with the played pitch classes. HexaChord also accepts input from any MIDI device, including MIDI keyboards. The interface then displays the trajectory corresponding to the performance of the player in any chord complex and in real time.

The transformations described in the section "Transformations of Trajectories" can be applied to any trajectory in a planar chord complex. For each transformation, the user specifies: (1) the reference complex in which the trajectory is initially computed; (2) a destination complex in which the trajectory is going to be embedded; (3) an angle of rotation; and (4) two vector coordinates (e.g., "North" and "Northeast"), enabling one to specify any translation in a triangular lattice.

The interface allows one to compose one rotation, one translation, and one embedding in a unique transformation. To proceed by a rotation or a translation only, the embedding complex will have to be the same as the reference complex. Also, to proceed with an embedding without any additional geometrical transformation, translation and rotation fields will have to be set to zero. The musical sequence resulting from any transformation can be exported to a MIDI file. The examples included as supplementary content at the URL listed earlier were generated using HexaChord.

Other features dedicated to analysis have been integrated in the application. For instance, HexaChord automatically determines the chord complex that is the most suitable for representing a musical sequence. This task is connected to the notion of *compliance* (Bigo et al. 2013) and is achieved by comparing the compactness of the trajectories representing the piece in the different complexes. The hypothesis behind this functionality is that the high compactness of the trajectory in a particular complex might be seen as a stylistic signature of the piece. Following this idea, the computation of the compactness has been experimented with, yielding interesting preliminary results in music classification (Bigo 2013).

Although it was initially conceived to assist musical analysis, we believe that HexaChord provides interesting pedagogical features. The visual aspects of the chord complexes give an intuitive understanding of some mechanisms of harmony, particularly in the context of modern set theory. Also, the MIDI device functionality is a valuable utility for the composer who uses such spaces as a tool for their compositions (Chouvel 2009). Following this line, HexaChord has inspired the development of the computer-aided system *PaperTonnetz*, which enables the user to manipulate a trajectory in a chord complex as a central object in the composition process (Bigo et al. 2012).

The bach.tonnetz Object

If one needs to deal with *Tonnetz* representations interactively, a natural solution is to handle them in a real-time environment. An easy way to do this is to take advantage of the bach library (on the Web at www.bachproject.net), a set of externals and patches for Max, bringing computer-assisted composition into the real-time world (Agostini and Ghisi 2012, 2013, 2015). Among its features, the bach library has a subset of tools dedicated to musical representations, including the `bach.tonnetz` object, which implements and displays a *Tonnetz* centered on a given pitch and generated by two given diatonic intervals. Nodes in the lattice can be selected interactively (via mouse and keyboard), or via incoming messages, containing information in any of several formats: cents, note names, pitch classes, diatonic intervals, or coordinates in the lattice space. Elementary transformations, such as translations and rotations as described in the section "Transformations of Trajectories," can be easily performed both via the user interface and via Max messages.

The `bach.tonnetz` object can easily echo the incoming data to its outlets, to allow real-time modification of the point coordinates or of the lattice properties. As a result, it is fairly straightforward to take any incoming flow of notes and rotate or shift it. To allow a more faithful representation of MIDI data, each selected node in the lattice can be given a velocity value, which can in turn be visualized graphically, either by varying node colors or by adjusting node sizes. Moreover, `bach.tonnetz` also supports purely diatonic networks, microtonal intervals, and just intonation.

Because pitches are not necessarily unique inside the lattice, a matching mode is available to determine which point in the lattice should be selected when more than one choice is possible. Three possibilities are available: the most central match, the match nearest to the last selected point, or a modification of the latter that prevents the selected points from exiting the visual rectangle.

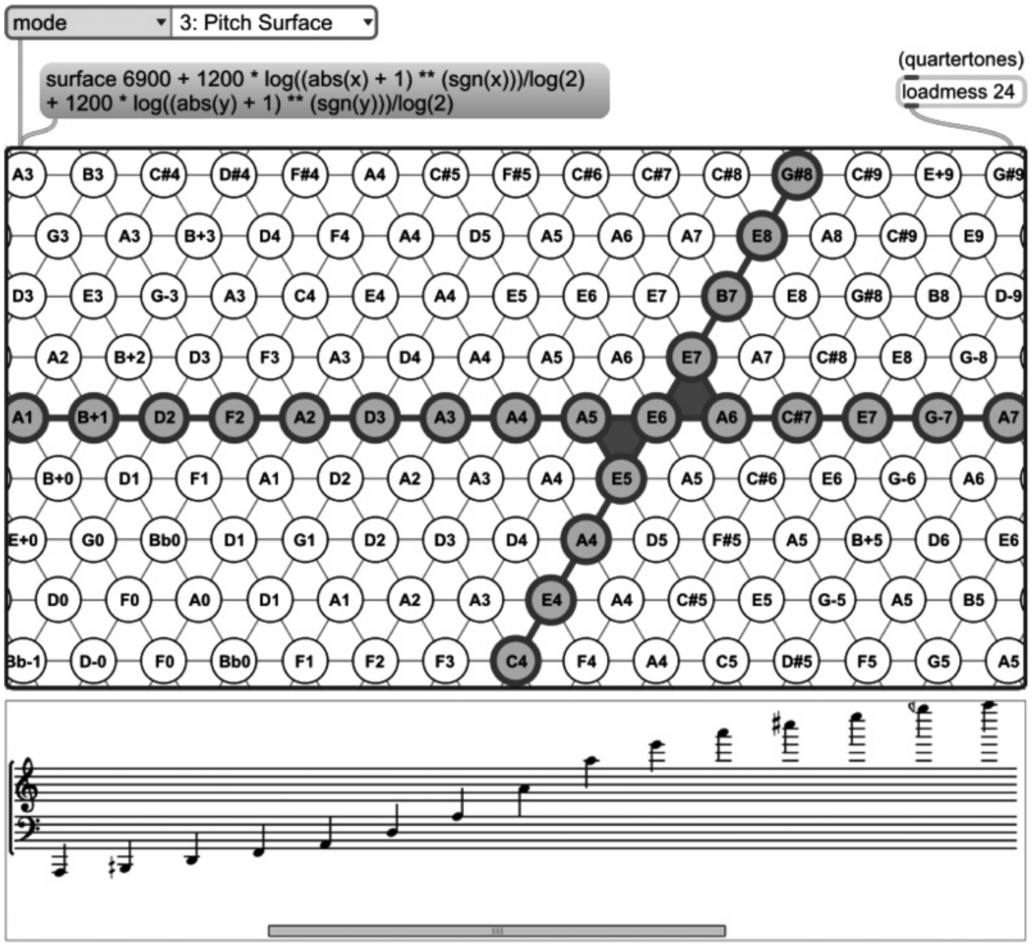
In addition, `bach.tonnetz` can be set in "surface mode," so that the pitch at each lattice point can be specified via an equation. This allows one to work with general pitch surfaces $f(x, y)$, sampled on an hexagonal lattice (with $x, y \in \mathbf{N}$ being the coordinates corresponding to two axes in the hexagonal lattice). The traditional *Tonnetz* corresponds to the special case $f(x, y) = \iota_1 x + \iota_2 y$, where ι_1 and ι_2 are the generating intervals in cents. Another interesting choice for the equation is

$$f(x, y) = b + \frac{1200}{\log 2} \log((|x| + 1)^{\mathrm{sgn}(x)}(|y| + 1)^{\mathrm{sgn}(y)}), \tag{6}$$

which constructs a sort of frequency-domain *Tonnetz*, created by a matrix of intersecting harmonic and subharmonic series and having the pitch b (in cents) as the central pitch (see Figure 13).

Because `bach.tonnetz` is real-time oriented, it is also an ideal tool to handle performative and generative processes. As an example, one can build a patch implementing two-dimensional cellular automata (such as Conway's Game of Life, or one of its adaptations for hexagonal grids; see Figure 14). This is made even easier by the cage library (a set of high-level abstractions based on the bach library, available at www.bachproject.net/cage). This library contains the Max object `cage.life`,

Figure 13. A frequency-domain Tonnetz, centered on A4 and having the combination of harmonic and subharmonic series as rows and as diagonals. The display is in quarter tones (quarter-sharp accidentals are denoted using a plus sign, quarter-flat accidentals with a minus). The musical notation at the bottom of the image shows the main harmonic and subharmonic series of the horizontal axis.



mode: 3: Pitch Surface

```
surface 6900 + 1200 * log((abs(x) + 1) ** (sgn(x)))/log(2)
+ 1200 * log((abs(y) + 1) ** (sgn(y)))/log(2)
```

(quartertones)

```
loadmess 24
```

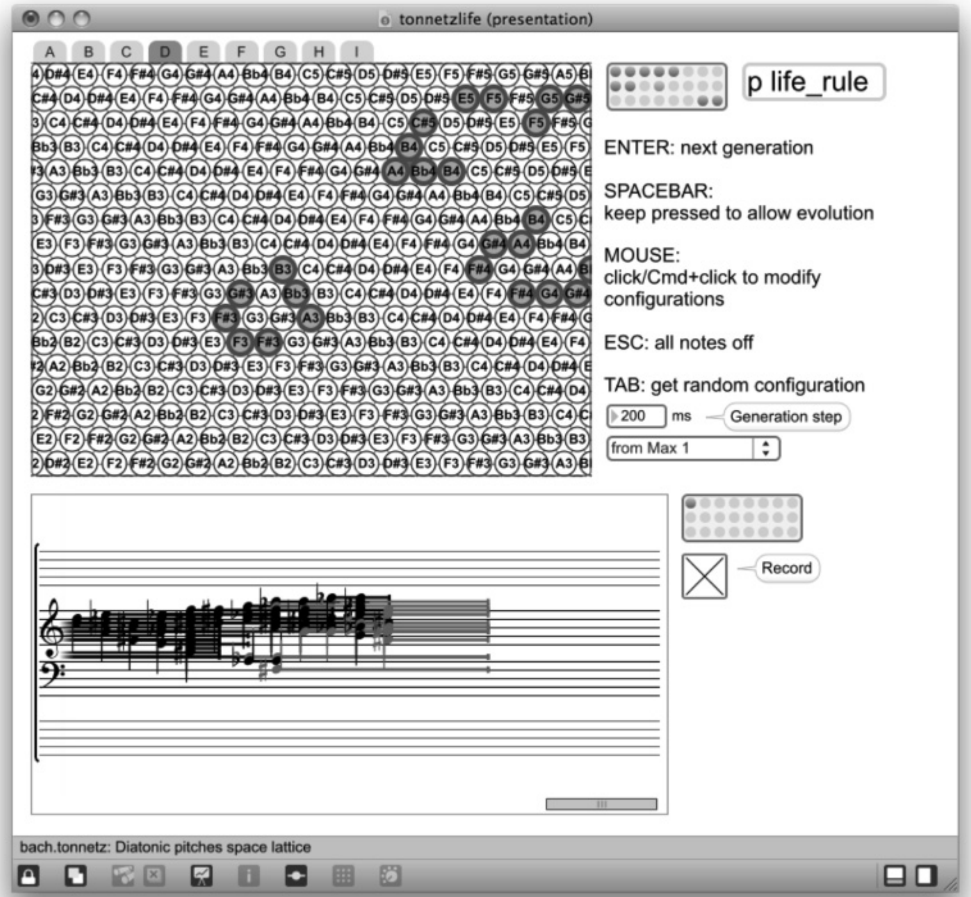
A3 B3 C#4 D#4 F#4 A4 C#5 F#5 C#6 C#7 C#8 G#8 C#9 E+9 G#9
G3 A3 B+3 D4 F4 A4 D5 A5 A6 A7 E8 A8 C#9 E9
D3 E3 G-3 A3 C4 E4 A4 E5 E6 E7 B7 E8 G#8 B8 D-9
A2 B+2 D3 F3 A3 D4 A4 A5 A6 E7 A7 C#8 E8 G-8
A1 B+1 D2 F2 A2 D3 A3 A4 A5 E6 A6 C#7 E7 G-7 A7
B+0 D1 F1 A1 D2 A2 A3 A4 E5 A5 C#6 E6 G-6 A6
E+0 G0 Bb0 D1 G1 D2 D3 D4 A4 D5 F#5 A5 B+5 D6 E6
D0 F0 A0 D1 A1 A2 A3 E4 A4 C#5 E5 G-5 A5 B5
Bb-1 D-0 F0 Bb0 F1 F2 F3 C4 F4 A4 C5 D#5 F5 G5 A5

producing two-dimensional cellular automata (Agostini, Daubresse, and Ghisi 2014), and includes two abstractions, `cage.tonnetz.flip` and `cage.tonnetz.rot`, capable of rotating and flipping *Tonnetz* coordinates.

Conclusion

In this article we have presented a general framework for the representation of musical structures and processes in simplicial chord spaces. Generic transformations on musical sequences depend only on their spatial representations, which make use of the topological structure of simplicial complexes. Their underlying algebraic structure enables an elegant formalization of these transformations, thanks to the notion of morphism between different support spaces, which preserve dimension and neighborhoods between two complexes. Whereas some of these morphisms correspond to familiar musical operations, most of them, such as the embedding of a given trajectory into a new chord complex, still await suitable musical interpretation. Most of the concepts introduced in this article have been implemented in HexaChord and the bach library, two environments for computer-aided music analysis and composition that explore the computational aspects of the theoretical concepts discussed in this article.

Figure 14. A patch enabling the exploration of musical material as two-dimensional automata on a Tonnetz. *Rules are defined by patching (in subpatch* `p life_rule`*), and a few basic interface commands are given to control the breeding. The output result is recorded in a* `bach.roll`*, for possible further use. The user can modify the* Tonnetz *structure and its content at any time (some presets are marked with letters in the upper part of the patch).*


tonnetzlife (presentation)
A B C D E F G H I
p life_rule
Generation step
Record
bach.tonnetz: Diatonic pitches space lattice

ENTER: next generation

SPACEBAR:
keep pressed to allow evolution

MOUSE:
click/Cmd+click to modify configurations

ESC: all notes off

TAB: get random configuration

200 ms

from Max 1

Acknowledgments

The authors are very grateful to Jean-Louis Giavitto from the RepMus team at IRCAM, Olivier Michel at LACL, Université Paris-Est, Jean-Marc Chouvel from Université de Reims, and Andrea Agostini for fruitful discussions and software developments. This article is a revised and enlarged version of the paper entitled "Spatial Transformations in Simplicial Chord Spaces" presented at the joint ICMC/SMC Conference in Athens (Bigo et al. 2014), where it won the Best Paper award. The authors would like to thank Doug Keislar, Editor of *Computer Music Journal*, for encouraging this revised and enlarged version, and for the fruitful exchanges during its preparation. This research is partially supported by the project Lrn2Cre8, funded by the Future and Emerging Technologies (FET) programme, within the Seventh Framework Programme for Research of the European Commission, under FET grant 610859.

References

Agostini, A., E. Daubresse, and D. Ghisi. 2014. "Cage: A High-Level Library for Real-Time Computer-Aided Composition." In *Proceedings of the Joint International Computer Music Conference and Sound and Music Computing Conference*, pp. 308–313.

Agostini, A., and D. Ghisi. 2012. "Bach: An Environment for Computer-Aided Composition in Max." In *Proceedings of the International Computer Music Conference*, pp. 373–378.

Agostini, A., and D. Ghisi. 2013. "Real-Time Computer-Aided Composition with bach." *Contemporary Music Review* 32(1):41–48.

Agostini, A., and D. Ghisi. 2015. "A Max Library for Musical Notation and Computer-Aided Composition." *Computer Music Journal* 39(2):11–27.

Andreatta, M., and C. Agon. 2003. "Implementing Algebraic Methods in OpenMusic." In *Proceedings of the International Computer Music Conference*, pp. 377–384.

Bigo, L. 2013. "Représentations symboliques musicales et calcul spatial." PhD dissertation, Université Paris-Est, Paris.

Bigo, L., and M. Andreatta. 2014. "A Geometrical Model for the Analysis of Pop Music." *Sonus* 35(1):36–48.

Bigo, L., et al. 2012. "PaperTonnetz: Music Composition with Interactive Paper." In *Proceedings of the Sound and Music Computing Conference*, pp. 219–225.

Bigo, L., et al. 2013. "Computation and Visualization of Musical Structures in Chord-Based Simplicial Complexes." In J. Yust, J. Wild, and J. A. Burgoyne, eds. *Mathematics and Computation in Music*. Berlin: Springer, pp. 38–51.

Bigo, L., et al. 2014. "Spatial Transformations in Simplicial Chord Spaces." In *Proceedings of the Joint International Computer Music Conference and the Sound and Music Computing Conference*, pp. 308–313.

Chew, E. 2002. "The Spiral Array: An Algorithm for Determining Key Boundaries." In C. Anagnostopoulou, M. Ferrand, and A. Smaill, eds. *Music and Artificial Intelligence*. Berlin: Springer, pp. 18–31.

Chouvel, J.-M. 2009. "Traversée du vent et de la lumière: Six remarques pour une phénoménologie de la création musicale." Available online at jeanmarc.chouvel.3.free.fr/textes/Traversee0.0.pdf. Accessed February 2015.

Cohn, R. 2012. *Audacious Euphony: Chromatic Harmony and the Triad's Second Nature*. Oxford: Oxford University Press.

Euler, L. 1739. *Tentamen novae theoriae musicae: ex certissismis harmoniae principiis dilucide expositae*. Russia: Saint Petersburg Academy. Available online at eulerarchive.maa.org/pages/E033.html. Accessed February 2015.

Forte, A. 1973. *The Structure of Atonal Music*. New Haven, Connecticut: Yale University Press.

Giavitto, J.-L., and O. Michel. 2001. "MGS: A Rule-Based Programming Language for Complex Objects and Collections." *Electronic Notes in Theoretical Computer Science* 59(4):286–304.

Gollin, E. 1998. "Some Aspects of Three-Dimensional Tonnetze." *Journal of Music Theory* 42(2):195–206.

Lewin, D. 2007. *Generalized Musical Intervals and Transformations*. Oxford: Oxford University Press.

Mandereau, J., et al. 2011. "Z-Relation and Homometry in Musical Distributions." *Journal of Mathematics and Music* 5(2):83–98.

Rameau, J. P. 1750. *Démonstration du principe de l'harmonie: servant de base à tout l'art musical thèorique et pratique*. Paris: Durand.

Tymoczko, D. 2011. *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. Oxford: Oxford University Press.