Artik: Articulation and Dynamics from Structural Pattern Analysis

Eliot Handelman and **Andie Sigler**

Abstract

Automatic arrangement is the problem of transforming an input score into a full orchestration, e.g. realized with samples. A subproblem, on which arrangement depends, is to establish the dynamic levels of individual parts, their *articulations*, and how parts are to align. A new kind of *structural* analysis is described: the interpretation of music means the delineation of existing structure, so that the performance *synchronizes* with the music. These ideas are realized in a system that creates fully articulated string orchestra renditions of concerti by Bach and Vivaldi. While still at an early stage, the system demonstrates how structural analysis can be deployed to solve complex musical tasks. We conclude with some speculations about the future of musical practices as suggested by this research.

Introduction

The *articulation problem* is, very roughly, to assign to an input score dynamics and instruments with their specific performance manners – springing bows, *Flatterzunge*, *cuivré* – such that the result is "musical." While difficult or impossible to *define* musicality, it is easy to understand what it isn't. Get someone to listen to music they like on headphones while you add dynamics by randomly twisting the volume knob. Did they like it? You effectively added a noise layer unrelated to how the music was behaving. Dynamics and articulations should not seem like an extra layer, or noise, but part of the music, all factors operating in concert. In some sense, dynamics, articulations, orchestrations – all called "articulations" in this paper – must be *synchronized* with or to the music.

The word "articulate" once meant "joint," referring to how parts of things are connected to form wholes. Articulation implies *difference* with the aim of constructing new groupings. The groupings must relate to the behavior of the music. Orchestration, for instance, should not be a haphazard layering of available sounds, but should add or a develop structural levels that bring out structures of the input score. We show how this idea can be made precise.

Previous research

Related research is reviewed in (Kirke and Miranda 2012) – methods for "expressive music performance." While performance generation is allied to our articulation problem, we point out a few significant differences.

First, performance methods lean to the modeling of conventions, whether by rule or by machine learning, with the aim of "sounding human," whereas no attempt is made here to model a human performance. Articulation is instead conceived as a *sonification of structure*, an elucidation of a constructed perspective on an input score. The idea is to induce an articulation over a structure. The hope is that though semantically agnostic, the results might nevertheless make sense from semantic points of view. But we also wish to generate "creative" (unconventional) interpretations, and in fact to investigate the problem of creatively deploying immense sample sets without being bound to conventional articulative precepts.

Second, performance methods work by modeling local structures – phrases and smaller note-groupings. In the case of phrase-structural models (Friberg, Bresin, and Sundberg 2006; Todd 1995), each phrase is shaped using a general rule on features such as internal boundaries, scale degrees and registral high points – without attention to the particular musical relations within or among phrases. For case-based and statistical methods (Suzuki, Tokunaga, and Tanaka 1999; Widmer 2001; Kim et al. 2012), a database of local note-groupings along with their performance parameters is queried in the construction of a new performance. Our model seeks to unite local and global structure, so that local contrasts operate within a set of relations spanning the entire piece.

A structural approach

In this paper, the approach is entirely *structural*. A *structure* is a set of notes (or other values) for which a well-defined property holds. It can be reasoned about without information loss.[1] With a notion of structural composition, we build more and more complex structures that we are still able to reason about. In this paper we are concerned mostly with *patterns* of simple structures. We construct compositions of

[1] An example is the *Z-chain*, described in (Handelman, Sigler, and Donna 2012), used there to generate orchestrations.

such patterns, showing how a *low*-dimensional "summary" structure can be derived from this space. The result is interpreted as an "articulation vector," that can be used to model arbitrary musical features, such as dynamics, selecting samples for given instruments, etc.

The notion of (well-defined) structure is in contrast to *semantic* objects, for example the *phrase*. There is no universal computational definition of a phrase, since the delineation of musical phrases is subject to human interpretation.[2] While it would be possible to invent a well-defined structure and *call* it a phrase (i.e. use it as a stand-in for the semantic concept of *phrase*), any use that depended on the correctness of the semantic tag would be subject to error. Therefore a purely structural approach would prefer to remain agnostic about semantic values, while relying on the constructive and combinatorial properties of well-defined structures to elucidate relationships within a score.

Background: the articulation problem

Articulating music is not a trivial problem. Not everything works: as suggested, it is necessary in some sense to "follow the music," lest we merely add additional layers of noise. The music may have distinctive parts, which could be orchestrated distinctively. But should they be loud or soft? Musical cues might suggest one way or another, but in principle there is no reason why anything *must* be performed in any given way, subject to being performable.

Dynamics should be dynamic: some parts of the music are soft, others building, others with sharp contrasts, etc. Although there is no a priori reason why any note *must* be either loud or soft, there is nonetheless a more global condition we should try to fulfill: we should choose louds and soft in such a way as to create larger groupings and patternings, preferably at different scales. There might be a long soft part, a mostly soft part, quick alternations, longer and more complex alternations, etc. In other words the loudnesses should generate structures.

Articulation encompasses a number of different problems, including segmentation, grouping, sectional boundaries – problems for which general solutions do not exist. The boundaries must (or might) then be *realized* with specific manners, creating distinctions in large and small parts.

In multi-part music, *polyphony* issues arise concerning the relative foreground and background of voices, and the problem of designating a "Hauptstimme" or main voice, if there is one. *Melodic* articulation pertains to "note-wise" performance within a given voice – how notes are to be *connected*, or broken up into smaller groups, or are to have their own inner lives: how notes become *sounds*. Anything that breaks up a homogeneous flow is an articulation. These are not merely *markers* but are appreciated in themselves, and in how they bring the whole to life.

[2]Computational approaches to phrasing (and to all problems of semantic interpretation) are either statistical or heuristic (Pearce, Müllensiefen, and Wiggins 2008; Cambouropoulos 2001), as opposed to being *structural* in our strict sense. Other semantic objects in music can include *bass-line*, *melody*, *theme*, and even – in the case of audio input – *note*.

The articulation problem includes the following components:

1. Establish groupings (parts, wholes, sections, structures)
2. Establish dynamic (general assignments of loud/soft)
3. Establish articulation (manner of performance)
4. Establish relations between parts (e.g. synchronizations)
5. Make timbral assignments (choose samples)
6. Distinguish foreground and background. (general levels, "automatic mastering")

Some of these problems are not treated in this paper, in particular the last. This would necessarily touch on structural distinctions between melody and accompaniment, a problem we currently avoid by working with polyphonic (as opposed to homophonic) music.

The general problem is to transform a pitch and onset sequence into a sequence of samples. In this transformation we are moving from one conception of music, which is note based, to another, which is sound-based, from the discrete *points* of a notational system to the continuum of perception. One "note" might now encompass a complex internal structure that is realized by N samples, (potentially with large N). The notes themselves do not provide sufficient information about how to generate sound: to do that we must somehow generate the information from within the music itself, that is, analytically. This is the problem of generating *supplemental structure*.

The typical sample library user has to key in every change of articulation manually, a tedious labor. As the number of available articulations increases, automation is desirable. However, this is tantamount to the problem of creating an *interpretation*, a difficult problem not yet addressed in the context of large multi-instrumental sample libraries.

Structure

Nonlinear Structure

Consider a one-to-one mapping of pitch to intensity. Every note of the same pitch maintains the same dynamic throughout a piece: for example bass soft, soprano loud. What's wrong with this, e.g. as applied to a Beethoven sonata? A set of predictable relations arise which are independent of otherwise perceived musical structure. Repeated notes will always have flat dynamics, with no possibility of accenting in the Beethoven manner. A new structure, the linear map from pitch to dynamic, is introduced – and this structure is not, in general, characteristic of music.

A historical example will help. The young Wagner faced this problem in an overture that featured a *fortissimo* drumstroke on the second beat of every fourth measure. The audience's awareness turned into derision and the composer was forced to flee (Wagner 1983). There is such a thing in music as *over-predictability*, and this certainly applies to a *linear* structure. There is something anti-musical about it.

Wagner might have fixed his piece by making the drum strokes a bit more irregular, playing with expectations. He might have used an *entropic* solution similar to the "humanizers" of midi sequencers (which add Gaussian noise). But

the problem with the persistent pattern is *structural*: a pattern that is "too simple" – linear – should be fixed *structurally*, not statistically: not adding noise, but adding structure.

But what then *is* a structure, and what makes some structures musical? This is a rather subtle problem, first, because it has no definitive solution: its judgment is a matter of taste. At best, we can express a *principled* articulation, where we claim that the staccatos and legatos reference and articulate the musical units.

In a simplified version of the problem, we wish to generate a bit vector B which we can use to represent legato/detaché. (In a rendering, this determines whether to use a "legato" sample.) Now in some sense, B should be synchronized with musical structures. One might think of "phrasing," but we need a finer grain of articulation, since we would not ordinarily think it musical to perform phrases *either* all legato or staccato, nor should each phrase necessarily follow a similar rule. In fact we will need to query every note in the *context* of the entire piece.

We call this approach *structural magic*: we begin by taking some structural analysis of the piece, and reducing its dimensionality to one bit per note, giving a per-note oracle. *Magic* takes place *if* this conversion *does* conserve structure. That is, we have somehow drawn, as if by a magic spell, inscrutable qualities of the whole into a per-note query mechanism that takes the whole into account. All of this must take place in a mathematical universe of course.

Patterns

The term "pattern" invokes looping music, and would seem to be irrelevant for anything non-looping, but this is not so: patterns are fundamental structures of all music. Patterns refer to sequences of *recurrences* of any feature of interest: we can speak e.g. of the pattern of entries in a fugue, of the intensity of the downbeat, the orientation of the melodic interval between two beats. The property will take some value at different points in the input score, and the sequence of values so obtained is a pattern.

Some brief precision. A "pattern" is a fixed-length sequence of terms, and extensionally anything that such a sequence can identify. The same term has the same extensional identity. Terms are identifiers drawn from the alphabet. A term must occur in the pattern more than once. Things that occur just once are assigned to •, read as "throw": it is a place where the pattern breaks or is "thrown." Terms begin with A, and subsequent new terms are alphabetically ascending. We write ABA as $A \bullet A$. When terms represent structures, recurrences are taken as *isomorphic*, as a single object with an expansion that places it in time.

Pc_N patterns

Pc_Ns are defined as maximal temporally contiguous collections of exactly N different pitch classes, from 1 (which includes all unisons and octaves) to 12 (trivially, a whole piece that uses all 12 pcs). All Pc_Ns can be efficiently identified in a score. As shown in (Handelman and Sigler 2013), some Pc_Ns uniquely determine a key, and are useful for tonal analysis. But Pc_Ns also turn out to be extremely useful in creating structural articulations.

Pc_Ns can be *normalized* with Forte's normalization procedure (though we keep inversionally equivalent sets separate). The *set* of normalized Pc_Ns then corresponds to all possible transpositions and orderings of the underlying pitch-class set. For example, Pc_{02} is the set of all sets of wholetone steps (modulo octaves) in an input score.

Intuition: Sectional Pc_NPartitions

To gain an intuition, here is a simple example of what Pc_Ns can do: they can create sectional partitions in complex pieces. This makes sense because different sections must technically realize difference with a restriction of material, say of intervallic combinations, and this is just what Pc_Ns capture. In the Anna Magdalena Bach Polonaise (Figure 1), Pc_{03}, the minor 3rd standing on its own, occurs only in the odd sections, and Pc_{047}, the major triad, mostly in the even sections. The overall formal pattern is clearly $ABABA$.

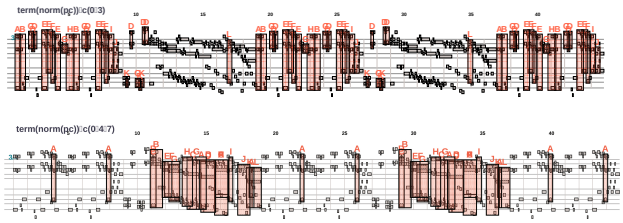

Figure 1: Pc_Ns reveal formal zoning in the Anna Magdalena Bach Polonaise. Top shows occurrence of Pc_{03} and bottom shows Pc_{047}.

Consider now an articulation. Using the imposed formal pattern $ABABA$, we could choose one articulation for A, another for B: e.g odd parts go to strings, even to woodwinds. But this would give a rigid contrast. Were we instead to assign instruments to particular Pc_Ns, randomly but consistently, then the structure of the piece itself would determine how the instruments group: the odd and even sections will automatically differ. But whereas in the formal pattern the articulation groups are encapsulated, here bits of the sections overlap: imagine, for example a trumpet, with brief accents in the odd sections, coming into its own in the even. The result, structurally, is more complex than an orchestration entirely based on $ABABA$. No assertion is made whether the result is esthetically more satisfying: we regard that as an *esthetic* model, which we do not provide.

Interval patterns

Let us now go into a bit more detail. As observed, Pc_{01} is the set of all sets of semitones in an input score. This can act as a filter: we see the music only in *one* aspect. What is left may nevertheless be highly structured.

Figure 2 shows the half steps of a simple but interesting Finnish tune. The sequential pattern of $\{0, 1\}$ is expanded in three different ways, labeled A, B, C. Each box provides a *sequence* of pcs. Here A is $(0, 0, 1, 0)$, B is $(0, 1, 0)$ and C is $(1, 0, 0, 0)$.

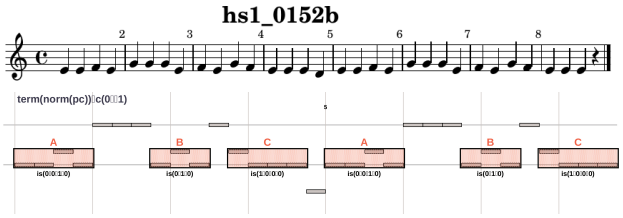

Figure 2: Half steps in a Finnish tune.

What sort of structure can we discover in these sequences? We need to transform them further, e.g., by alignment (the shaded entry suggests a translated symmetry):

0	0	1	0		
	0	1	0	0	
		1	0	0	0

This reflects something you can easily see in the boxes: note f occurs progressively earlier relative to e, which perhaps accounts for a subjective "phase shift" quality in the tune. All this is merely to illustrate a possibility for analyzing a sub-pattern that, through the alignment, generates a new, perhaps more regular pattern. This also illustrates what there is to do with patterns: you can generate more patterns.

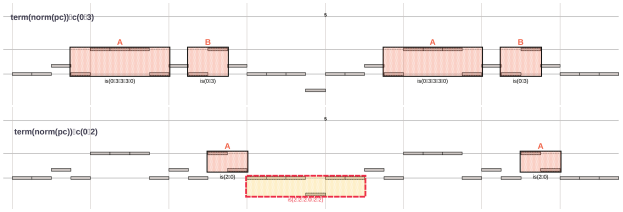

Figure 3: Top: 3rd as micromelody; pattern is $ABAB$. Bottom: the second Pc_Nreveals a *unique* shape at the phrase boundary, created by the low note d; pattern is $A \bullet A$.

Figure 3 shows Pc_{03} and Pc_{02} as micromelodies in the same tune. (bottom). The instances of Pc_{02} involve different sets of notes, as well as showing a pattern break – a unique shape at the phrase boundary, created by the low note d.

Pattern superpositions

The patterns and structures so far represented were taken one at a time, confined to Pc_Ns, representing underlying note sequences. These sequences (which always point back to notes in the input score) can be converted to other parameters: duration, say, or position in measure, constructing a new term vector, which will not necessarily match the first. We refer to such transforms as *parametric pattern maps*. We can then stack these up in a matrix of superpositions, and study the way in which the terms line up.

Fig. 4 shows this for two patterns in a German folktune. The top system shows the Pc_N term pattern, based on pitch. The bottom system uses the same Pc_Nboxes, but generates

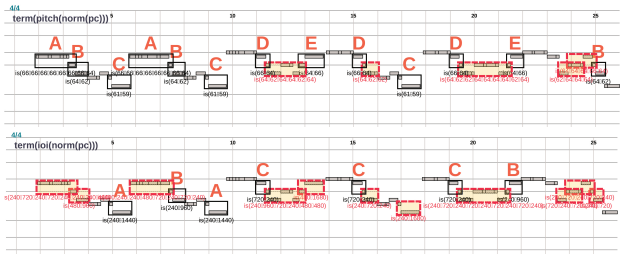

Figure 4: Superpositions of parametric pattern maps in a German folk tune.

another *pattern* based on sets of *ioi*s (inter-onset-intervals). The dotted boxes are non-repeating terms, *hapaxes*.[3]

$$\frac{\text{ABCABCD}\bullet\text{ED}\bullet\text{CD}\bullet\text{E}\bullet\bullet\text{B}}{\bullet\bullet\text{A}\bullet\text{BAC}\bullet\bullet\text{C}\bullet\bullet\text{C}\bullet\text{B}\bullet\bullet\bullet}$$

The correspondence between the two vectors is clear: they line up when pitch and ioi happen to be co-occurrent. Pitch D always lines up with ioi C: pitch C is a twice co-occurrent with ioi A, and once not, where the ioi is longer than before. Co-occurrences might be taken as more *salient* that those where pitch and ioi are independent: the pattern can be "modeled" as *stronger*. We moreover possess information about *where* and *how* patterns are broken, a situation certainly advantageous to the modeling of dynamic expectations (a topic outside our current scope). Here the question is how to use this pattern as a supplemental structure for articulation. If we were to construct a linear map of just the top vector, then each term would be transformed in exactly the same way. The bottom vector informs us of a *subpattern* in pitch C, namely $AA\bullet$.

$$\frac{\text{CCC}}{\text{AA}\bullet}$$

The first two pitch C's could be articulated in the same way, with an option to do something different to the final C: it could be louder, for example. Semantically, C is the long low note of a phrase ending: we have captured a detail of a musical convention, how it repeats and is varied.[4]

The correspondences between the two vectors are nevertheless not perspicuous: some way of formalizing their relation is needed. To this end consider first-order logic's definition of the conditional: $A \implies B$ is true except when A is true and B is false. This exactly captures the property of term pattern inclusion: the top includes the bottom except where there there is something on top and $\bullet$ below. The property of inclusion—suggesting the traditional picture of structural hierarchies–might be linked to perceptual salience: so the analysis of inclusions might help in trying to model perceptual process. But this is not yet exactly what we have.

[3]The term *hapax*, short for *hapaxlegomenon*, is borrowed from linguistics, where it refers to a uniquely occurring word.

[4]"Semantics" here refers to the interpretation of (formally defined) structure in the conventional categories of music theory.

Pattern subpositions

The general problem is that a score can generate *many* term vectors and we would like to study their inclusion relations. At the same time, we would like to *test* the idea that inclusion patterns *are* in fact salient, and we propose to investigate this by the construction of articulations. The claim is that while these articulations are really sonifications of structure (as understood here) we hope to find an overlap with a subjective evaluation of musicality.

$$\begin{array}{c} \bullet AA \\ \hline B \bullet B \\ \hline CC \bullet \end{array}$$

In the pattern above we cannot construct an inclusion pattern without losing information: nothing is entirely included in something else. (The poset does not have single *top* node.) The best we can do is to take the patterns one at a time, together with anything that they include. The A pattern would now look like this:

$$\begin{array}{c} AA \\ \hline \bullet B \\ \hline C \bullet \end{array}$$

That is to say, we take the *point of view* of one pattern and allow all inclusions from other patterns to filter through. A typical set of inclusions is shown in Figure 5: a top pattern generates the columns, and each row represents the inclusion pattern of a parametric transform as seen through the lens of the top pattern.

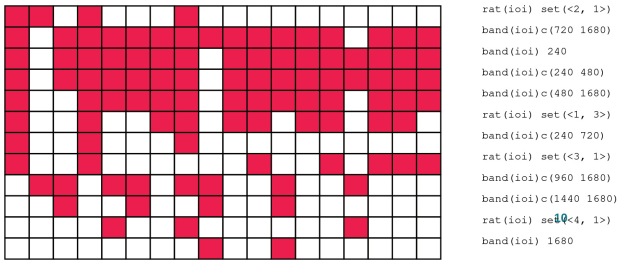

Figure 5: Superpositions of parametric pattern transforms

Call this the *subposition* matrix, in which the property is that we are *allowed* to ignore terms that don't fit the inclusion perspective of the top pattern. We have two problems: to discover how to use a subposition matrix for articulations, and to assemble a picture that uses *all* subpositions, or some interesting set of subpositions such as a *cover*, where everything makes some contribution – the totality of structure captured speaks at once.

Artik Model

We now briefly describe the articulation algorithm, *Artik*.[5]

Among the problems of articulation are those that can be meaningfully reduced to a yes/no decision: should we construct a legato transition between these two notes? Should we take the full value of this note or shorten it? We can answer these kinds of questions with a *bit vector*. Now consider what such a bit vector should look like, say for a Bach sonata: some parts will be mostly legato, others mostly staccato, and others will have close mixtures. A uniformly alternating vector will not do, nor will uniform randomness help, since differentiated higher-level groupings will be lacking. The bit vector can in fact have the complexity of music, since 1 = drum tap and 0 = no tap is sufficient to produce drum solos with sections, transitions, periods of growing complexity, etc. This observation shows that the *temporal*, unfolding structure of music *can* be (partially but usefully) encoded in a bit vector. We are in essence trying to create a 2D precis of a high-dimensional object in such a way that aspects of structure from the more complex object somehow find their way into the binary structure. Now our problem is merely how to generate one, given a subposition matrix. How exactly do we regress pattern dimensionality?

Magic model for dynamics

One idea is to use *magic*. In a first reduction experiment, we wanted to create a *union sum* for all subposition matrices so as to use all matrices, and tried to do so merely by *counting* how many times each note appeared in a subposition in the set of all matrices. The velocity of each note was then set to its count modulo N (where N is the number of available velocities). The guess/intuition behind this was that the the *longest* and most *frequent* patterns would make the greatest contribution, creating a new "summary" structure, and this is just what seems to happen. A typical result (a set of different "interpretations") is shown in Figure 6. Each channel is count vs. time, and the difference between channels arises by varying the count modulus. The periodicities are roughly correlated and certainly project a sense of organization. We can interpret these numbers as midi velocities.

What does this do? The results can be quirky but consistent: related structures are articulated in related ways. In the works investigated, results do not necessarily stand in any relation to prescribed dynamics: sometimes they merely sound wrong. But often the velocities, which generate spontaneous sectional contrasts, echo effects, crescendi, and patterns of accents, seem surprisingly adjusted to the music.

Binary articulation model

Now we return to the problem of constructing a binary representation of a single subposition matrix, attempting another approach. Again, we elect a fairly *magical* approach in the following algorithm. Choose a matrix that spans more or less an entire piece (in general one can be constructed through a suitable choice of structures). Then, for each column, mark the *first note* in each included subgroup. (We alternatively could mark the ends, or both, or some other subset.) The magic is that we cannot foretell what will happen. We applied this to Vivaldi's *Tempesta di Mare*, the opening of the solo violin shown in Figure 7, where red=1, blue=0. Note how the red/blue alternations in mm. 4-7 create a "streaming" (virtual polyphony) effect. We also get a big group, in a sense climactic, in mm. 7-9, and a mixed articulation for the scale-like passages.

[5]The "k" is intended to be suggestive of G. Ligeti's 1958 electronic composition *Artikulation*.

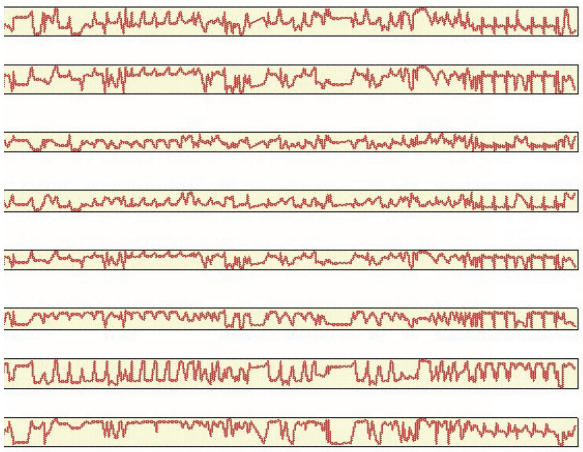

Figure 6: 8 variations on a 2D structural reduction of roughly 1/2 of Scarlatti Sonata K452. Values assigned to each note are used *oracularly* where choices are available. Interpreting values as intensity leads to a quirky yet "musical" performance.

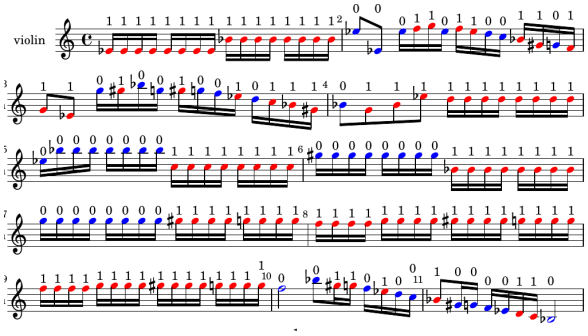

Figure 7: Binary articulation of Vivaldi, Op. 8 No. 5, "Tempesta di Mare."

Articulating polyphony

Given that we have a method for articulating a single voice, the problem is how to *migrate* articulations to other voices. In the 3rd *Brandenburg*, scored for 3 violins, 3 viole, and 3 celli with contrabass continuo, virtually all combinations of unison/octave playing arise. And obviously it is interesting for the unisons to be articulated in the same way, as they are in orchestral string sections, where the bowing is to be in concert.

One solution was simply to force each voice to assume the matrix given by whatever structures a previous voice used, and the articulation vector is otherwise constructed without reference to the concerted effect. For the Brandenburg, the desired concert effect was achieved, with the included benefit that the themes all group into their own articulations. The opening theme, repeated at the end, comes out with identical dynamics and articulation. The many variants of the theme, on the other hand, form subgroups with distinct identities.

Results

Some provisional but suggestive results are presented at www.computingmusic.com. The problem was to transform midi directly into samples that offer articulations and a general sense of performance realism. The midi files used were left uncorrected, with velocity and volume information ignored. Our own software was used to render the samples, so that we were fully in control, modulo the stochastic qualities of the samples themselves.

One obvious difficulty is the narrowness of the dynamic range, a consequence of sample normalization and the difficulty of deciding on appropriate amplitude for different attack styles (pp, ff, etc.). Otherwise the interpretations seem at least half-valid: sometimes they spontaneously follow score indications, as in the Brandenburg, whose opening is f, with contrasting sections, marked p, played in an appropriately sweet legato manner. There are certain stumbles: the sections with pulsing tones and the repeated arpeggiated tune (see Figure 9, mm. 92) invert the usual foregrounding of the tune and backgrounding of the throbbing pulse. But the result is not without charm, creating eerie Bernard Hermann-like suspense. The absence of rules (about how to articulate music) proves to be advantgeous in creating different points of view, expanding the interpretive possibilities. One example occurs in mm. 87-90, a climactic moment in the piece. The algorithm produced an unconventional but musically interesting articulation that adds a new *rhythmic* level, shown in Figure 9, bottom. The fruitfulness of the concept of articulation as the generation of supplemental structures is clear.

Since we distinguish between structure and model, assignments to velocity are *arbitrary*. Hence we can construct alternative models of the articulation vector that include *inverses*, where loud and soft are inverted: since no constraints are placed on what *can* be loud and soft, the inverse model ought to be just as valid. The inverse articulation (of staccato/legato) is also presented on the website. Readers can judge whether the effort was worthwhile by comparing these with a *randomly* articulated Brandenburg, which also has its own merits. In all cases, the same (Artik-generated) dynamic set was used.

But *why* does the algorithm work at all? At present, we don't know. Thus, we meet a criterion of the "Lovelace Test," which takes the lack of an explanation for the causal properties of a generative algorithm to be an indication of "intelligence" (Bringsjord, Bello, and Ferrucci 2001). We can of course continue to experiment with more "rational approaches," through which we might better understand the present algorithm. In the worst case scenario, we must declare the algorithm to be "magic": usable, but theoretically opaque.

The approach bears out our hypothesis that piggybacking on musical structure – even without semantic theorization of that structure and without "knowledge" of human musical practices (e.g. data, heuristics about specific kinds of musical situations) – can provide a basis for *touching* music computationally.

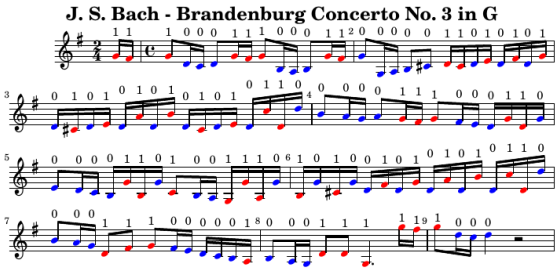

Figure 8: Red/1 might stand for *legato*, blue/0 for *staccato*: legato implies a connection with the next note. The distribution creates many new groupings and the patterns tends to shift with the music: notice, e.g., the red/blue reversal in mm. 6 corresponding to the "bariolage" crossover, reversing the orientation of the interval that is taken legato. A charming effect?

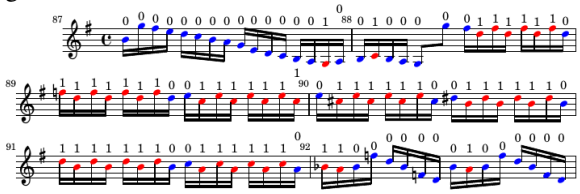

Figure 9: Music at the center of the Brandenburgian labyrinth: the ensemble breaks out into a riotous descending passage with mostly new material. (Only the 1st violin is shown: violins and violas play rhythmically unison, while the bass walks downwards.) The algorithm has produced a fresh "rhythmic" articulation that stands in interesting contrast to the conventions usually observed in performance. In doing so it has created a supplemental structure.

Conclusions

Articulation spaces

There is a tendency in musical AI research to think of a generative program as something that is like a composer who produces individual compositions. This is the legacy of the Turing Test, in which computers must simulate individuals. Programs, however, generate spaces. In order to seem like individuals, methods have to be devised to select "the best," a criterion that may make sense when the game is *Jeopardy*, but not music, where it can only devolve on taste or "likes." With *Artik*, we would have to ask: is there a a vector that captures *as much structure as possible*? The question seems incoherent. In our thinking, the *most* structure is always given by repetition. Obviously this is not what we want.

It seems instead interesting to consider the generated *structural* space as a mathematical and musical object in its own right. Consider the possibility of moving from one point in this space to another: the result would be like turning a knob to get a different interpretation of the same music. The space could be one of *orchestrations*, but also plausibly of variations. The question of which is "best" reduces to an expression of human interest. A user spins a wheel and transforms a pop song into an orchestral fantasy. A researcher might wish to explore a specific subspace by constraining the pattern space. The computational esthetician will want to try correlating structure to esthetic states. An avant-gardist will demand access to the meta-space from which a whole new concept of pattern might emerge. These are the sorts of possibilities that arise when we reject the idea that computers must be individuals, rather than spaces. The results should be programs that are *usable*, not merely postulations (worse: hypostatizations) of "expressiveness."

But the problem does not completely go away. In order to be usable, a music-transforming program must somehow or other stay within the world of music in a sense that is difficult to make precise. A tonal composition that is transformed into an atonal piece has lost a certain kind of significant structure. The "musical sense" of *Happy Birthday* is lost when played backwards (Jackendoff 1994). Can we partition the articulation space so as to favor "music-preserving" structures? Might we, for example, constrain a space so that each of its points is theorized as belonging to a specific corpus—say, the solo violin music of J. S. Bach, or a certain flavor of EDM? The problem might be cast entirely structurally, in which case progress might be made.

But at any rate it seems likely that cultural identity of music will shift from the older emphasis on individual "masterworks" to the constructive space of possibilities. This follows from the way technology is likely to influence the practice of music, through increased participation and openness in everything that pertains to the knowledge and practice of music. In this kind of world works created are likely to forfeit the individuation of music in structurally frozen things, instead taking on a presence as a world, an environment: Eco's theorization of the "open work" (Eco 1989) – itself derived from Boulez and others – is applicable. Music might come to be understood as a kind of living space inhabited, rather than consumed, in constant flux, making good Wagner's precept of "music as the art of transition."

Structure and Semantics

Making music is widely conceived as creating esthetics, beauty and meanings, and this is what AI must somehow grasp: research presently faces a semantic barrier. The question is: can we capture experience – musical meaning – through structure? We put the question inversely: can we *generate* meaning through structure? Supposing we *can* (however you regard our results). In that case we might say: the semantics *is* that particular structure. But then we would want to know how that structure generalizes, while preserving whatever aspects were relevant for human semantics. It might be possible to come up in this way with a descriptive space of musical semantics, coded as structure, rather than as non-computational labels. It is conceivable that *musical* structure, standing in unique relation to semantics, might offer basic clues to the general problem of coding human experience: music as a foundational science of semantics.

References

Bringsjord, S.; Bello, P.; and Ferrucci, D. 2001. Creativity, the turing test, and the (better) lovelace test. *Minds Mach.* 11(1):3–27.

Cambouropoulos, E. 2001. The local boundary detection model (lbdm) and its application in the study of expressive timing. In *International Computer Music Conference*.

Eco, U. 1989. *The Open Work*. Harvard University Press.

Friberg, A.; Bresin, R.; and Sundberg, J. 2006. Overview of the KTH rule system for musical performance. *Advances in Cognitive Psychology, Special Issue on Music Performance* 2(2-3):145–161.

Handelman, E., and Sigler, A. 2013. Key induction and key mapping using pitch-class set assertions. In *International Conference on Mathematics and Computation in Music*. Springer.

Handelman, E.; Sigler, A.; and Donna, D. 2012. Automatic orchestration for automatic composition. In *International Workshop on Musical Metacreation*, 43–48. AAAI Press.

Jackendoff, R. S. 1994. *Patterns in the Mind: Language and Human Nature*. Basic Books.

Kim, T. H.; Fukayama, S.; Nishimoto, T.; and Sagayama, S. 2012. Statistical approach to automatic expressive rendition of polyphonic piano music. In Kirke, A., and Miranda, E., eds., *Guide to Computing for Expressive Music Performance*. Springer.

Kirke, A., and Miranda, E. 2012. *Guide to Computing for Expressive Music Performance*. Springer.

Pearce, M.; Müllensiefen, D.; and Wiggins, G. A. 2008. An information-dynamic model of melodic segmentation. In *International Workshop on Machine Learning and Music*.

Suzuki, T.; Tokunaga, T.; and Tanaka, H. 1999. A case based approach to the generation of musical expression. In *In Proc. of IJCAI*, 642–648.

Todd, N. 1995. The kinematics of musical expression. *The Journal of the Acoustical Society of America* 97(3):1940–1949.

Wagner, R. 1983. *My Life*. Cambridge University Press.

Widmer, G. 2001. Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research* 31:37–50.