# Random Walks on Neo-Riemannian Spaces: Towards Generative Transformations

**Philon Nguyen**
Concordia University
1550 De Maisonneuve Blvd. W.
Montreal, QC, CA H3H IG8
philon.nguyen@gmail.com

**Eldad Tsabary**
Concordia University
1550 De Maisonneuve Blvd. W.
Montreal, QC, CA H3H IG8
eldad.tsabary@concordia.ca

## Abstract

Random walks, fractional Brownian motion and stochastic processes have been used extensively by composers such as Iannis Xenakis and others, creating instantly recognizable textures. A trained ear can differentiate a uniform random walk from a Poisson process or an fBm process and random rotations. In the opera *Sophocles: Antigone* by one of the authors of this paper, random walks on neo-Riemannian PLR spaces were experimented with yielding mixed impressions of process music and post-romantic chromaticism. When the random walk is steered by transformational rules, special textures and harmonies emerge. We propose a new kind of parameterizable random walks, a generative system, on a space of arbitrary length chords equipped with an arbitrary distance measure steered from a customizable corpus learned by the system. The corpus provides a particular texture and harmony to the generative process. The learned neo-Riemannian spaces equipped with some distance measure provide the transformational rule base of the concatenative synthesis process.

## 1 Introductiion

In a now famous 2005 paper, *Spectra and Sprites*, composer Tristan Murail marvels at how difficult it has become for composers to choose notes [Murail 2005]. By the time he made the statement, choosing notes had evolved from highly educated choices or mathematically oriented processes to total random choices. It is well-known that classical music mostly chooses from the diatonic scale, post-romantic and Hollywoodian music from the chromatic scale in a somewhat neo-Riemannian fashion, atonal and serial music is chromatic in the sense of a 12-tone scale but by now changes to the transformation rules had occurred such as the introduction of reversals, inversions, non-repetition… By the time serial music had stormed the cultural elites in America and Europe, composers such as John Cage were experimenting with chance music and indeterminacy, while others, such as Iannis Xenakis were building ways to generate arbitrary scales; other models such as Peter Schat's tone clock were meant to encompass tonal and atonal music alike (e.g. Cage's chance music [Pritchett 1996], Xenakis' sieves [Gibson 2010], Peter Schat's tone clock [Schat 2003]). The particular sound and textures that Xenakis creates are often based on some random walk on a carefully chosen sieve. Continuing on scale building, spectralism would then return to harmony its birthright by equating in a now famous stance harmony and timbre. The goal was to make music with sounds and not abstract mathematics unrelated to sound [Murail 2004]. One of the techniques of the so-called spectral school that is now used by a generation of composers was to generate a scale from the partials of a sound file [Murail 2004]. More recently, the emergence of machine learning and AI used in for example generative systems has spawned many techniques based on corpus learning: the choice of a note is now a parameter depending on some corpus in some feature space. Also, echoing Murail's statement, we can now say that the history of recent music seems to have been at least partially conditioned on that difficult choice of notes any composer must do, whether educated or random.

In this paper, we are interested in generative methods that are harmonically aware. Our generative process is some parameterizable random walk (where similarity or dissimilarity can be used to steer

3rd Conference on AI Music Creativity (AIMC 2022)

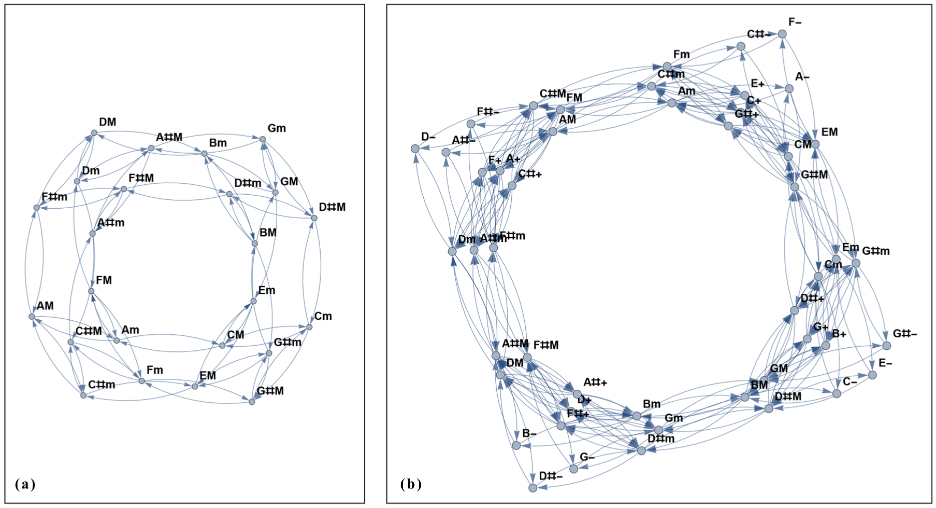



Figure 1: *(a) Graph linking all major and minor triads such that each transition is a neo-Riemannian transformation; (b) graph linking all major, minor, augmented and diminished triads such that each transition is a semitone transformation.*

the walk). The technique we propose can be used off-line or in real-time performances. The machine learning and feature discovery yields the generative aspect while a chosen distance space equipped with some metric yields the harmony in the sense of a neo-Riemannian harmony.

### 1.1 Generative Systems and Neo-Riemannian Spaces: From Patterns and Processes to Transformations

[Lévy 2013] distinguishes purely algorithmic systems and composition systems from generative systems, which display a degree of learning and control. A system that learns can be trained on a corpus of works and extract a database of patterns whereas a system that displays control can function autonomously with minimal intervention. Generative systems have admittedly made use of heuristics such as cellular automata and artificial life concepts [Miranda 2001], swarm intelligence [Blackwell & Bentley 2002] or genetic algorithms [Biles 1994]. Generative systems have also exploited performance and improvision to effectively serve as real time accompaniment systems [Dannenberg 1993; Lewis 2000; Thom 2000; Young 2008; Patchet et al. 2013; Carsault 2017; Nika et al. 2017]. In the generative systems literature, the use of the Oracle Factor algorithm [Dubnov & Assayag 1998], an algorithm initially built for genetic sequencing, spawned a series of generative systems available in the Max environment and used as research systems (e.g. OMax [Assayag & Bloch 2007] , DYCI2 [Nika et al. 2017], Somax [Carsault 2017]), all based out of IRCAM. Such systems try to predict an improvisation using a set of most probable next pattern of some suffix tree which can be generated in real time [Assayag et al. 2006] . The underlying assumption here is that repeated patterns will yield more probable patterns. Many of these techniques have been regrouped under the vocable of *concatenative synthesis*.

Alternatively, the revival in the 1990's and 2000's of harmonic transformational theories such as those of Hugo Riemann and Carl Friedrich Weitzmann in the XIXth century was championed and popularized by scholars such as David Lewin [Lewin 1987], Fred Lerdahl [Lerdahl 1996], Richard Cohn [Cohn 2000, 2012] and Dmitri Tymoczko [Tymoczko 2011] among others and have generated many followers. A general pattern in these neo-Riemannian theories is the embedding of harmony in a space equipped with a distance measure. The traditional PLR Riemannian transformation space is the well-known Tonnetz (a bidirectional graph equipped with a shortest path metric). In an opera composed in 2021, *Sophocles: Antigone* [Nguyen 2021], one of the authors of this paper and also composer used a simple random walk in a Tonnetz; the result is a texture which conveyed a sense of marvel and fantasy (neo-Riemannian transformations are known to have a Hollywoodian and cinematic texture [Lehman 2018]). This particular piece generated our curiosity in such techniques. A colour is given to something as neutral as a random walk by constraining the space and the

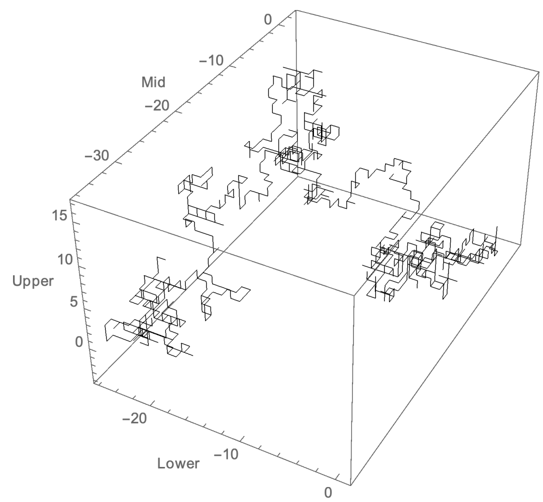



Figure 2: *A uniform random walk in triadic space where each step is a transformation of a semi-tone. The random walk periodically clusters in some regions of space before bifurcating.*

metric of the walk. Extensions of the Tonnetz (which accounts for major and minor triads) to, say, augmented and diminished triads are the so-called Weitzman transforms which can generate a question where Riemannian transforms convey an answer. In Section 2, we extend these so-called Weitzman transform to a generalized Weitzman transform, a closed graph in which nodes represent an arbitrary subset of chords equipped with a shortest path metric. Closure is important in the context of random walks if we want any chord/node in a graph to be reachable. Also, a transformation space can be represented as some space equipped with a distance measure (e.g. a graph equipped with the shortest path metric, or a distance space equipped with an $\ell_p$ metric). Section 3 discusses a set of chordal distance measures which can be found in the literature. Also, tonality is not a universal concept. Tonality was created as the sum of the multiple decisions taken by practitioners over the centuries in Western music, some rejected and some integrated in the doxa of musical aesthetics. It is best seen as a random walk or better still an evolving process in a space of cognitive affinities and preferences.

Section 4 discusses the combination of neo-Riemannian theories and generative systems and is an attempt at creating a system based on generative transformations. We propose such a system that performs parametrizable random walks (similarity and dissimilarity can be user parameters) effectively steered by some space equipped with some metric that can be learned from an existing corpus that is either represented in MIDI or audio. In contrast with other existing generative system, our system does no attempt to reproduce a style or a genre, or learn patterns from a corpus, it creates a harmonic distance space from a corpus in which a process can navigate in a parametrized fashion, allowing for morphing between two corpuses for example. The produced artefact does not necessarily preserve style or idiom, but rather harmony and timbre.

Finally, examples of applications of our method to existing musical works, Rachmaninov's *Second Concerto* (a piano reduction of the second movement) and Schoenberg's *Klavierstücke* Op.11 No.3, are discussed and illustrated.

## 2 Random Walks on Riemannian and Weitzman Graphs

It is natural to perform random walks over the Tonnetz. Given a large enough random walk radius (step size), any chord in the Tonnetz is reachable from any other chord. This property of the Tonnetz is termed closure. Weitzman transformations are not technically closed (semitone transformations) unless we enforce major, minor, augmented and diminished triads in the output of the transformation. This constraint could be that the ordered intervalic content be exactly $(300, 400)$, $(400, 300)$, $(400, 400)$ or $(300, 300)$ midicents. In this sense Weitzman transformations are closed. We can generalize Weitzman transformations (semitone) by enforcing an arbitrary check on the output. If the generalized transformation generates a connected graph (closure), the generated space is said to be

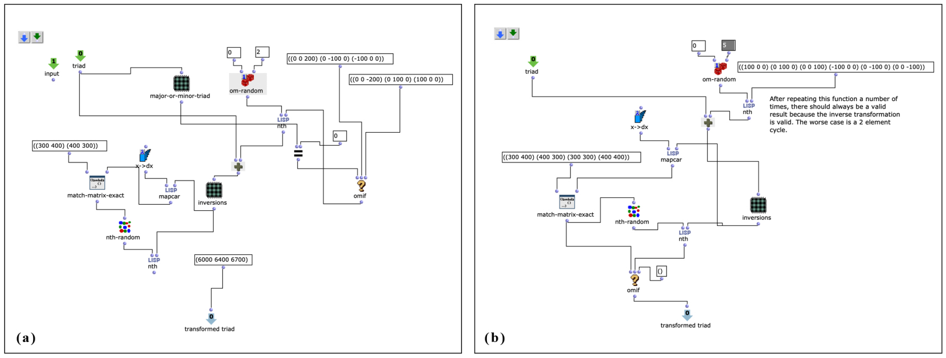



Figure 3: *(a) An OpenMusic patch for random Riemannian PLR transformations; (b) an OpenMusic patch for random Weitzman transformations (closure is ensured by the matrix-match-exact function which checks if the transformed triad is a major, minor, augmented or diminished triad).*

closed under the transformation. We could therefore generate say a whole tone triadic transformation space such as the root, the middle and top notes are respectively part of a whole tone scale (i.e. separated by an integer multiple of a whole tone). The condition would then be that any note of the chord be part of a given whole tone scale.

A guided walk is equivalent to a random walk because the walk is necessarily embedded in a space whose topology guides the choice function (distance minimization for example). For the sake of argument an ordered sequence is necessarily isomorphic to at least one random walk in some equivalent space. The given space yields a specific harmony. In this sense, a system such as Omax is, for the sake of argument, a parametrizable real time random walk on suffix tree feature spaces equipped with some randomized shortest path distance that was used in real concert settings with significant human computer interaction

Figure 1 shows a graph of Riemannian (major and minor triads) and Weitzman spaces (major, minor, augmented and diminished triads). A random walk of radius 1 is then a random walk where each step crosses a single edge. A random walk with a radius of 2 crosses 2 edges and so on. The smaller the radius, the more consonant are the chosen chords. The more distant, the more likely dissonance is introduced, until the random walk wraps around and becomes equivalent to a random walk of radius 1. Figure 2 shows a semitone random walk in some triadic space where each note is a dimension of the graph. Figure 3 displays OpenMusic patches for random walks on the Tonnetz and Weitzman spaces. The Weitzman space patch can be generalized by replacing the constraint "matrix-match-exact".

## 3 Distance Measures on Chordal Spaces

Neo-Riemanian theories are based on the embedding of a set of possibly infinite chords into a distance space, which can be Riemann's original triangular lattice of major and minor triads, Weitzman lattices, Lerdahl or Chew distance spaces, Timoczko manifold of all possible 4 note chords or some arbitrary Euclidean space or weighted graph for instance. Also, [Tymoczko 2009] categorizes chordal distance measures in 3 categories; the first, based on voice leadings (e.g. orbifolds); the second, based on acoustics (e.g. Tonnetz); the third, based on intervalic content (e.g, Estrada distance). In a survey of chordal distance measures, [Rocher et al. 2010] lists the following distance measures on chordal content: the Costère distance (based on affinity potentials which favour minor seconds, perfect fourths, perfect fifths and major sevents), intervalic content and the Estrada distance (both based on intervals of the product space of all notes in a chord), the Chew distance or Spiral Array distance (based on a geometric representation of the circle of fifths), Lerdahl's tonal pitch space, the Paiement distance (which is in frequency space and takes into account the harmonics of the components of a chord) and the pitch interval distance.

In Figure 4, we calculate the distance matrices of all possible triads using different distance measures (i.e. chromagrams, Estrada distance, Costère distance and Chew distance) and then use Multidimensional Scaling to reconstruct the embedding space of these triads. We can easily notice that intervalic distances (Estrada distance) do not distinguish as much as other distances (i.e. many chords are

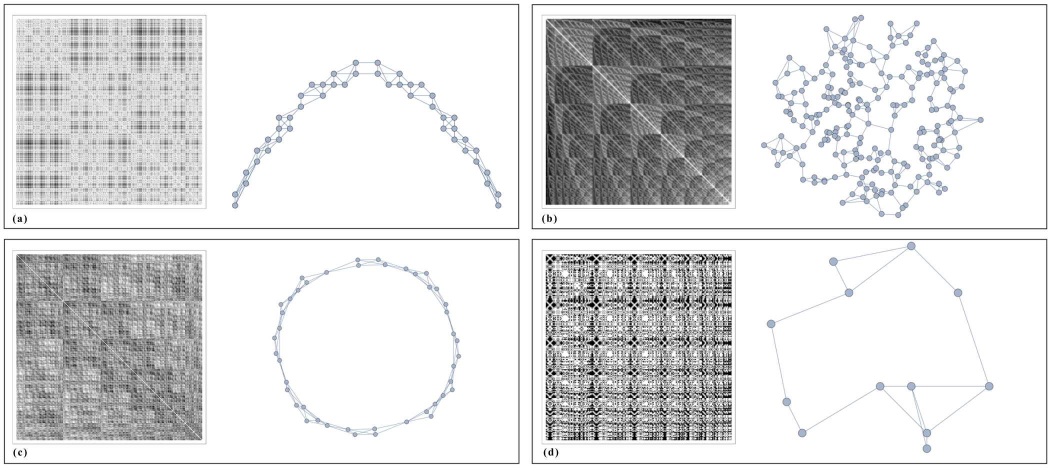



Figure 4: *On the left, the distance matrices (i.e. similarity matrices); on the right, the Multidimensional Scaling (MDS) embeddings (using metric MDS) in 2-dimensional Euclidean space (from the embedding, a nearest neighbor graph is constructed) for the following distance measures: (a) Chew distance (Spiral Array distance); (b) Chroma distance; (c) Costère distance; (d) Estrada distance.*

Table 1: Nearest Neighbor Graph Connectivity Constant

| Distance Measure | Connectivity Constant |
|---|---|
| Chew | 4 |
| Chroma | 3 |
| Costère | 4 |
| Estrada | 2 |

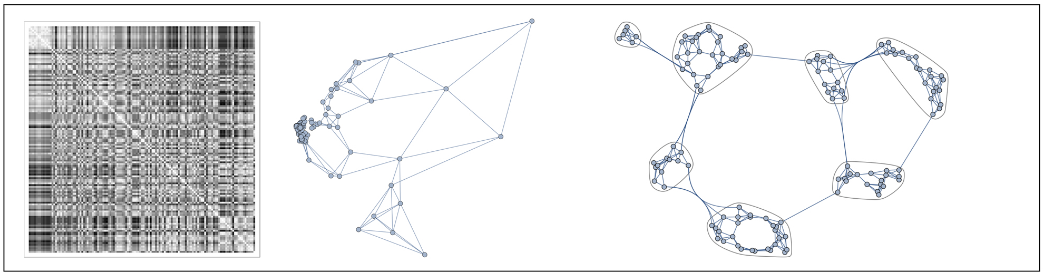
mapped to the same point in space as intervalic content is invariant under many transformations). Let $n$ be a constant such that any node of the reconstituted graph $\mathbf{G}$ is connected to $n$ nearest neighbors and $\mathbf{G}$ is connected. Table 1 lists the values of $n$ for each of the analyzed distance spaces. Also, the Costère distance and the Chew distance take into account musical properties such as the circle of fifths or the importance of perfect fourths and fifths in traditional music theory. It is interesting to notice that they yield geometric embeddings. The Estrada distance and the chromagrams yield point clouds with the Estrada distance being less distinguishing due to the invariance properties of intervalic vectors. The distance matrices, which are ordered in subset order, display fractal properties (fractional fractal dimensions), especially the chroma distance matrix.

## 4 Learning Spaces Equipped with Distance Measures

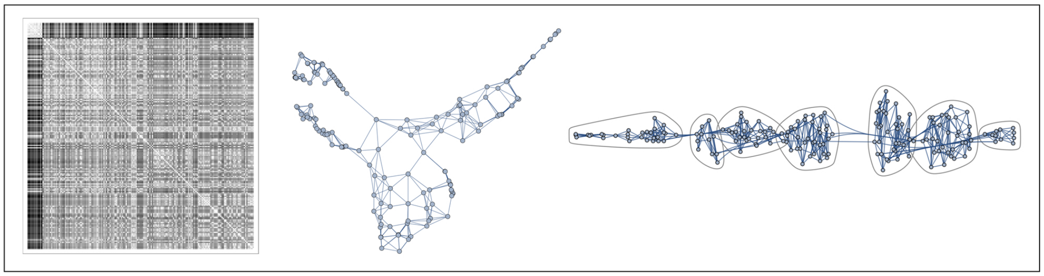
Algorithm 1 shows an iteration of the feature extraction process on some duration-chord tuple of a segmentation. This can be repeated as needed, as long as there are segments in the input, in real time or offline. We use a segmentation algorithm similar to the Somax MIDI segmentation algorithm implemented in the Max software [Carsault 2017]. What is generated, the matrix $\mathbf{M}$ is actually a similarity matrix. For any two chords, this matrix, which indexes all chords of some arbitrary lengths up to some constant $l$ in subset order, actually accumulates the distance between these two chords whenever they are encountered in the feature extraction process. To account for the distance in the time dimension (segment 1 may occur at time 1, segment 2 at time 2 and segment 3 at time 3), we introduce a decay function in the form of an exponential given some scaling factor $n$ and a duration $t$:

$$\delta(t) = \begin{cases} 1, & \text{if } 1 - e^{-t/n} + c \geq 1 \\ 1 - e^{-t/n} + c, & \text{if } 1 - e^{-t/n} + c < 1 \end{cases} \tag{1}$$

for some small constant $c$ to avoid 0 distances. This decay function is used to scale distances to $(0, 1)$ as well. It may be useful to set the diagonal of the transformation matrix obtained in Algorithm 1 to



**(a)**



**(b)**

Figure 5: *On the left, the transformation matrices (i.e. similarity distance matrices); in the center, the Multidimensional Scaling (MDS) embeddings into a 2-dimensional Euclidean space; on the right, the network community graphs of the MDS embeddings (clustering) of the following pieces: (a) features generated from Schoenberg's Klavierstücke Op.11 No.3; (b) features generated from Rachmaninov's Second Concerto (Movement 2)*

---

**Algorithm 1** Feature extraction algorithm from a Somax-type (duration, midi-on notes) MIDI segmentation algorithm

---

**procedure** EXTRACT($durations, chords$)
    $window$ = some constant in ms ▷ initialize a window size in ms
    $n_1, c_1$ = some scaling constants for the decay function $\delta_1$ ▷ initialize to some constant
    $n_2, c_2$ = some scaling constants for the decay function $\delta_2$ ▷ initialize to some constant
    Read the ($durations[i], chords[i]$) tuple at index $i$
    $t$ = 0
    $j = i + 1$
    $\mathbf{M} = \mathbf{1}$
    **while** $t < window$ **do** ▷ iterate over the window length
        $i1$ = index of $chords[i]$ in distance matrix ▷ index in subset order
        $i2$ = index of $chords[j]$ in distance matrix
        $\mathbf{M}[i1, i2]$ = $\mathbf{M}[i1, i2]$ $\delta_1(t)$ $\delta_2(d(chords[i], chords[j]))$ ▷ using some distance measure
        $t = t + durations[j - 1]$
        $j = j + 1$
    Return $\mathbf{M}$

---

**Algorithm 2** Generation/synthesis algorithm given an input chord, a transformation matrix and a segmentation

---

**procedure** GENERATE($initchord, \mathbf{M}, noteson$)
    $r$ = constant (possibly a list) defining the radius of the random walk
    $o$ = constant (possibly a list) defining the order chosen in the nearest neighbor ordering
    $i$ = random index from $r$-nearest neighbors in all chords minimizing a distance to $initchord$
    $chordNumber$ = random index from the $r$-nearest neighbors of order $o$ on row $\mathbf{M}[i]$
    $seg$ = segment in $noteson$ matching the $chordNumber$ index
    Return $seg$

---

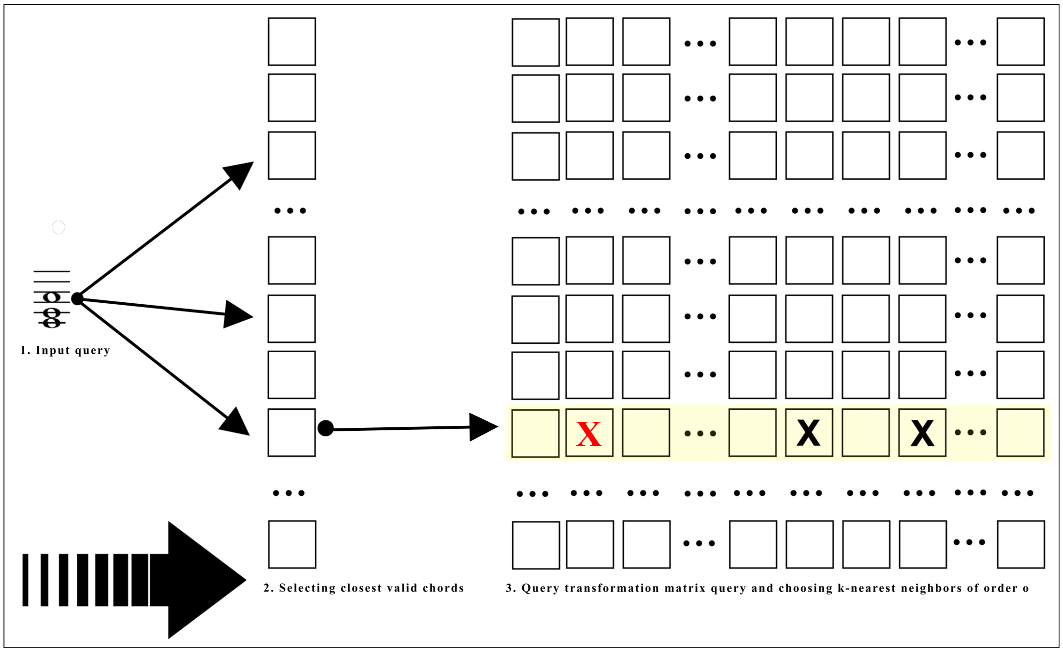



Figure 6: *Synthesis/generation algorithm: an input chord is used as a query (the input can be taken from machine listening or be some offline query chained in such a way that the next query is the previous query result). From the list of valid chords (chords that are present in the segmentation) nearest to the input based on some chosen distance measure, a random chord index is chosen (indices are ordered in subset order). The chosen chord index is used to query the transformation matrices. The values with lowest values (in the case of a nearest neighbor query) or the k-nearest neighbors at ordering index $i$ are collected and a random choice is then made from them.*

1 as well to avoid repetitions in the nearest neighbor search of Algorithm 2. Algorithm 2 shows an iteration of the generation algorithm given some initial input chord. This algorithm can be repeated as many times as required while the algorithm state (the segmentation and the transformation matrix) can be updated in real time. The input chord can be a chord coming from a machine listening system or some arbitrary initial input to trigger a random walk in an off-line generative process.

Figure 6 shows the synthesis process. A query can return a random choice among, say, the 3 nearest neighbors of the query, or some range, say, a random choice between the 12-14 nearest neighbors of the query (in which case, the radius of the random walk could be 3 and the order 4) or the 21-23 nearest neighbors (in which case, the radius of the random walk could be 3 and the order 7).

## 5 Learned Distance Spaces: *Rach2* and Schoenberg's *Klavierstücke* Op.11 No.3

We tested our generative process based on our concatenative synthesis algorithm steered by a distance space on some metric learned on some arbitrary corpus on 2 works from the repertoire: Schoeberg's *Klavierstücke* Op.11 No.3 and a piano reduction of Rachmaninov's *Second Concerto* (Movement 2). The distance matrices of both pieces show high similarity (white tones) in the upper left corner of the matrices: these indexes are occupied by chords of short lengths (e.g. 1, 2). Transitions from short lengths to longer lengths are rare (darker tones). Transitions from longer lengths to other longer lengths chords also occur in some visible patterns. The nearest neighbor graph connectivity constant of the Rachmaninov piece was 6 whereas the Schoenberg had a constant of 4. This means that for the Rachmaninov embedding to generate a complete graph, 6 nearest neighbors were necessary. That number is 4 for the Schoenberg piece. The clustering of the Rachmaninov piece was seemingly linear. This may hint at the fact that the Rachmaninov piece had a more traditional structure going from one theme to the other, whereas the Schoenberg piece was more deconstructed and its harmony seemed to cluster into 7 groups.


Figure 7: *(a) A synthesis based on Schoenberg's Klavierstücke Op.11 No.3; (b) a synthesis based on a piano reduction of Rachmaninov's Second Concerto (Movement 2).*

Figure 7 displays examples of synthesis with initial chord set to C major with a window parameter of 10 seconds. This means that each segment window lasted 10 seconds in the feature extraction algorithm. The result may be surprising to some ears. Increasing the segment length (if the algorithm matches segment $k$ and the length is $l$, then the algorithm returns segments $k, k+1, \ldots, k+l-1$), may increase the stylistic and idiomatic recognizability (a parallel to granular synthesis can be made). Also, one can then easily imagine a morphing process between to transformation matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ based on some morphing parameter $0 < \mu < 1$. When $\mu = 0$, the algorithm uses $\mathbf{M}_1$ with probability 1; when $\mu = 0.5$, the algorithm uses $\mathbf{M}_1$ with probability 0.5 and $\mathbf{M}_1$ with probability 0.5; and when $\mu = 1$, the algorithm uses $\mathbf{M}_2$ with probability 1.

## 6 Conclusion

This work is an exploration of random walks in distance spaces equipped with some chordal metric. We have analyzed different distance measures (i.e. chromagrams, Estrada distance, Costère distance and Chew distance) and showed some realizations of their 2-dimensional embedding based on their associated distance matrices. We have proposed an algorithm to learn transformation matrices (based on distance matrices) from some corpus of data. Our algorithm has a feature extraction phase and a generation/synthesis phase, both of which can be executed in real time and concurrently. The generation/synthesis phase can be seen as a random walk where the radius and the order of the random walk can be parameterized. The higher the order, the more dissimilar the results of some query will be. The higher the radius, the larger the probability of some random jump in another loci will be. The system can also be used in machine listening mode. The use of multiple corpuses is supported by our system and morphing between different corpuses can be implemented.

Future explorations on neo-Riemannian transformational spaces and generative systems will yield new musical objects, new timbres and new textures.

## References

[Assayag et al. 2006] Assayag, G. et al. (2006) Omax Brothers : a Dynamic Topology of Agents for Improvization Learning. In *Workshop on Audio and Music Computing for Multimedia, ACM Multimedia 2006*. Santa Barbara.

[Assayag & Bloch 2007] Assayag, G. & Bloch, G. (2007) Navigating the Oracle: a Heuristic Approach. *Proc. ICMC'07, The In. Comp. Music Association*. Copenhagen.

[Biles 1994] Biles, J.A. (1994) GenJam: A Genetic Algorithm for Generating Jazz Solos. In *International Computer Music Conference*.

[Blackwell & Bentley 2002] Blackwell, T.M. & Bentley, P. (2002) Improvised music with swarms. In *Proceedings of the 2002 Congress on Evolutionary Computation*. **2**: 1462-1467.

[Carsault et al. 2021] Carsault, T. & Nika, J. & Esling, P. & Assayag, G. (2021) Combining Real-Time Extraction and Prediction of Musical Chord Progressions for Creative Applications. *Electronics 2021*, **10**(21), 2634.

[Carsault 2017] Carsault, T. (2017) Automatic Chord Extraction And Musical Structure Prediction Through Semi-Supervised Learning - Application to human-computer improvisation. Master ATIAM, UPMC.

[Cohn 2000] Cohn, R. (2000). Weitzmann's Regions, My Cycles, and Douthett's Dancing Cubes. *Music Theory Spectrum* **22**(2): 89-103.

[Cohn 2012] Cohn, R. (2012). *Audacious Euphony: Chromatic Harmony and the Triad's Second Nature*. 2nd Edition. New York: Oxford University Press.

[Collins 2012] Collins, N. (2012) Automatic composition of electroacoustic art music utilizing machine listening. *Computer Music Journal*. **36**(3): 8-23.

[Dannenberg 1993] Dannenberg, R.B. (1993) Software Support for Interactive Multimedia Performance. *Journal of New Music Research* **22**(3): 213-228.

[Dubnov & Assayag 1998] Dubnov, S. & Assayag, G. (1998) Universal Classification Applied to Musical Sequences. *ICMC: International Computer Music Conference*. Ann Arbor Michigan.

[Gibson 2010] Gibson, B. (2010) *The Instrumental Music of Iannis Xenakis: Theory, Practice, Self-Borrowing*. Pendragon Press.

[Lehman 2018] Lehman, F. (2018) *Hollywood Harmony: Musical Wonder and the Sound of Cinema*. Oxford University Press.

[Lerdahl 1996] Lerdhal, F. (1996) *A Generative Theory of Tonal Music*. The MIT Press.

[Lévy 2013] Lévy, B. (2013) *Principes et architectures pour un système interactif et agnostique dédié à l'improvisation musicale*. PhD Thesis at Université Pierre et Marie Curie, Paris.

[Lewin 1987] Lewin, D. (2007) *Generalized Musical Intervals and Transformations*. New Haven, CT, and London: Yale University Press, 1987. Reprinted, Oxford and New York: Oxford University Press, 2007.

[Lewis 2000] Lewis, G.E. (2000) Too Many Notes: Computers, Complexity and Culture in Voyager. *Leonardo Music Journal*. **10**: 33–39,

[Miranda 2001] Miranda, E.R. (2001) Evolving cellular automata music: From sound synthesis to composition. In *Proceedings of 2001 Workshop on Artificial Life Models for Musical Applications*.

[Murail 2004] Murail, T. (2004) Modèles et artifices. Presses Universitaires de Strasbourg.

[Murail 2005] Murail, T. (2005) Spectra and Sprites. Contemporary Music Review. **24**(2/3): 137-147.

[Nguyen 2021] Nguyen, P. (2021) *Sophocles: Antigone (orchestral score)*. Vienna: Universal Edition.

[Nika et al. 2017] Nika, J. & Déguernel, K. & Chemla-Romeu-Santos, A. & Vincent, E. & Assayag, G. (2017) DYCI2 agents: merging the "free", "reactive", and "scenario–based" music generation paradigms. International Computer Music Conference, Shangai, China.

[Patchet et al. 2013] Pachet, F. & Roy, P. & Moreira, J. & d'Inverno, M. (2013) Reflexive loopers for solo musical improvisation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2205-2208.

[Pritchett 1996] Pritchett, J. (1996) *The Music of John Cage*. Cambridge University Press.

[Rocher et al. 2010] Rocher, T. & Robine, M. & Hanna, P. & Desainte-Catherine, M. (2010) A Survey Of Chord Distances With Comparison For Chord Analysis. *ICMC 2010*.

[Schat 1993] Schat, P. (1993) *The Tone Clock*. Routledge.

[Thom 2000] Thom, B. (2000) BoB: an interactive improvisational music companion. In *Proceedings of the fourth international conference on Autonomous agents, AGENTS '00*, pp 309–316.

[Tymoczko 2009] Tymoczko, D. (2009). Three Conceptions of Musical Distance. In: Chew, E., Childs, A., Chuan, CH. (eds) *Mathematics and Computation in Music. MCM 2009. Communications in Computer and Information Science*. **38**. Springer, Berlin, Heidelberg.

[Tymoczko 2011] Tymoczko, D. (2011) *A Geometry of Music*. Oxford University Press.

[Young 2008] Young, M. (2008) NN Music: Improvising with a 'Living' Computer. In Kronland-Martinet, R. & Ystad, S. & Jensen, K. (eds), *Computer Music Modeling and Retrieval. Sense of Sounds*, volume 4969 of Lecture Notes in Computer Science, pp. 337–350. Springer Berlin Heidelberg.