Spatial Transformations in Simplicial Chord Spaces

Louis Bigo[1] **Daniele Ghisi**[2] **Antoine Spicher**[3] **Moreno Andreatta**[4]
[1] Department of Computer Science and Artificial Intelligence
University of the Basque Country UPV/EHU
[2] IRCAM
[3] Université Paris-Est LACL
[4] UPMC IRCAM - CNRS
louis.bigo@ehu.es
danieleghisi@gmail.com
spicher@lacl.fr
andreatta@ircam.fr

ABSTRACT

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

1. INTRODUCTION

Music theorists often represent sets of symbolic objects (notes, chords, rhythms, etc.) by spatial structures. The specification of a number of these structures can be facilitated by an algebraic reformulation of the represented objects. Studying combinatorial, geometrical or topological properties of these spaces inspires new approaches in musical theory. Moreover, these spaces can be exploited as "support spaces" to represent and analyze existing musical sequences. For example, one can observe *neo-Riemannian transformations* in the *Tonnetz* [1], voice-leading progressions in *orbifolds* [2], or track key boundaries in the *spiral array* [3].

We propose to use a set of chord spaces, inspired by the *Tonnetz*, to operate some musical transformations. These spaces are chord-based simplicial complexes which have proved to be useful in musical analysis [4]. In this work, we show their benefit in the context of musical transformations. Sections 2 and 3 provide technical and musical backgrounds. Section 4 presents what is a chord-based complex and how a musical sequence is represented as a trajectory within it. In section 5, we investigate some spatial transformations of trajectories and their musical interpretation. Section 6 presents two implementations of the concepts presented in the previous sections. The first one, HexaChord is an experimental software dedicated to computational music analysis. The second one is the Max object *bach.tonnetz*.

2. TECHNICAL BACKGROUND

2.1 Simplicial complexes

Let V be a set of elements. A *simplicial complex* $\mathcal{K}$ defined on V is a set of non-empty finite subsets of V, called *simplices* and denoted $\sigma \in \mathcal{K}$, verifying the *closure condition*:

For any simplex $\sigma \in \mathcal{K}$, every non-empty subset $\sigma' \subset \sigma$ is also an element of $\mathcal{K}$, i.e., $\sigma' \in \mathcal{K}$

We say that σ' is *incident* to σ, written $\sigma' \prec \sigma$. Every simplex σ of $\mathcal{K}$ is characterized by its *dimension* such that $dim(\sigma) = card(\sigma) - 1$ where the function $card$ gives the cardinality of σ. A simplex of dimension n is called a n-simplex. 0-Simplices can be represented by vertices, 1-simplices by edges, 2-simplices by triangles, etc.

The closure condition implies that every n-simplex is incident to $n+1$ $(n-1)$-simplices (*e.g.*, an edge is incident to 2 vertices, a triangle is incident to 3 edges, etc). A proper subset of a simplicial complex $\mathcal{K}$ which is also a simplicial complex is called a *sub-complex* of $\mathcal{K}$. For the sake of simplicity, we will often consider that the term "simplex" designates the sub-complex containing a simplex and all its incident simplices of lower dimensions. Figure 1 illustrates examples of n-simplices for $n \in \{0, 1, 2\}$.

A *simplicial d-complex* is a simplicial complex where the highest dimension of any simplex is d. A graph is a simplicial 1-complex. Figure 4 shows a simplicial 2-complex and a simplicial 3-complex. For any natural integer n, the *n-skeleton* of a simplicial complex $\mathcal{K}$ is defined by the sub-complex $\mathcal{S}_n(\mathcal{K})$ of this complex formed by its simplices of dimension n or less.

2.2 Simplicial collections

A *simplicial collection* $\mathcal{K}$ is a labeled simplicial complex. The term "collection" comes from the notion of *topological collection* used in the MGS programming language [5] which has strongly inspired this work.

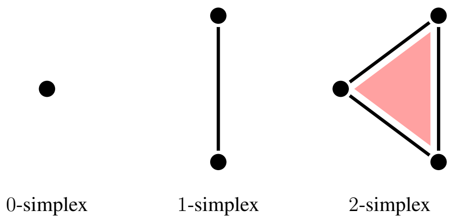

Figure 1. Some simplices.

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

2.3 Structural inclusions

In this work, we will be interested in the ways a complex can be embedded into an other one (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 $\phi : \mathcal{K} \to \mathcal{K}'$ is a *morphism of simplicial complexes* if for every cell σ and σ' of $\mathcal{K}$:

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

These two conditions preserve respectively the neighborhood between simplices and their dimension. In other words, a morphism of complex is a function which preserve its structure.

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

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

The *structural inclusion* enables to formulate how a 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 complex is *injective* if $\forall \sigma, \sigma' \in \mathcal{K}, \phi(\sigma) = \phi(\sigma') \Rightarrow \sigma = \sigma'$. Injectivity enables to distinguish in $\mathcal{K}'$ a sub-complex isomorphic to $\mathcal{K}$. We thus say that $\mathcal{K}$ is *structurally included* in $\mathcal{K}'$. Finally, every automorphism in a complex defines a structural inclusion into itself. The set of automorphisms of a complex represents its *structural symmetries*.

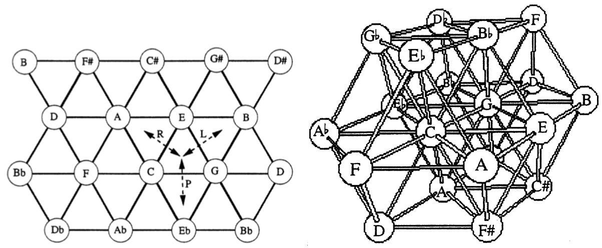

Figure 2. On the left the neo-Riemannian *Tonnetz* and the three neo-Riemannian operations P, L and R. On the right, a three-dimensional derivation of the *Tonnetz* by E. Gollin [8].

3. MUSICAL REPRESENTATIONS

In this section, we present two well-known notions in music theory: the *Tonnetz*, which is a spatial organization of pitches, and T/I classes, which provide a classification of musical chords. These notions constitute the musical starting point of this work.

3.1 The *Tonnetz*

One of the stronger motivations of this work is the wish to formalize a widely used tool in music theory, analysis and composition, named *Tonnetz*. The *Tonnetz* is a symbolic organization of pitches in the euclidean space following infinite axes associated to particular musical intervals. It was first investigated by L. Euler [6] for acoustical purpose [1] and rediscovered later by musicologists A. von Oettingen and H. Riemann. More recently, music theorists have shown a strong interest in this model, in particular to represent typical post-romantic chord progressions [1] currently called *neo-Riemannian transformations*. This model has been used in musical composition as well [7].

The *neo-RiemannianTonnetz* (on the left side of figure 2) is a graph in which pitches are organized along the intervals of fifth (horizontal axis), major and minor thirds (diagonal axis). This representation has the interesting property to reveal major and minor triads as triangles. The three arrows illustrate the *neo-Riemannian*operations P (Parallel), L (Leading-tone) and R (Relative) which enable transitions between two triads having two common notes. Many inspired theorists have investigated different derivations of the *Tonnetz*, often referred as to *generalized Tonnetze*. For instance, the figure 2 on the right illustrates a three-dimensional *Tonnetz* presented in [8]. This model corresponds to the one on the left figure 2, in which some new interval axes have been added. Tetrahedrons represent dominant seventh and half-diminished chords. Three-dimensional models are well adapted to study 4-note chords progressions. Finally, similar structures can be built by associating axes with intervals that are diatonic instead of chromatic. In these *diatonic Tonnetze*, vertices and shapes only represent notes and chords belonging to a unique tonality.

[1] Euler's *Tonnetz* (*Speculum Musicum*) organizes pitches in just intonation along pure fifths (horizontal axis) ans pure major thirds (vertical axis).

In this work, we are limited to the context of equal temperament and octave reduction *i.e.*, we are dealing with *pitch classes*, without consideration of octaves. For example, the notes $C\sharp 3$, $C\sharp 4$ and $D\flat 4$ are all considered under the same *pitch class*. In particular, what we call *Tonnetz* more exactly refers to the *pitch class Tonnetz*. In this context, the graph on the left of figure 2 repeats infinitely the 12 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). The Max object *bach.tonnetz* presented in section 6 enables, for example, to avoid octave identification within the *Tonnetz*.

3.2 *Generalized Tonnetze* and T/I classes

The highlighting of particular chords (minor/major chords as triangles in the neo-Riemannian *Tonnetz* and dominant seventh/half diminished chords as tetrahedrons in the three-dimensional *Tonnetz*) suggests the idea that the starting point of the construction of a *Tonnetz* could be a set of chords rather than a set of interval axes. In the two examples above, the represented chords in a *Tonnetz* are all equivalent up to transposition and inversion *i.e.*, they belong to the same T/I class. This property comes from the repetition and the invertibility of the intervals on the axes.

It is usual to identify a T/I class by the intervallic structure which is shared by all the chords of the class. For instance, major and minor chords all share the intervalic structure $[3, 4, 5]$ because the row of intervals between pitch classes they are resulting from is composed of a minor third (3 semitones), a major third (4 semitones) and a fourth (5 semitones). 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. This notation of the intervallic structure is defined up to reflection and circular permutation. Indeed, intervals are not ordered in the same direction for major ($[4, 3, 5]$) and minor chords ($[3, 4, 5]$). Dominant seventh and half diminished chords are identified by the intervallic structure $[2, 3, 3, 4]$.

T/I classes can be associated with the orbits of the action of the dihedral group $\mathbb{D}_N$ on the subsets of $\mathbb{Z}_N$ [9]. There exists 224 such classes in the chromatic system ($N = 12$), also known as *Forte classes* [10]. In the diatonic system ($N = 7$), which divides the octave in seven (non equal) parts, there exist 18 such classes. Following this line, we are interested in building the *generalized Tonnetze* associated with the 224 T/I chromatic classes and 18 T/I diatonic classes.

4. CHORD COMPLEXES AND TRAJECTORIES

In this section, we propose to represent 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.

4.1 Chord-based complexes

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

4.1.1 Generalized Tonnetze *as simplicial collections*

We use a method presented in [11] to represent chords as simplices. A n-note chord is represented by a $(n-1)$-simplex. A 0-simplex represents a single pitch class, a 1-simplex represents a 2-note chord, a 2-simplex represents a 3-note chord. Figure 3 illustrates on the right a simplicial collection representing the C major chord. It includes 7 simplices representing each sub-chord of C major (including pitch classes).

We represent a *generalized Tonnetz* as a simplicial collection composed by n-simplices representing the chords of a given T/I class. In the following, we note $\mathcal{K}[a_1, \ldots, a_i]$ the complex associated with the T/I class identified by the intervalic structure $[a_1, \ldots, a_i]$. Figure 4 illustrates regions of the complexes $\mathcal{K}[3, 4, 5]$ and $\mathcal{K}[2, 3, 3, 4]$. They respectively correspond to the two graphs of figure 2 in which 2-simplices and 3-simplices have been integrated. In other words, the neo-Riemannian *Tonnetz* and the three-dimensional *Tonnetz* respectively correspond to the 1-skeletons of $\mathcal{K}[3, 4, 5]$ and $\mathcal{K}[2, 3, 3, 4]$.

4.1.2 Chord complex construction

We build the chord complex $\mathcal{K}[a_1, \ldots, a_i]$ as follows. First, a n-note chord belonging to the class identified by the intervallic structure $[a_1, \ldots, a_i]$ is chosen and represented by a $(n-1)$-simplex. For example, the C major chord illustrated on the right figure 3 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, like in the *Tonnetz*. Then, simplices are naturally replicated along these axes, in a way that the represented chords respect the transpositions induced by the axes intervals. 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 unique tonality.

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

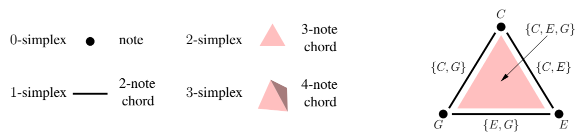

Figure 3. A chord represented as a simplex. The complex on the right corresponds to the C major chord and all 2-note chords and notes included within it.

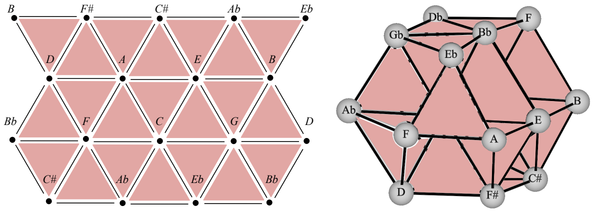

Figure 4. On the left, a region of the complex $\mathcal{K}[3,4,5]$ composed by major and minor chords. On the right, the complex $\mathcal{K}[2,3,3,4]$ composed by dominant seventh and half-diminished chords.

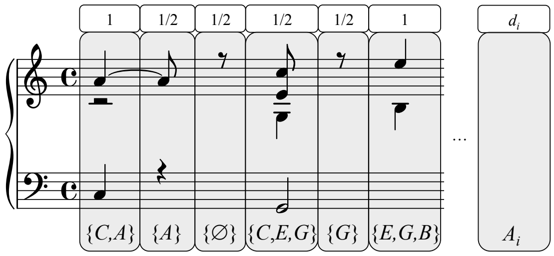

Figure 5. Segmentation of a sequence depending on the set of played pitch classes. The duration unit corresponds here to the quarter note.

$\mathcal{K}[3,4,5]$ and $\mathcal{K}[2,3,7]$ are both two-dimensional infinite triangular tessellation, as illustrated on the left of figure 4.

4.2 Representation of a musical sequence in a chord complex

In this section, we propose to represent a musical sequence by a *trajectory* in a chord complex. A *trajectory* is a sequence of regions of the complex. Each successive region represents a temporal segment of the piece. In this work, we use a very simple segmentation method, based on 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 figure 5. Each segment is characterized by its relative duration, compared to the other segments. Thus, we reduce a musical sequence P to a sequence of pitch class sets, each labeled by a relative duration. We have thus $P = [(A_0, d_0), \ldots, (A_N, d_N)]$ where A_i is the set of *active* pitch classes during a duration d_i.

As previously mentioned, each pitch class is represented in multiple locations in the *pitch class Tonnetz*. In the same way, a n-note chord is represented by multiple $(n-1)$-simplices in a chord complex.

A *trajectory* in a chord complex $\mathcal{K}$ is a sequence of sub-collections 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 $T_{\mathcal{K}}$ *represents* P if for every i, $d_i = d'_i$ and $\mathcal{K}_i$ is a simplicial sub-collection such that:

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

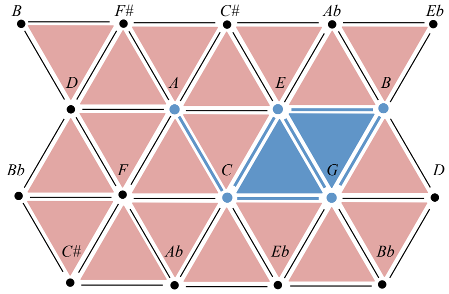

Figure 6. Trace (in blue) of a trajectory representing the sequence illustrated figure 5 in $\mathcal{K}[3,4,5]$.

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

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

Figure 6 illustrates the trace of a sequence representing the sequence illustrated figure 5 in $\mathcal{K}[3,4,5]$. The definition enables a large number of different trajectories to represent a given sequence P in a complex $\mathcal{K}$ (remember that each chord is represented in multiple locations in the complex). To automatically attribute a trajectory to a sequence, we use an algorithm based on two main criteria:

- chords must be represented as compact sub-complexes,
- chord transitions must correspond with small movements.

For a complete description of the algorithm, please refer to [11].

5. TRANSFORMATIONS OF TRAJECTORIES

In this section, we present different transformations of musical sequences, defined by spatial operations on trajectories. These operations can be rotations, translations, or embeddings of the trajectory in a new support space. Some spatial transformations correspond to well-known musical operations, for example transpositions and inversions. Some others do not have at the present any familiar interpretation.

5.1 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 $T_{\mathcal{K}}$ which represents the sequence P in $\mathcal{K}$. Let ϕ be a structural inclusion of $|\mathcal{T}|$ in $|\mathcal{K}'|$. The morphism ϕ enables a relabelling of the simplices of $\mathcal{K}$ to shape a trace $\mathcal{T}^{\phi}$ in $\mathcal{K}'$. This modification of labels induces then a transformation $\mathcal{T}_{\phi}$ of the sequence P into a different sequence P' defined by:

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

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

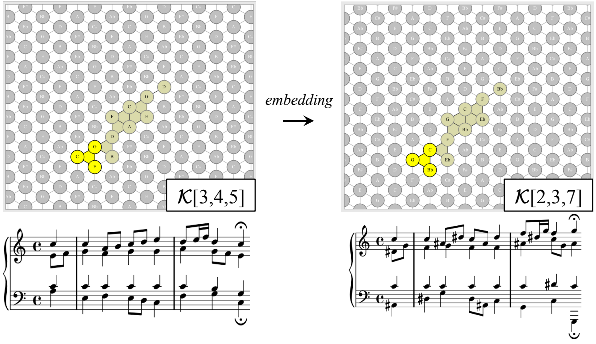

Figure 7. On the left, the first measures of the choral BWV256 of J.-S. Bach represented by a trajectory in $\mathcal{K}[3,4,5]$. On the right, the transformation of the sequence resulting from the embedding of the trajectory in $\mathcal{K}[2,3,7]$

The notation $\mathcal{T}_\phi$ stresses the fact that the transformation only depends 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 propose to choose the octave of the transformed pitch class in a way that the distance with the original pitch is minimized. Then, a transformation affects pitch classes without modifying too much the pitch register in which the new sequence is evolving. Furthermore, as this work concentrates on pitch transformations, duration of segments are left unchanged.

5.2 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 intuitively be understood as an embedding in a complex of a trajectory coming from an other complex. In particular, every isomorphism between two complexes $\mathcal{K}_1$ and $\mathcal{K}_2$ enables the embedding of a given trajectory built in one of the complexes into the second one.

As mentioned in section 4, chord complexes $\mathcal{K}[a_1,\dots,a_i]$ are isomorphic when they are of the same dimension. For example, $\mathcal{K}[3,4,5]$ and $\mathcal{K}[2,3,7]$ are isomorphic since they both result from 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 a second one.

Figure 7 illustrates this transformation with the first measures of the choral BWV256 of J.-S. Bach. The trajectory on the left represents the sequence in $\mathcal{K}[3,4,5]$. In this complex, triangles represent major and minor chords. On the right, the same trajectory in $\mathcal{K}[2,3,7]$, in which triangles represent "incomplete minor seventh chords", *i.e.*, minor or dominant seventh chords without fifth. These chords have the interesting property to include typical intervals of the pentatonic scale, which gives a particular color to the transformed sequence. [2]

Embedding in a chord complex a trajectory built in an other chord complex enables to give to a musical sequence a new harmonic color, with conservation of its characteristic shape. In particular, embedding a trajectory in a diatonic complex has the obvious consequence to give to the transformed sequence the tonality characterizing the complex.

5.3 Automorphism in a support space

When the chord complex $\mathcal{K}$ includes structural symmetries, the associated automorphisms define isometries which 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 division of the octave). Indeed, the construction method described section 4 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 intuitively be associated with the possible "simplex to simplex" superpositions of two copies of the complex, after some 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 a discrete translation or rotation of the trajectory. Some musical transformations produced by automorphisms in chord complexes are available in the previous online page.

5.3.1 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 in σ_2 is characterized by the interval class i that transforms the pitch classes associated with this vertices:$i = \mathcal{K}(\sigma_2) - \mathcal{K}(\sigma_1)$. 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 $n+i$. We thus have for any sequence P:

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

The application of a translation on a trajectory in a complex associated with a T/I class corresponds to a transposition. If $N = 12$, it is a chromatic transposition. If $N = 7$, it is a diatonic (or modal) transposition. If the sequence belongs to a tonality, a translation in the associated diatonic complex will reach to a change of mode *e.g.*, a "major to minor" transformation.

5.3.2 Discrete point reflections

A discrete point reflection in a complex has for consequence to transform intervals in their opposite. 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 distance separating m to a pitch class is inverted to produce the new pitch class.

[2] The result of the transformation is available online in Midi format on the page `http://www.lacl.fr/~lbigo/icmc-smc14`.

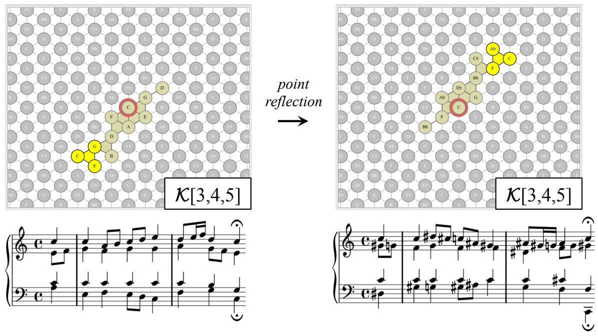

Figure 8. On the left, a trajectory representing the first measures of the choral BWV256 of J.-S. Bach in $\mathcal{K}[3, 4, 5]$. On the right, a point reflection is applied on the trajectory and produces a new sequence.

As for the translations, a pitch class is transformed according to its value and not to the position of its simplex in the complex. We thus have for a given sequence P:

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

Figure 8 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.

5.3.3 Other transformations

As mentioned, the two precedent transformations have the property to produce pitch classes entirely determined by their original values, and 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 specified as simply on pitch classes. For example, line symmetries or rotations can transform vertices labeled by a same pitch class into vertices labeled by different pitch classes. The same generally applies in the case of the embedding of a trajectory in a new chord complex. These transformations don't have any musical interpretation to our knowledge and result in new musical operations.

6. SOFTWARES

In this section, we present two softwares enabling to work with the notions presented on the previous sections: HexaChord which is an application dedicated to music analysis, and the Max object *bach.tonnetz* which is dedicated to composition.

6.1 HexaChord

HexaChord [3] is a computer-aided music analysis environment based on the spatial representations previously presented. The software provides a visualization of any chord

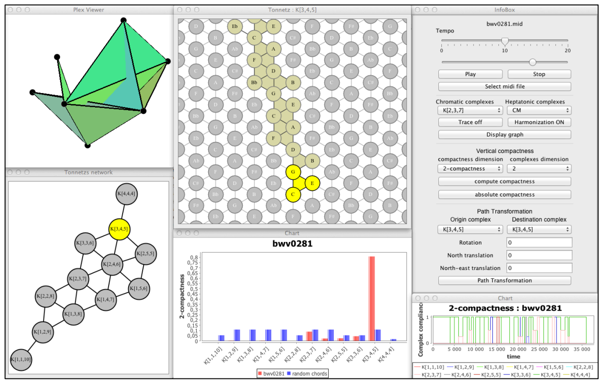

Figure 9. Graphical user interface of HexaChord.

complex related to a T/I class grouping 3-note chords in diatonic and chromatic scales. These complexes are infinite two-dimensional triangular tessellations.

Musical pieces are imported as MIDI files. A trajectory is automatically computed for any pair of musical piece/chord complex. The trajectory is represented as a path which evolves in real time in its complex during the play of the piece. Transformations presented section 5 can all be applied on a trajectory. The transformed musical sequence can be exported as a MIDI file. The results on the online page have been generated with HexaChord.

Other features dedicated to analysis have been integrated in the application. For instance, HexaChord determines automatically the chord complex which is the more adapted to represent a musical sequence. This task relies to the notion of *compliance* [4] and is achieved by comparing the compactness of the trajectories representing the piece in the different complexes.

6.2 The *bach.tonnetz* object

If one needs to deal with Tonnetz representations interactively, a very natural solution is to handle them in a real-time environment. An easy way to do it is to take advantage of the *bach* [4] library, a set of externals and patches for Max, bringing computer-assisted composition into the real-time world [12, 13]. Among its features, *bach* has a subset of tools dedicated to musical representations, including the *bach.tonnetz* object, which implements and displays a Tonnetz centered in 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 one of the following formats: cents, note names, pitch-classes, diatonic intervals, coordinates in the lattice space. Elementary transformations such as translations and rotations presented in section 5 can be easily performed both via the interface and via messages.

bach.tonnetz can easily echo the incoming data to its outlets, in order 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

[3] http://vimeo.com/38102171

[4] http://www.bachproject.net

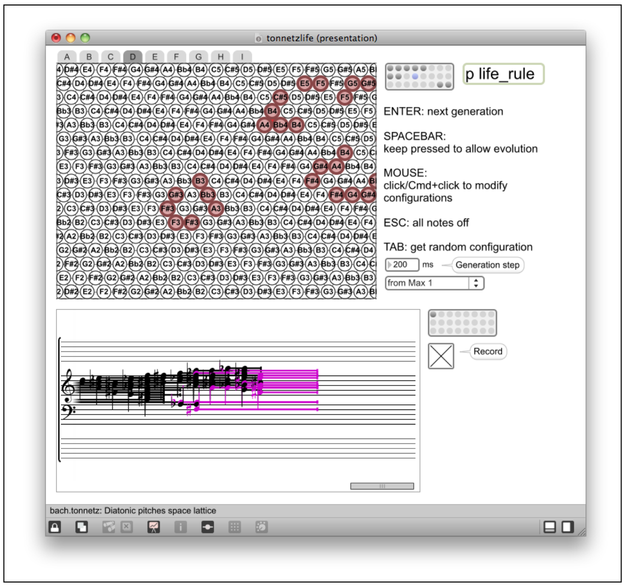

Figure 10. A patch used by [name omitted for review] in order to explore musical material as two-dimensional automata on a Tonnetz. Rules are defined by patching (in subpatch `p life_rule`), and a few basic interface commands are given to control the breeding. The output result is recorded in a *bach.roll*, for possible further usage. User can modify at any moment the Tonnetz structure (some presets are marked with letters in the upper part of the patch), as well as its content.

and variously rotate or shift it. To allow a more faithful representations of MIDI data, each selected node in the lattice can be given a *velocity* value, which in turn can be visualized graphically either by varying node colors or by adjusting node sizes. Moreover, *bach.tonnetz* also supports microtones and just intonation.

Since *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 possible adaptations for hexagonal grids, see Fig. 10); this is made even easier by the brand new cage [5] library (a set of high-level abstractions based on *bach*), which contains the module *cage.life* producing two-dimensional cellular automata [14].

7. CONCLUSION

In this paper we have presented in a formal way a general framework to apply transformations on musical sequences based on their spatial representations. These representations 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 well-known musical operations, most of them are still waiting for pertinent musical interpretation. This is typically the case of the embedding of a given trajectory into a new chord complex.

[5] `http://www.bachproject.net/cage`

Acknowledgments

The authors are very grateful to J.-L. Giavitto from the RepMus team at IRCAM, O. Michel at LACL - Université Paris-Est, J.-M. Chouvel from Université de Reims and A. Agostini for fruitful discussions and software developments. This research is supported by LACL - Université Paris-Est, IRCAM and the project Lrn2Cre8 which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

8. REFERENCES

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

[2] D. Tymoczko, *A geometry of music: harmony and counterpoint in the extended common practice*. Oxford University Press, USA, 2011.

[3] E. Chew, "The spiral array: An algorithm for determining key boundaries," in *Music and artificial intelligence*. Springer, 2002, pp. 18–31.

[4] L. Bigo, M. Andreatta, J.-L. Giavitto, O. Michel, and A. Spicher, "Computation and visualization of musical structures in chord-based simplicial complexes," in *Mathematics and Computation in Music*. Springer, 2013, pp. 38–51.

[5] J.-L. Giavitto and O. Michel, "Mgs: a rule-based programming language for complex objects and collections," *Electronic Notes in Theoretical Computer Science*, vol. 59, no. 4, pp. 286–304, 2001.

[6] L. Euler, *Tentamen novae theoriae musicae ex certissismis harmoniae principiis dilucide expositae*. Saint Petersburg Academy, 1739.

[7] J.-M. Chouvel. (2009) Traversée du vent et de la lumière. six remarques pour une phénoménologie de la création musicale. [Online]. Available: http://jeanmarc.chouvel.3.free.fr/textes/Traversee0.0.pdf

[8] E. Gollin, "Some aspects of three-dimensional tonnetze"," *Journal of Music Theory*, pp. 195–206, 1998.

[9] M. Andreatta and C. Agon, "Implementing algebraic methods in openmusic." in *Proceedings of the International Computer Music Conference, Singaphore*, 2003.

[10] A. Forte, *The structure of atonal music*. Yale University Press, 1973.

[11] L. Bigo, "Représentations symboliques musicales et calcul spatial," Ph.D. dissertation, LACL/IRCAM, Université Paris-Est, 2013.

[12] A. Agostini and D. Ghisi, "*bach*: an environment for computer-aided composition in Max," in *Proceedings of the International Computer Music Conference (ICMC 2012)*, Ljubljana, Slovenia, 2012, pp. 373–378.

[13] ——, "Real-time computer-aided composition with *bach*," *Contemporary Music Review*, no. 32 (1), pp. 41–48, 2013.

[14] A. Agostini, E. Daubresse, and D. Ghisi, "cage: une librairie de haut niveau dédiée à la composition assistée par ordinateur dans Max," in *to appear*, 2014.