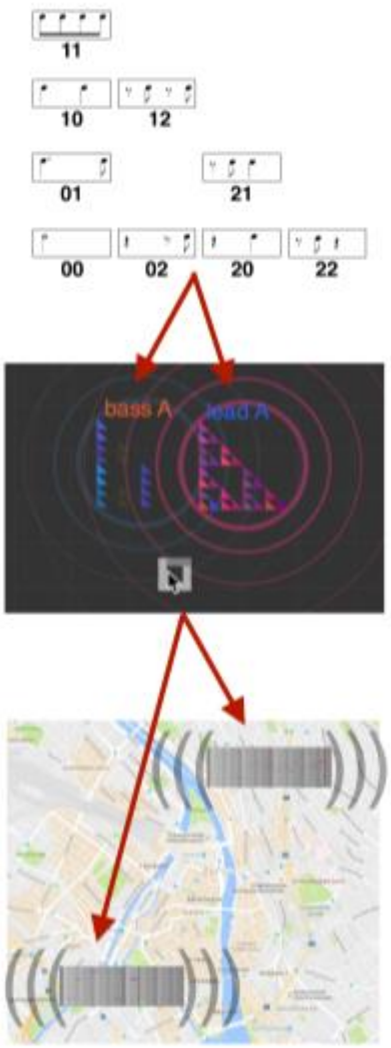
Variations on a Generative Rhythmic Landscape

Paper

Topic: **Music**

Author:
Jay Hardesty
Zurich, Switzerland
www.coord.fm

Abstract



Short, repetitive melodic figures are a common feature of various musical genres, including electronic dance music (EDM). The rhythms of such figures are localized to the extent that surface patterns and perceived structure can be algorithmically connected at a low level in order to create intuitively related variations without explicitly simulating compositional styles or strategies.

This paper describes creation of localized rhythmic variations using an approach that is *generative* in two senses of the word. It applies a finite set of rules to generate all rhythmic patterns that are well-formed according to strictly defined rhythmic relationships. Then it uses the resulting patterns as building blocks with which to dissect, compare and generate actual rhythms.

The set of all potential building blocks forms a hierarchy that encodes intersections of basic rhythmic expectation. That hierarchy has a self-similar structure that crystallizes the parallelism within and between those structural intersections into an enumerable set of surface subpatterns, facilitating reuse of the overall generative analysis to create specific rhythmic variations.

Rhythms that have been dissected into these building blocks can be manipulated at a high level of abstraction, enabling organic rhythmic variation via real-time improvisation.

jayhardesty@gmail.com

Key words: Generative music, music theory, number theory, rhythmic morphing

Main References:

[1] Hardesty, J. “A self-similar map of rhythmic components”, Journal of Mathematics and Music 10(1):36–58, 2016
[2] Hardesty, J. “Building Blocks of Rhythmic Expectation”, International Workshop on Musical Metacreation, Paris, 2016

Variations on a Rhythmic Landscape

Jay Hardesty
Zurich, Switzerland
www.coord.fm
e-mail: jayhardesty@gmail.com

Abstract

Short, repetitive melodic figures are a common feature of various musical genres, including electronic dance music (EDM). The rhythms of such figures are localized to the extent that surface patterns and perceived structure can be algorithmically connected at a low level in order to create intuitively related variations without explicitly simulating compositional styles or strategies.

This paper describes creation of localized rhythmic variations using an approach that is *generative* in two senses of the word. It applies a finite set of rules to generate all rhythmic patterns that are well-formed according to strictly defined rhythmic relationships. Then it uses the resulting patterns as building blocks with which to dissect, compare and generate actual rhythms.

The set of all potential building blocks forms a hierarchy that encodes intersections of basic rhythmic expectation. That hierarchy has a self-similar structure that crystallizes parallelism within and between those structural intersections into an enumerable set of surface subpatterns, facilitating reuse of the overall generative analysis to create specific rhythmic variations.

Rhythms that have been dissected into these building blocks can be manipulated at a high level of abstraction, enabling organic rhythmic variation via real-time improvisation.

1. Introduction

Musical composition and listening both involve an intuitive sense of musical structure versus musical surface. For instance, an improviser intuitively plays particular notes given a harmonic structure, and the listener intuitively perceives melodic structure upon hearing those notes. In this discussion *structure* refers to organization that a listener intuitively perceives when making sense of a *surface*, a concrete pattern of notes [1].

Structure that a listener intuitively attributes to music material might involve many levels including meter, grouping, phrasing, and higher level organization [1]. Short, repetitive melodic figures often occur within various musical genres, including electronic dance music (EDM); this discussion focuses on the rhythms formed by stripping away everything but the attack patterns of such figures. Those rhythms comprise surface patterns that can be connected to intuitively perceived structure at a relatively low level, independent of higher-level elements such as phrasing and harmony [2]. Such relative independence simplifies algorithmic generation of musical structures from musical surfaces and vice-versa. Encapsulation of that bidirectional capability in real time, even at a low rhythmic level, connects listening and composing to some degree, at a level of improvisation that negotiates local musical structures and surfaces.

This paper describes an approach that is *generative* in two senses of the word: as a finite set of rules to generate all items that are well-formed within a particular domain (such as language) [3], and as an algorithm for generating actual results [4]. An overview of the goals will be followed by an outline of the algorithmic analysis. Then generation of variations based on that analysis will be described. Finally, a scenario is presented of the algorithm in use, and the potential for ecosystems of musical material and musical apps is discussed.

2. Goals

Any computer-based approach to music creation is driven by particular goals. The goal might be to create entire works by modeling compositional strategies [5]. Or it might be to generate variations by modeling the style of existing music [6]. A survey of computer music approaches to composition be found in [7]. The system described here operates strictly on localized melodic material. It creates rhythmic variations generating low-level rhythmic structure and then uses that structure to generate new surface patterns. The low-level structure disentangles rhythmic anticipation and parallelism in order to form encapsulations of musical coherence. The aim is to create new note patterns that are recognizably related to the source material, but to leave higher-level structure in human hands or under the control of other processes.

This approach described here is motivated by two goals. The first goal is to enable organic, interactive improvisation by providing intuitive, real-time navigation of a fine-grained spectrum of coherent variations. The second goal is to inject variation into the playback of musical pieces, by mining low-level implications of existing material without intruding on higher-level compositional design.

2.1 Domain

The rhythmic patterns considered by this approach are constrained to short looping patterns of quantized, rhythms in strictly binary meter, because the implementation of this approach relies on a connection to the parity of binomial coefficients (a brief overview will be given in Section 3). It is hoped that the connection between music theory and number theory can be expanded so as to be less restrictive. For now, these constraints are tolerable because EDM also frequently contains looping melodic figures in binary meter, and it provides an amenable production environment for exploring variations within a larger musical context.

This paper focuses exclusively on rhythm, but the variations under discussion are meant to encompass melodic material. What is described here is variation of attacks within melodies; the calculation of corresponding pitches is beyond the scope of this presentation. Determining *when* and *if* notes should occur is not an entirely separate problem from determining *what* notes should occur, but it does pose sufficient challenges to warrant its own discussion.

3. Rhythmic Analysis

In this approach, analysis of two or more input rhythms is followed by generation of a new rhythm that is a variation on those inputs. More than one input rhythm is required because the variation of a given rhythm will only be specified in terms of other rhythms. That is, each variation is a rhythmic surface based on hybrid rhythmic structure.

3.1 Rhythmic building blocks

Generative analysis and construction is key to this approach. Rhythms are parsed into building blocks that encapsulate rhythmic anticipation and parallelism, two low-level features that are key to musical expectancy and coherence.

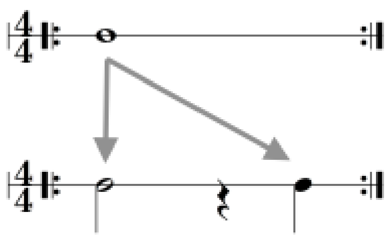


Figure 1. Elaboration

Anticipation is the unconscious expectation that a note on a weak beat at a given metrical level will be followed by a note on the subsequent stronger beat [8]. When the expected note does occur, the resulting pair of notes is an *elaboration* of that expected note. When the expected note fails to occur, its absence along with the remaining note is a *syncopation*.

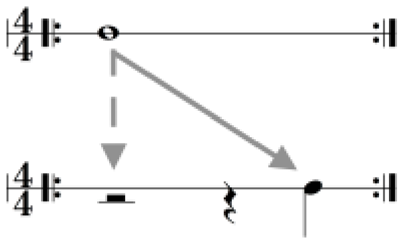


Figure 2. Syncopation

Parallelism arises as elaboration or syncopation is applied to elaboration and syncopation that has occurred at other metrical levels. For instance, a pair of notes that results from elaboration can itself be elaborated, meaning that the first pair occupies a relatively weaker metrical position than the original pair. Similarly, a syncopation might by elaborated or syncopated at another metrical level, and so on. Repeated syncopation can lead to perceived structure where surprise at one level becomes expected at a higher level [9]. This internal pattern formation creates repetition of anticipation along with anticipation of repetition, fusing key aspects of low-level rhythmic structure.

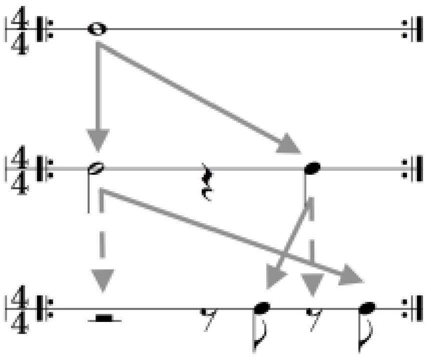


Figure 3. Syncopation of elaboration

An actual rhythm, even a short one, is a potentially complex intersection of rhythmic expectations [8]. Dissecting the rhythm into building blocks via generative rhythmic operations exposes every internal pattern that is formed purely of intersections of anticipation and parallelism. The building blocks form units of rhythmic coherence because the attacks within each building block evolved as a group solely via anticipation of repetition and vice-versa.

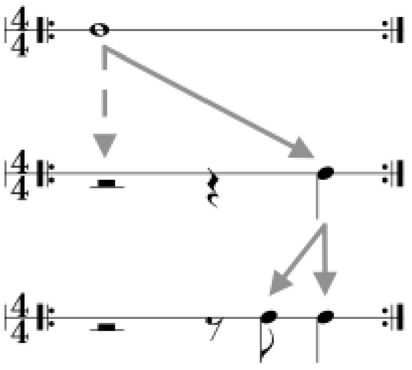


Figure 4. Elaboration of syncopation

3.2 Generative operations

The hierarchy of building blocks is defined as the set of rhythmic patterns that can be generated by applying one of the following rhythmic operations at each metrical level *m*, starting with a pattern that consists of a single attack on the downbeat [2]:

1. (Syncopation) Shift the rhythm one beat earlier at *m*.

2. (Elaboration) Shift the rhythm one beat earlier at *m* and combine it with the original rhythm, thereby doubling the number of attacks.

A vital constraint in the construction of a particular building block is that *at most one operation can be applied* at a given metrical level. Otherwise there would be an inversion of the relative beat strength of adjacent notes, and the building block would no longer strictly encode anticipation. This prohibition has an important connection to number theory that determines the structure of the hierarchy of building blocks. Details are in [2] but a brief sketch follows.

3.3 Elaborations, binomial coefficients and Pascal's triangle

When applied strictly to elaboration operations, the prohibition against applying two operations at a given metrical level mirrors the absence of binary (base 2) carries in the sum $n + k$ where $\binom{n+k}{k}$ is an odd number [10][11]. As a result, the patterns of odd binomial coefficients on rows of Pascal's triangle look exactly like the set of rhythms that comprise the building blocks [2].

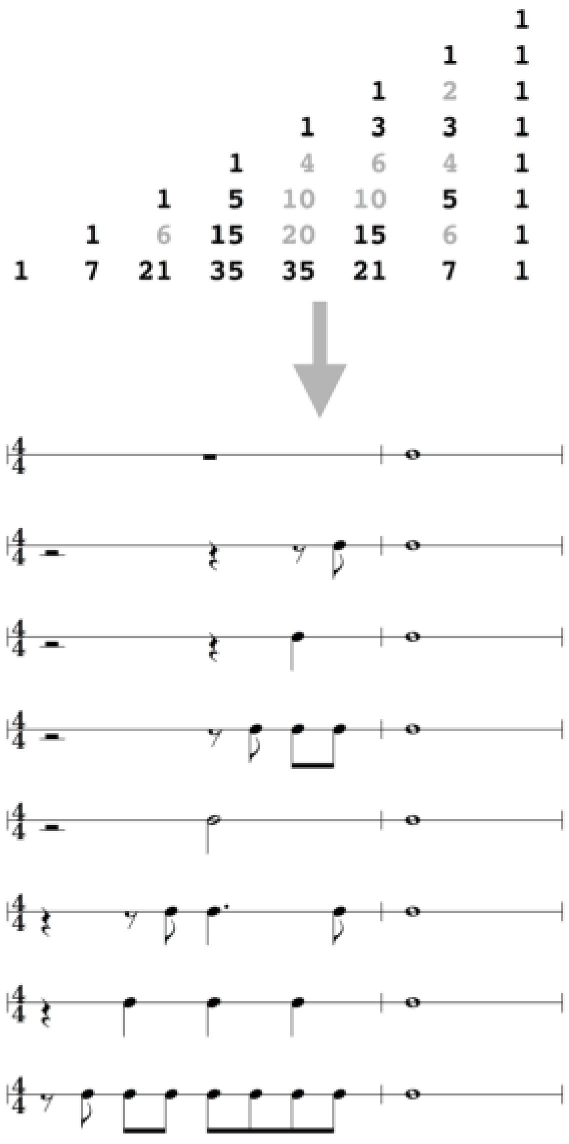


Figure 5. Odd binomial coefficients on Pascal's triangle mapped to elaborations and nested elaborations.

The patterns on Pascal's triangle can also be interpreted another way. The position of each odd coefficient on a row represents an allowable sequence of syncopations to be applied to a particular sequence of elaborations represented by another row [2]. The position of each syncopation offset and the row number containing the corresponding elaborations will share no binary bits, the presence of which would indicate that elaboration and syncopation had been applied at the same metrical level.

3.4 Syncopation and the Sierpinski gasket

The organization of elaborations and syncopations on Pascal's triangle corresponding to binary numbers that share no binary bits suggests an encoding that combines the binary number into a ternary one, since only three out of four possibilities is allowable at each corresponding pair of binary places (a 1 in that place in both numbers being prohibited) [2].

This ternary encoding forms an address scheme for the Sierpinski gasket, where each digit indicates a sub-triangle of a sub-triangle, and so on. This self-similar mapping makes it possible to enumerate every facet of kinship between rhythmic surface patterns that gives rise to intuitive structure based strictly on low-level anticipation and parallelism. The mapping also establishes that the order in which generative rhythmic operations is applied has no effect on the final result, meaning that rhythmic patterns can evolve along many different pathways while maintaining that kinship in various forms.

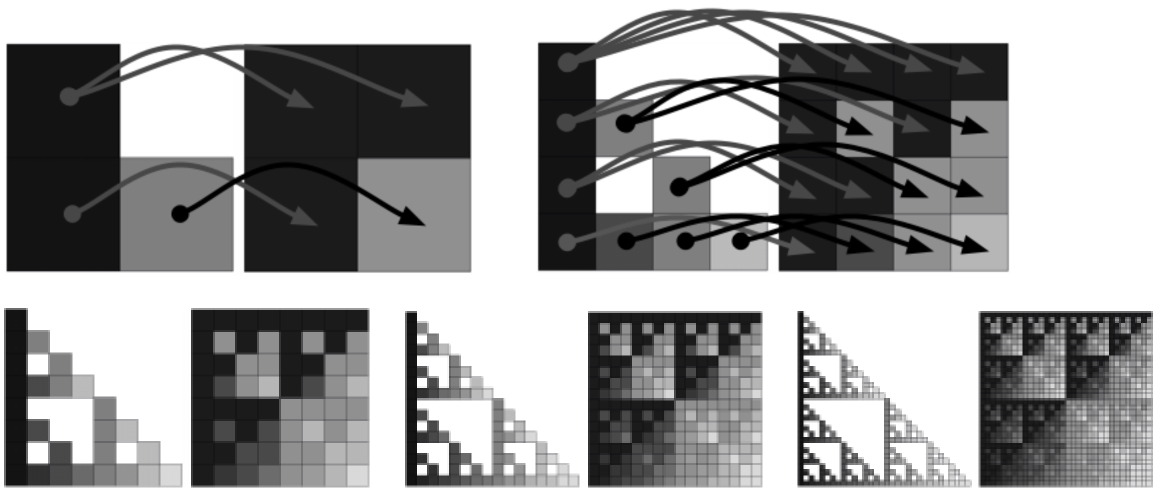


Figure 6. Locations on the Sierpinski gasket mapped to (reversed) call allowable combinations of elaboration and syncopation, shown at up to five metrical levels.

3.5 Encoding building blocks

The ternary encoding mentioned above makes enumeration and characterization of the building blocks extremely straightforward. Each building block is encoded as a ternary (base 3) number where the digit at each place indicates the operation applied at the corresponding metrical level *m* as follows:

- 0 indicates no operation at *m*.

- 1 indicates elaboration at *m*.
- 2 indicates syncopation at *m*.

With this encoding the set of building blocks is simply the sequence of integers from 0 to 3^n-1, given *n* metrical levels [2]. For example, the building block that consists of only a single attack on the downbeat had no elaboration or syncopation operation applied at any metrical level, and so it has a 0 at each ternary place, making it simply 0.

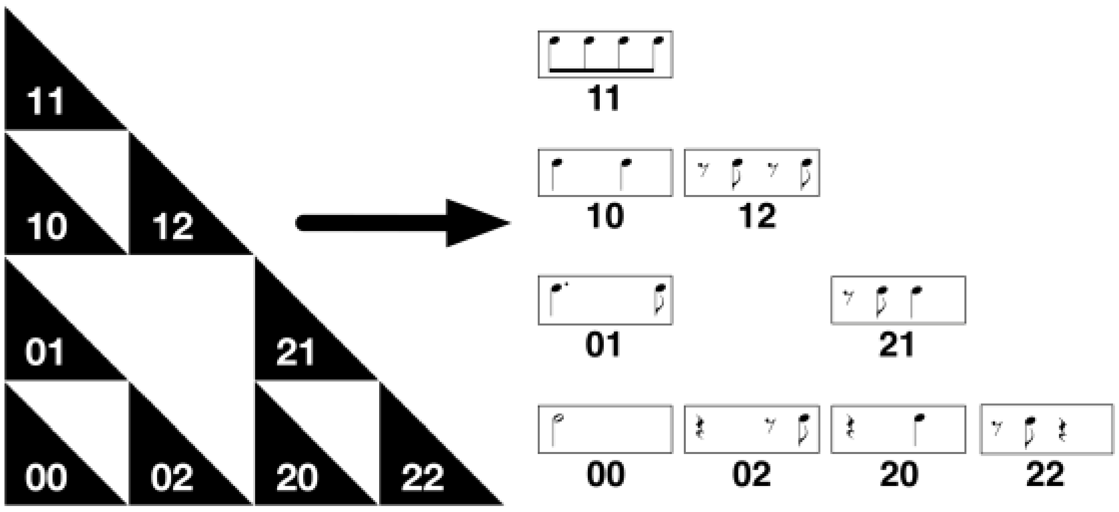


Figure 7. Building blocks encoded as addresses on the Sierpinski gasket [2].

This encoding is the basis for a distance measure used by the variation algorithm and described in Section 4. Since each ternary place in the encoding corresponds to a particular rhythmic operation, the Hamming Distance between the ternary digits in the encodings of two building blocks indicates the number of differences in the evolution of those building blocks [12]. The more digits two building blocks share at respective ternary places, the more common rhythmic evolution there is between those building blocks.

3.6 Actual rhythms

A very important point is that actual rhythms, whether inputs or outputs to this approach, are not assumed or intended to be well-formed in the manner of the building blocks. A rhythm that is parsimonious with the building blocks is not judged to be better, and it not more likely to appear in the output. *The analysis is merely intended to expose where particular low-level intuitive structure accounts for a rhythmic pattern and where it does not*. In fact, some popular and compelling rhythms can be distinguished by the fact that they are not parsimonious with the building blocks. In particular, this is the case with rhythms related to the *clave* pattern, a pervasive presence in much music of the past century [13].

The goal is not to create a new kind of music that sounds “generative” or “fractal”, but rather to obtain intuitive handles on kinds of music that already exist**.**

4. Rhythmic Morphing

As mentioned in Section 3, the only specification used to describe how a rhythm should be varied is other rhythms. The user controls the relative influence given to each of those rhythms. A new rhythm is generated by considering by each time point (typically each 16th note position) and assigning the likelihood of an attack occurring at that point, based on the weighted influences of the input rhythms [2].

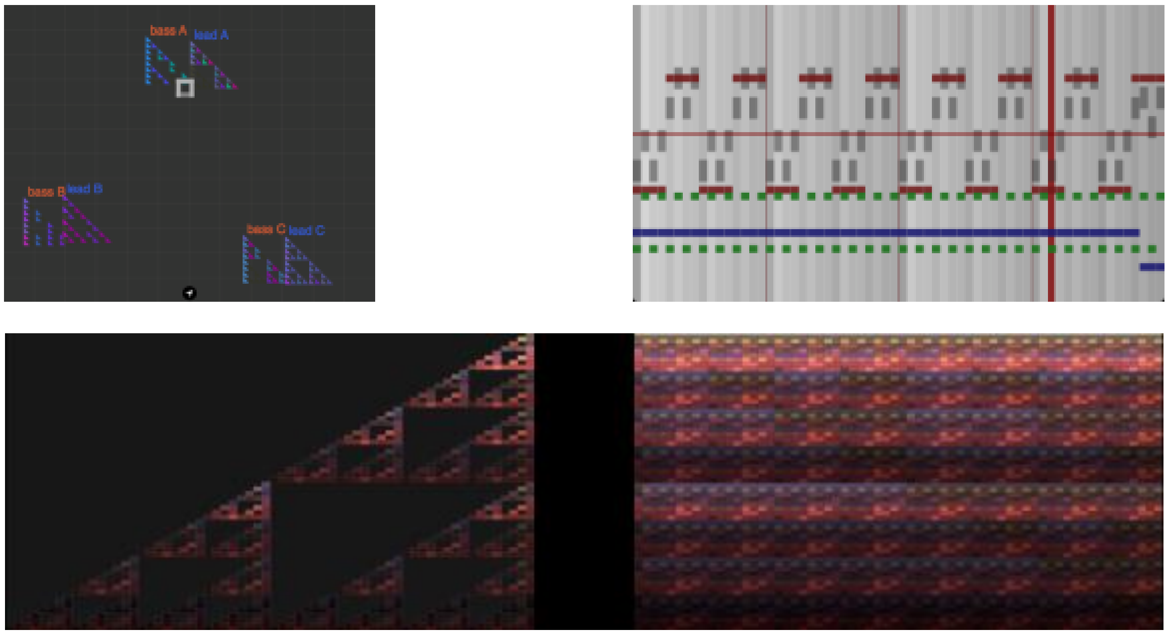


Figure 8. Navigation in the top left pane determines rhythmic variation based on likelihood of an attack at each time points, as displayed in the top right pane. At bottom left building blocks show activation on the Sierpinski gasket based on input rhythms and weights, with corresponding rhythmic structures shown at bottom right.

4.1 Building block ternary representations

Each input rhythm is parsed into a set of building blocks. Each building block is encoded by a ternary number *address*, which can be split into two binary numbers *a*, and *b,* the digits of which correspond to the digits in *address* at each place as follows:

- For 0 in *address*, set *a* = 0 and *b* = 0.
- For 1 in *address*, set *a* = 1 and *b* = 0.
- For 2 in *address*, set *a* = 0 and *b* = 1.

For example, the ternary *address* = 121 is equivalent to the combination of binary *a* = 101 and binary *b* = 010.

The number *a* encodes the combined elaboration operations that generated the building block, with a 1 at each binary place corresponding to a metrical level where elaboration occurred. (This is the row number on Pascal's triangle where odd entries form the same pattern as the rhythm). The number *b* encodes the combined syncopation operations, with a 1 at each place corresponding to a metrical level where syncopation occurred. (This is the position of an odd entry on row $2^n – a – 1$ of Pascal's triangle, given *n* metrical levels).

For *n* metrical levels, the set of displacements earlier in time (from strong to weak beats) generated by *a* is:

$$D = \{x \mid x = x \& a\}.$$

and the set of attacks generated by *a* and *b* is:

$$P = \{x \in D \mid (2^n - x - b) \ mod \ 2^n\}.$$

Throughout this discussion time points are numbered starting with 0 [2].

Using ternary address 121 again as an example, the associated binary number encoding elaborations is 101, indicating elaborations at the highest and lowest metrical levels, but not the middle one. A set of time displacements that satisfies these constraints is {0, 1, 4, 5}, corresponding to the rhythm 10011001. Applying the syncopation offset 010 then produces the rhythm represented by address 121: 0110011.

4.2 Morphing algorithm

A high-level description of the process for generating a rhythmic variation is as follows [12]:

1. Parse each input rhythm *R* into the set of all building blocks *A* that consist solely of attacks in *R*, and which are not subsets of any other building block in *A.*
2. For *m* metrical levels and $n = 2^m$, let $\boldsymbol{v} = (v_1, v_2, \ldots, v_{n-1}, v_n)$ with all entries initially 0.
3. For each time point $t < n$:
    a. Find the set *B* of all building blocks that include *t*.
    b. For each building block *b* in *B*
        i. Set *d* equal to the minimum Hamming distance between *b* and any building block in *A*.
        ii. Set $v_{t+1} = \max(v_{t+1}, 2^{-d})$.

If *t* is an attack in R, then v_{t+1} will equal 1. Otherwise v_{t+1} is halved, starting with 1, for every mismatched generative rhythmic operation at a particular metric level between the closest potential building block that contains t to any building block in the parsing of *R*.

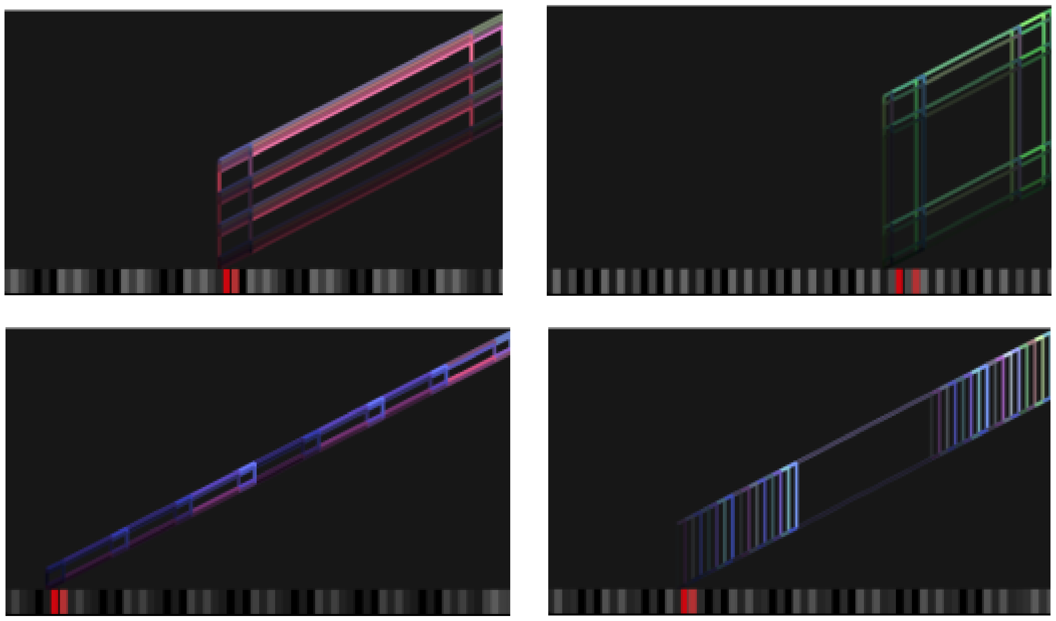


Figure 9. Different building blocks project rhythmic expectation at different times during playback. The expectation at the playback head (shown in red) occurs in parallel with expectation at other times, as shown by the displayed symmetries. Points on the time line along the bottom of each window are darker where expectation is higher, indicating that other building blocks are also at play.

In other words, we assign an attack likelihood to each rest in *R* by asking: *since none of the rhythmic evolution paths that produced the input rhythm includes a given time point, how close is any of those paths to one that would?* We calculate an amount of rhythmic expectation projected on that rest, an amount that falls beneath the threshold of triggering an attack (that threshold having a value of 1 by default). This is computationally tractable because all possible evolutionary pathways can be readily enumerated and interrelated in terms of their encodings.

Finally, these vectors are weighted according to the user's actions and summed. For example, given attack vectors ***u*** and ***v*** and weights *w1* and *w2* a new vector is calculated:

$$\boldsymbol{x} = \mathrm{w1}\cdot\mathbf{u} + w2\cdot\boldsymbol{v}$$

Any time point in ***x*** with a value that exceeds a threshold determined by the user is assigned an attack; others are assigned a rest. Lowering the threshold increasing the density of the resulting rhythm; raising the threshold makes the rhythm more sparse.

4.3 Landscape of rhythmic variations

A plane of Cartesian coordinates effectively captures the selection of musical targets and navigation of musical results [12]. Each musical target, for instance a looping bass or lead part, is placed as a landmark on the plane, and the cursor is moved across the plane during playback to explore different variations. As the cursor approaches particular musical targets, those targets receive a larger weight in the algorithm. When the cursor is directly on a target, the result sounds just like that target. At intermediate points the result sounds like a variation that shares evolutionary structure with each target, relative to the distance to that target.

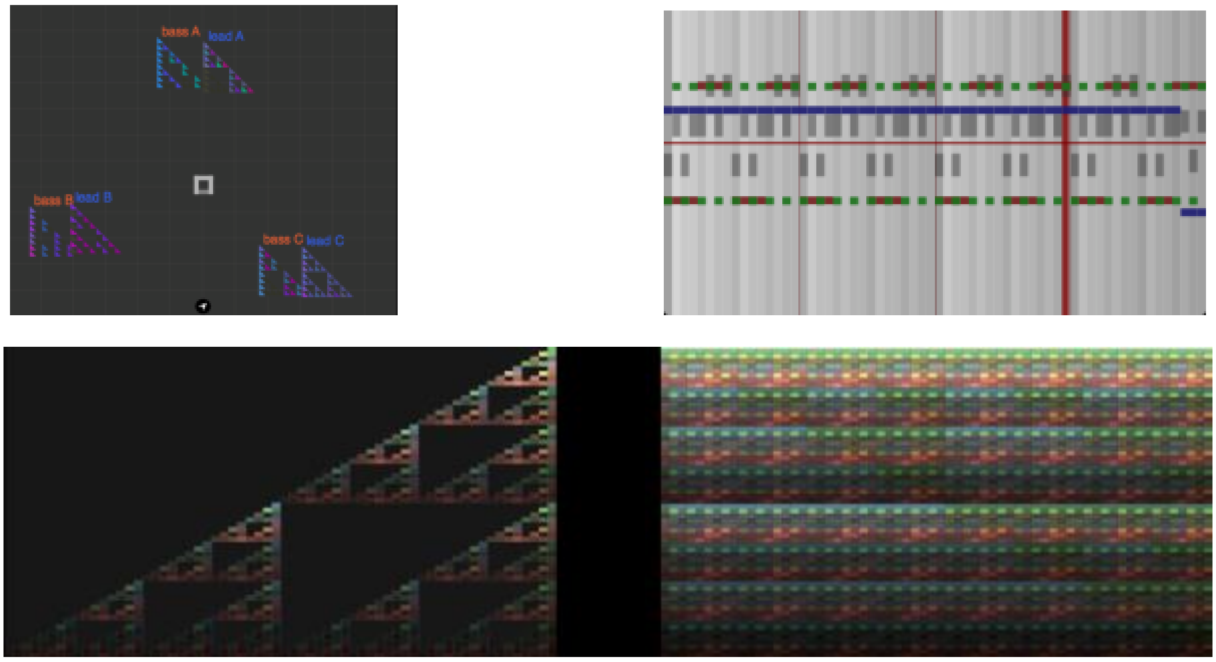


Figure 10. Visualization as presented in Figure 8, but with different weights. Three musical pieces (each represented by lead and bass loops) are musical landmarks placed on the variation landscape at top left. Their relative influence is color coded by musical input: the piece at the top of the landscape is red, the one at lower left is green, and lower right is blue.

Steering by ear in real-time, immediately adjusting location based on the musical results, the user generates a stream of rhythmic variations, effectively generating a concrete musical surface from an intersection of abstract music structures, collapsing a large number of branching rhythmic relationships based on low-level musical expectation in the selected musical targets. Fine-grained improvisation at this level, devoid of any rule-based or statistical results, could be argued to constitute musical improvisation that captures some low-level intuitive aspects of typical musical improvisation.

5. Applications

Localized rhythmic variation as described above, when combined with pitch variation, could insert mutability into a number of musical scenarios. The common elements in each of the following applications are the following:

- Multiple input parts are required; the application does not generate music from scratch.
- Some form of interactive or higher level control is required; the application does not model human actions or strategies, and it does not evaluate its own output.
- The application acts on short, looping melodic or rhythmic elements within a larger musical piece. Other elements of that piece are left undisturbed or under other control.

5.1 Music composition and performance

The most straightforward application of this approach is as an aid to creating musical output in conjunction with a music production app such as Ableton Live [12]. The composer selects one or more note-based (for instance, MIDI or OSC) parts; these are called *tracks* in Live. On each track, one or more alternate versions are specified; these are called *clips* in Live.

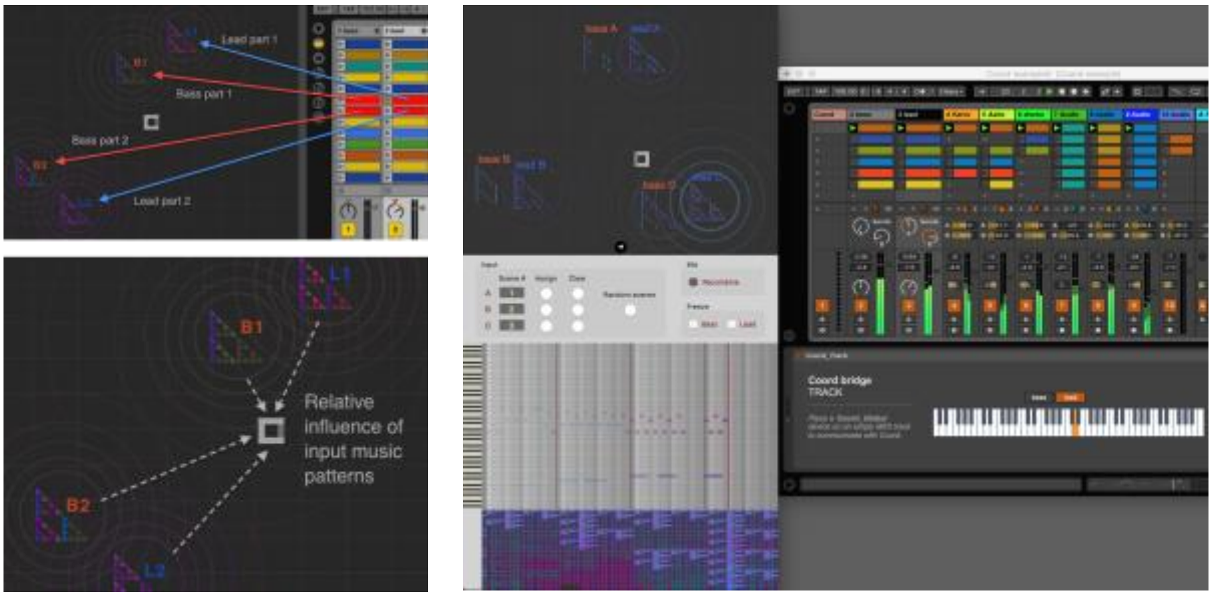


Figure 11. Coord app used to morph monophonic MIDI clips in Ableton live during playback. The Swift-based app communicates with Live via OSC and Max-for-Live.

The selected clips on a given track are used as the music inputs described in Section 4. Rhythmic morphing can be carried out, using those clips as musical targets, and the composer steers the musical results by adjusting distance to those targets.

Resulting variations are streamed in real time to the output of the track that contains the clip or clips, as musical variations that replace the original output of that track. The current implementation consists of a Max/MSP devices placed on tracks in Live; these devices communicate with a separate app called Coord that performs the musical analysis and interactive generation of variations. The Max devices communicate via OSC, supplying asynchronous note data from Live to Coord, and receiving streaming note patterns back to Live, which are output to the respective track. Coord provides a

UI for collectively morphing and mutating the tracks.

With this setup the composer can generate new material that is purely a function of existing melodies and rhythms, which is recognizably related to the original material but in nonobvious (non-mechanical and nonrandom) ways. Other elements of the larger piece are left in the composer’s hands. (Coord does also employ an algorithm for remixing clips on accompanying tracks, but that is beyond the scope of this discussion.) This approach is useful for generating new musical ideas out of existing ones, or for varying material in a live performance [12].

5.2 Variable music playback

A listener who likes a piece of music will typically listen to that piece on multiple occasions. Over repeated listenings the listener is engaged by a budding sense of what to expect as the piece unfolds, against the backdrop of general expectations formed by other music that is familiar to the user. But repetition becomes less rewarding once the new piece becomes overly familiar, and monotony replaces the integration of new expectations.

Playback that incorporates low-level variation and rhythmic morphing of selected tracks would restore some degree of the uniqueness that music had before recording technology. It would require a rendering engine that was not merely playback of stereo audio tracks, and it would require some interaction or process to steer the variations.

Music that was published to target such playback would consist of audio parts intended for combined fixed tracks together with note-based data and audio rendering for each variable track. It could be made possible for variable parts to be shared across variable pieces, and for resulting variations to be captured and reused as sources for further variation.

5.3 Ecosystem of musical material

Morphing variations could support an ecosystem of musical elements, where preferred variations become inputs for further mutations. Composers might allow musical elements from other composers to be used as inputs to their own work, as well as to create palettes of their own material musical material for others to use as inputs. Listeners could drive creation, selection, sharing, and reuse of preferred variations.

Location-based apps could make such an ecosystem explicit, by projecting the morphing landscape described in Section 4 to physical locations, where users tag locations with preferred variations generated at other locations based on musical tags made by others. Overlaying the map with multiple planes would add an additional dimension and provide channels where cooperation and competition could self-organize based on location and on collective musical judgements.

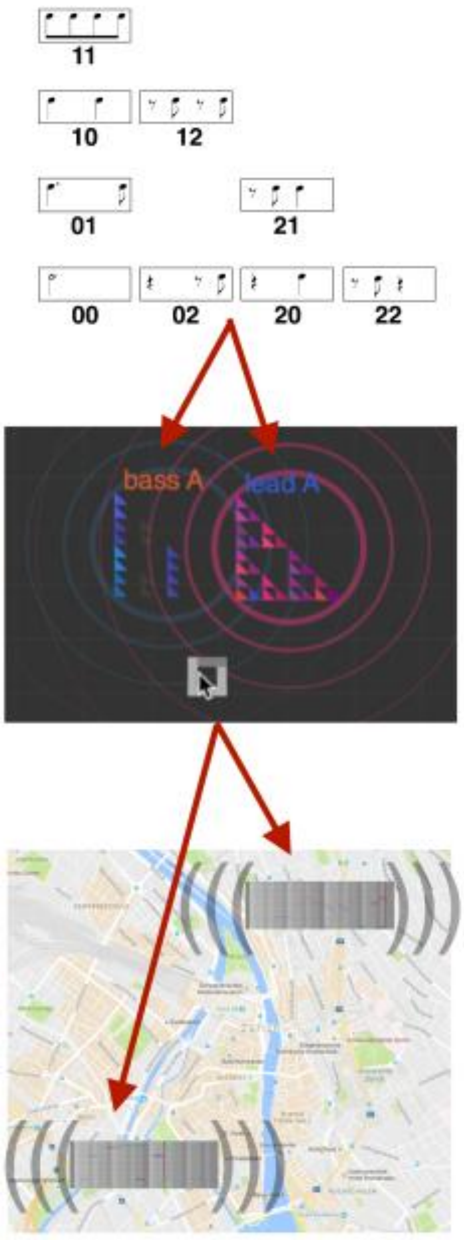


Figure 14. Morphing music, tagging locations with results which become inputs to further morphing and subsequent tagging at other locations. A multiuser scenario that casts humans in the role of bees and music in the role of pollen.

5.4 Ecosystem of musical apps

Variable playback and real-time improvisation exist at opposite ends of the spectrum of possible applications in terms of engagement, being relatively passive and completely engaged respectively. Other scenarios could include more selective engagement that allows feedback based on results and responsiveness to actions, leading to apps where the listener shapes musical results by ear by intersecting familiar musical material and particular situations.

5.5 Video demonstrations

Some examples of Coord in use are online at http://coord.fm/videos.

6. Conclusion

Musical creation that leverages psychological aspects such as expectation and intuitive organization can be accomplished at a localized, low-level by taking a generative approach that exploits a mathematical connection between basic rhythmic operations and self-similarity formed by patterns of binomial coefficients.

Being low-level, this approach does not intrude on compositional strategies, actions, or judgments made by humans. It does however enable human-machine interaction that provides a degree of the fine-grained expressiveness usually accessible only to trained musicians. The goal is to leverage human musicianship, including that involved in listening, rather than to simulate or replace human activity.

6.1 Future directions

Finding complementary roles between the approach described here and approaches involving machine learning, statistical style modeling, musical grammars, signal processing, and algorithmic arranging is hoped to bring greater expertise and theory to bear.

Immediate tasks include extending the theory to handle non-binary rhythms and extending the approach to handle multiple voices and harmonic structure.

References

1. Lerdahl, F., and Jackendoff, R. S. 1983. *A Generative Theory of Tonal Music*. Cambridge: MIT Press.

2. Hardesty, J. 2016. A self-similar map of rhythmic components. *Journal of Mathematics and Music* 10(1):36–58.

3. Chomsky, Noam. Three models for the description of language. *IRE Transactions on information theory* 2.3 (1956): 113-124.

4. Galanter, P. What is Generative Art? Complexity theory as a context for art theory”. in *International Conference on Generative Art*. 2003. Milan, Italy: Generative Design Lab, Milan Polytechnic.

5. Cope, David, and Melanie J. Mayer. 1996. *Experiments in musical intelligence*. Vol. 12. Madison, WI: AR editions.

6. Pachet, F. The Continuator: Musical Interaction with Style. In ICMA, editor,

Proceedings of ICMC, pages 211-218, Göteborg, Sweden, September 2002. ICMA.

7. Nierhaus, G. 2009. *Algorithmic Composition*, Vienna: Springer-Verlag.

8. Huron, D. 2006. *Sweet Anticipation: Music and the Psychology of Expectation*. Cambridge: MIT Press.

9. Temperley, David. 2001. *The Cognition of Basic Musical Structures*. Cambridge, MA: MIT Press.

10. Kummer, E. 1852. Über die ergänzungssätze zu den allgemeinen Reciprocitätsgesetzen. *Journal für die reine und angewandte Mathematik* 44:93–146.

11. Lucas, É. 1878. Théorie des fonctions numériques simplement périodiques. In *Amer. J. Math*., 1. 184–240.

12. Hardesty, J. 2016. Building Blocks of Rhythmic Expectation. In *Proceedings of the Fourth International Workshop on Musical Metacreation*, Paris, 2016.

13. Toussaint, Godfried T. 2013. *The Geometry of Musical Rhythm: What Makes a “Good" Rhythm Good?* Boca Raton, FL: Chapman & Hall/CRC.