Controlled music style mixing improvisation

Rafael Valle

Center For New Music And Audio Technologies
University of California, Berkeley

Abstract. This paper introduces the concept of controlled harmonic and rhythmic music style mixing and improvisation, the process of generating a random sequence of control events guided by a reference musical sequence and satisfying musical constraints. This paper proposes an empirical solution applied to the domain of choir music. More specifically, this paper considers the real-time generation of a mixed style choir piece given two choir pieces. Some improvements on previous music style mixing models applied to music are presented, including multivariate distance and similarity metrics based on high-level features. This approach can be decomposed into two phases that run in real-time : a generalization phase, that learns from a training sequence (e.g., obtained from a sound file or live-input) an automaton, and a validation phase that uses a Levenshtein automaton to enforce a specification on the sequence being generated. The supervision uses a measure adapted from the Levenshtein Distance to estimate the divergence between states and employs strategies to bound this divergence. An empirical evaluation is presented on a sample consisting of choir music from Gesualdo and Debussy.

1 Introduction

Machine improvisation and related style simulation problems usually consider building representations of time-series, such as music and the Down Jones Industrial Average, either by explicit coding of rules [Experiments in Musical Intelligence] or applying machine learning methods [OMax]. Music stylistic mixing can be defined as the process of applying such methods to musical sequences in order to capture salient musical features, organize these features into a model, estimate the divergence betweens the models, and generate new musical sequences based on the mixture of these models. The controlled music style mixing improvisation process browses the models for events within the specifications to generate variant musical sequences that represent a stylistically recombination consistent with the learned material.

Like machine improvisation systems, the learning phase and the improvisation phase happen concomitantly and in in real-time, thus creating a continuous dialogue between source of input and machine output. Context-inference modeling as applied to musical sequences has been experimented since the very beginnings of computer music. This is supported by the fact that humans are able to learn the syntax and grammar of a musical piece and, therefore, predict an event from the sequence of preceding events. These models provide the conditional probability distribution over an alphabet [1] given a preceding sequence called a context. This distribution is used for predicting the next symbol within the given context

2 Control Improvisation

This section formally defines the Controlled Music Style Mixing Improvisation problem. In this work, only the symbolic aspect of music is considered, leaving aside the problem of audio transcription. Currently, this work deals with a mixed symbolic music notation based on discrete sets, namely a discrete set of pitches with quarter-tone resolution a discrete set of durations with 62.5 ms resolution.[2] Borrowing from the undergoing research developed by the music and machine learning group at CNMAT, the formal background can be set up in terms of finite state automata.

2.1 Notation and Background

Definition 1. *A finite state automaton (FSA) is a tuple* $\mathcal{A} = (Q, q_0, F, \Sigma, \rightarrow)$ *where* Q *is a set of states,* $q_0 \in Q$ *is the initial state,* $F \subset Q$ *is the set of accepting states,* Σ *is a finite set called the alphabet and* $\rightarrow \subset Q \times \Sigma \cup \{\varepsilon\} \times Q$ *is a transition relation with infix notation* $q \xrightarrow{\sigma} q'$ *to mean that* $(q, \sigma, q') \in \rightarrow$*, and* ε *is the empty word.*

[1] Letters of the alphabet represented as musical events, for example

[2] These resolutions represent the limits of human performability

Letters of the alphabet are interpreted as observable events of the system under consideration. A word w is either empty (ε) or is a finite sequence of letters in the alphabet Σ, i.e. $w = \sigma_1 \sigma_2 \ . \ . \ . \ \sigma_j$ for some integer $j \geq 1$. The length of a word is defined inductively as $|\varepsilon| = 0$ and $|w\sigma| = |w| + 1 \ \forall \ \sigma \in \Sigma$. A word is a trace of a FSA $\mathscr{A}$ if and only if there exists a $\sigma_1 \ \sigma_2 \ . \ . \ . \ \sigma_{n-1} \ \sigma_n$ sequence of states $q_i \in Q$ such that $q_0 \xrightarrow{\sigma 1} q_1 \xrightarrow{\sigma 2} . \ . \ . \xrightarrow{\sigma n-1} q_{n-1} \xrightarrow{\sigma n} q_n$. It is an accepting trace of $\mathscr{A}$ if and only if q_n is in F . The language of $\mathscr{A}$, noted $\mathscr{L}(\mathscr{A})$ is the set of accepting traces of $\mathscr{A}$.

2.2 Problem Definition

Controlled music style mixing improvisation aims at randomly recombining traces among a family of traces within the specifications which are equivalent based on some similarity measure. One possible approach iteratively chooses a model m_i randomly or with probability Pm_i as the probability that the specification automaton chooses m_i, and on every iteration selects traces within the specifications. Some variations of this approach exist in musical style mixing algorithms, and most of them excel in generating musical results that quote fragments from some input data, as illustrated in Figure 1, a fragment of Memex by Shlomo Dubnov, which consists of a recombination of excerpts written by several composers from the baroque and early classical periods. Memex achieves style mixing by performing analysis and improvisation on musical sequences derived from the corpus of western music. The analysis generates a Factor Oracle through which a random walk takes place according to constraints related to constraints such as the suffix law and the max continuation.

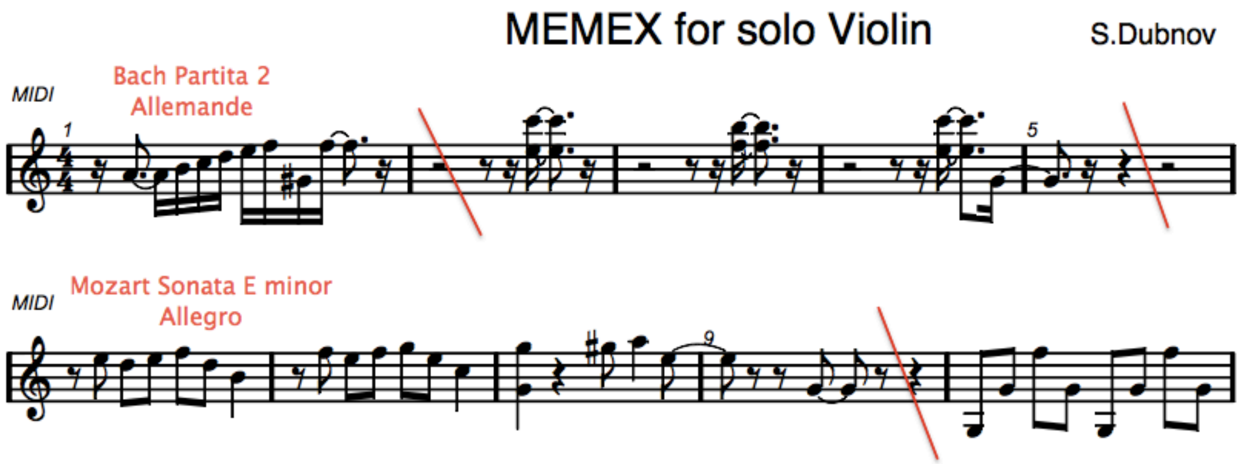

Figure 1. The first 10 measures of Memex by Shlomo Dubnov. Annotations show identify excerpts derived from the training data used in the piece, which included, among other pieces, Bach's Partita N. 2 in D. minor BWV 1004 and Mozart's E minor violin sonata K. 504.

It is clear, however, that quoting fragments of a composer or composers does not mean modeling her or their style, nor does quoting fragments from multiple pieces represent music style mixing. The main problem is to find a similarity metrics that addresses low and high-level musical features. At the current stage, this research assumes that musical similarity can be computed by by a non-negative function d_{wref} on words, such that $d_{wref}(w_{\mathrm{ref}}) = 0$ and $d_{wref}(w)$ increases as w becomes more divergent from w_{ref}.

The control style mixing improvisation problem is then defined with respect to $\mathrm{w}_{\mathrm{ref}}$ and d_{wref} in addition to the plant [3] $\mathcal{A}^{\mathrm{p}}$ and specification $\mathcal{A}^{\mathrm{s}}$. A controller solving the controlled music style mixing improvisation problem resolves the non-determinism in $\mathcal{A}^{\mathrm{p}}$ in two ways:

1. If several transitions of $\mathcal{A}^{\mathrm{p}}$ are within the constraints specified by $\mathcal{A}^{\mathrm{s}}$, one is picked following a likelihood threshold [0,1] to ensure the conditional probability distribution of events;

---

[3] the system being controlled, for example a musical sequence or event

2. When no transition within the constrains is found, to prevent blocking while still preserving similarity, one transition to a similar word w_{sim} from $\mathcal{A}^{s}$ is chosen, and new transitions are searched within $\mathcal{A}^{p}$ again.

As expected, if the controller fails completely in finding in $\mathcal{A}^{p}$ any transition specified by $\mathcal{A}^{s}$, all states will be selected from $\mathcal{A}^{s}$ because the transitions in $\mathcal{A}^{p}$ do not fulfill the specifications in A^s. It is possible, however, to modify some transition in $\mathcal{A}^{p}$ to generate a new symbol $\sigma_{modified}$ that, albeit not present, fulfills some of the specifications of $\mathcal{A}^{s}$. Naturally, this requires that the process generates accepting words of a minimal length n within the specifications. This, however, is beyond the scope of this paper.

3 Music Seeds : Control Music Style Mixing Improvisation

In this section, we describe the components of Music Seeds and use it to apply control improvisation to music style mixing.

3.1 Musical Specifications

The specification FSA encodes constraints involving musical events[4] which enforce some general structure and basic musical consistency. This calls for the definition of some terms used within the context of this paper :

Definition 2. The musical component is defined as any item from the set {amplitude, frequency, duration, relative onset time} that is used to articulate a musical event.

Definition 3. The musical event is the insertion or removal of a process into the sonic space, e.g. note, chord, motif. Its smallest unit is the four-tuplet {Amplitude, Frequency, Duration, Relative onset time}

---

[4] Represented by a symbol in the alphabet

Albeit several articles present statistical models for calculating similarities between musical events and sequences of musical events, to the knowledge of this author there is no machine improvisation or music style mixing engine that embeds music perception and cognition into their metrics. We describe a system to calculate the musical distance between musical events or a sequence of musical events in a set of states Q or between events of sets of events from different sources. In this paper, we realize experiments with excerpts of *Luci serene i chiare* written by Carlo Gesualdo and *Yver, vous n'estes qu'un vilain* composed by Claude Debussy. For the purpose of this paper and within the characteristics of these two pieces, harmonic, rhythmic and persistence similarity are the main metrics for calculating musical distance.

Definition 4. The musical distance is defined as the weighted average of the distances of n components of at least two musical events σ_i and σ_j with length n and $m \in \Sigma \cup \{\varepsilon\}$, *with n and m* ≥ 0.

Since the Levenshtein distance is given by counting the minimum number of operations needed to transform one word into the other, where an operation is defined as an insertion, deletion, or substitution of a single character, it is possible to musical events of different lengths, such as rhythms with different lengths and number of onsets or chords with different number of notes. This implies that, given the correct metrics, the Leveshtein distance can be an efficient measure of the distance between musical phrases, motifs and high-level structures.

3.2 Harmonic Distance

There exists different metrics for calculating the harmonic distance between vertical pitch structures. There is a myriad of strategies that include the use of periodicity transforms, set theory, music cognition. The method used in this paper combines the Levenshtein distance with music perception and cognition. We consider a vertical pitch structure as a word comprised of some letters (pitches). The analysis of the Levenshtein distance between vertical pitch structures happens as follows :

- *modword1* and *modword2* are generated by computing the *modulo n* (n equal to desired resolution), of every letter in *word1* and *word2* respectively.[5]
- $interval_{word1}$ and $\mathrm{interval}_{\mathrm{word2}}$ are generated by computing the sequential music intervals, L_i- L_{i-1}, with $0 < i < n$, of $\mathrm{mod}_{\mathrm{word1}}$ and $\mathrm{mod}_{\mathrm{word2}}$ respectively.
- the values in $\mathrm{mod}_{\mathrm{word1}}$, $\mathrm{mod}_{\mathrm{word2}}$, $\mathrm{interval}_{\mathrm{word1}}$, and $\mathrm{interval}_{\mathrm{word2}}$ are converted to characters to comply with Levenshtein requirements.
- the Levenshtein distance between $\mathrm{interval}_{\mathrm{words}}$ and $\mathrm{mod}_{\mathrm{words}}$ is computed, weighted by the number of characters in each word and their maximum distance, which is 11 in this case, and summed.

After the analysis of the Levenshtein distance, the Harmonicity distance between the vertical pitch structures is computed. This process is based on neuronal periodicity and calculates the consonance of each individual *vps* as the sum of the consonances within the range [0,1] of every pair of notes *i*, *j* with *i != j*, in the *vps*, divided by total number of notes. Although the neuronal periodicity model is based on the generalized coincidence function, which is continuous, for these calculations one can limit the note resolution to quarter or eighths of tones[6] and use the discrete coincidence function within the range [1,2].

If the ratio is beyond these bonds, one recursively scales the target value by multiplying or dividing it by multiples of 2 until the ratio is within the prescribed range. Then we weight the consonance of this ratio by 1/2^n, where n is the number of divisions by two until the values were in the range [1,2]. The harmonicity distance between *vps* X and Y is computed as the euclidean distance between the harmonicities of each chord plus 1 - the consonance between the roots of the target and the base vps. Finally, the global distance between vps X and Y is computed as the sum of their Harmonicity and Levenshtein distances. Since our method combines the Levenshtein distance with the discrete coincidence function, it is able to compare *vertical pitch structures* with different number of notes.

---

[5] Vertical Pitch Structures are stored in lexicographic order

[6] According to music performance constraints

3.4 Rhythmic Distance

There are also metrics for calculating the distance between individual note durations as well as sequence of note durations (rhythms), such as gaussian mixture models and periodicity transforms. Our method is again based on the neuronal periodicity and computes the similarity between two durations by scaling their ratio to the range [1,2], by scales target value by multiplying or dividing it by multiples of 2, and then weighting their similarity by 1/2^n, with *n* equal to the number of operations by 2 that were realized until the ratio was within the expected limits. For sequences of durations, we first calculate the derivative of each sequence and then compute it's periodicity as the sum of similarities of each interval divided by the number of intervals used. In a rhythmic sequence, all durations are independent and equally distributed. We compute the expected value X, duration, as the sum of random variables X_0, X_1, ... X_n, where X_i is equal to the duration of the i-th note, D_i. The expected value is calculated as the weighted sum of durations divided by the total number of durations. The similarity between rhythmic sequences is then calculated as the euclidean distance between the consonances of each rhythmic sequence plus 1 - the similarity between the expected values of each rhythmic sequence.

3.3 Persistence of musical event

A variation of a probabilistic suffix tree is used to estimate the persistence of a musical event or a transition between musical events within a certain context. For every feature vector, the music seeds computes the probability of a q_i as the ratio between q_i duration weighted by sum of all durations $q_1, q_2 \ldots q_n$. The transition probabilities from state q_i to state q_j, with q_i and $q_j \in Q$, is calculated as the total number of transitions from state q_i to state q_j weighted by their duration divided by all possible transition from state q_i weighted by their duration.

This strategy was design to calculate, for example, the probability of sequence S_a (G, G, G, Eb) given a certain context and the probability of the transition (G, Eb) given a certain context. In musical parlance, the probability of S_a given the exposition of the first movement of Beethoven's fifth symphony and the probability of the transition (G, Eb) giving it is the main motive of Beethoven's Fifth Symphony.

4 Experiments with Music Seeds and Controlled Music Style Mixing Improvisation

Two experiments were designed to address the problem of interpolating between two musical styles in music generated automatically. Two choir pieces, one from Debussy and another from Gesualdo were chosen and submitted to Music Seeds described. The first experiment was designed to address Music Seeds' ability to improvise within given a specification. Both pieces were submitted to the Music Seeds algorithm, with persistence probability limits set to [0,1] and no rhythmic specifications. The results of Music Seeds's improvisation are described Figure 2 and Figure 3.

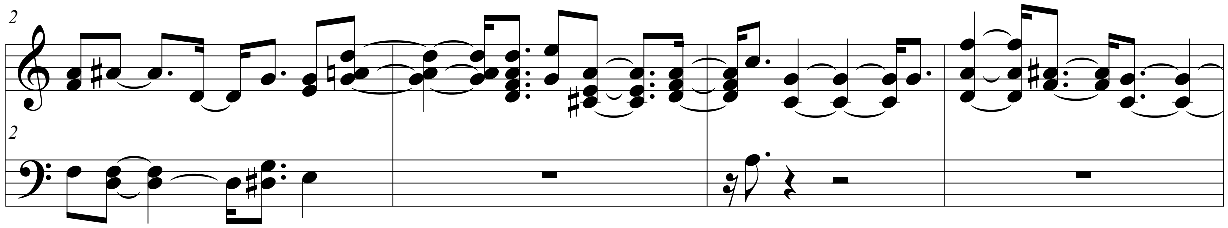

Figure 2 Music Seeds improvisation on Debussy's *Yver, vous n'estes qu'un vilain*

Figure 3 Music Seeds' improvisation on Gesualdo's *Luci serena e chiara*

The second experiment mixes the styles of both composers by using a mixed system (symbolic and subsymbolic) which applies the specification automaton to the sets of states Q_D and Q_G, selecting a state from each one with probability p. It interpolates between each set them by first calculating the musical distance between the reference state and possible excerpts, separating those whose music distance is within the thresholds set up by the user, and then selecting one of them according to the priority sets by the persistence model. Figure 3 shows a realization with harmonic distance threshold set to 14 and equal probability of selecting Q_D and Q_G. No rhythmic specifications were applied.

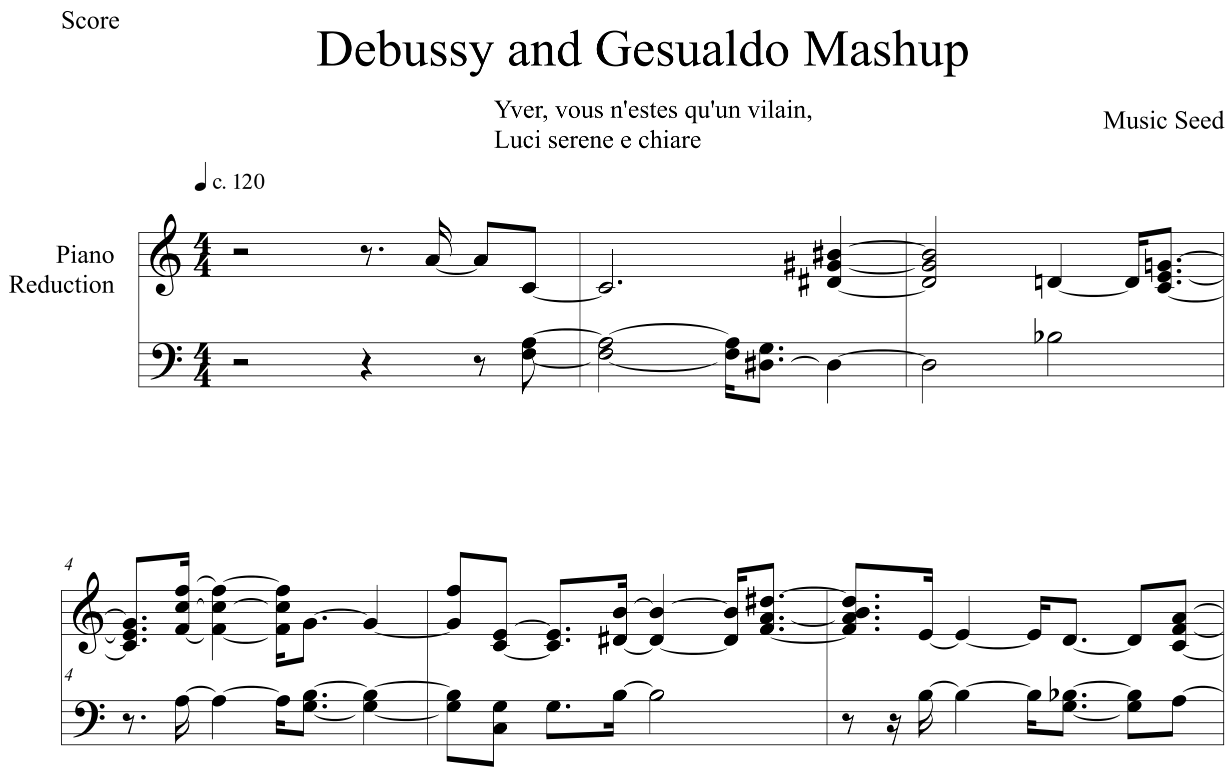

Figure 4 Notice that harmonic chords disposition that are used in Debussy's piece are mixed with chord progressions from Gesualdo's piece. Furthermore, chords used are closely related to both pieces.

7 Related Work

To our knowledge, a concept like controlled music style mixing improvisation has not been introduced in the literature before. We shortly described two approaches to automatic music improvisation: rule-based and data-driven. Recently, musical machine improvisers tend towards data-driven or "predictive" approaches that employ machine learning methods.

This work's approach extends this state of the art by providing a way to (i) enforce musical constraints in musical style mixing improvisations, (ii) creating a measure of musical similarity.

8 Conclusion

This paper presented the concept of controlled music style mixing improvisation and described a strategy to solve it. The strategy presented in this paper provides an efficient solution to harmonic and rhythmic music style mixing and improvisation. This preliminary experiments pave the way to for future research in a generalized controlled music style mixing improvisation in which other distance metrics such as form and motif are taken into account. It was shown that the further uses of the Levenshtein distance can develop an efficient similarity measure between high-level music structures. In addition, this system can be expanded to be used as a tool for learning improvisation by creating, on the spot, harmonic accompaniments according to the student's level and in different styles. Furthermore, the musical application of controlled improvisation with music style mixing, presented in this paper can be generalized and applied to different domains such as tourism, behavioral studies, home automation and finance.

9 References

[1] Allauzen, C. and Crochemore, M. and M. Raffinot, “Factor oracle: A new structure for pattern matching”, In Conference on Current Trends in Theory and Practice of Informatics, 1999.

[2] Argamon, S. and Burns, K. and Dubnov, V. editors “The Structure of Style”, *Springer-Verlag,* USA, 1998.

[3] Assayag, G. and Bloch, G. “Navigating the Oracle : an heuristic approach”, *International Computer Music Conference,* Denmark, 2007.

[4] Bidelman, Gavin. “The role of the auditory brainstem in processing musically relevant pitch”, *Frontiers in Psychology,* USA, 2013.

[5] Conklin, D. “Music Generation from Statistical Models”, Symposium on Artificial Intelligence and Creativity in the Arts and Sciences, Wales, 2003.

[6] Cope, David. “Computer synthesis of musical style”, *MIT Press,* USA, 2004.

[7] Donze, A. and Libkind, S. and Seshia, S. A. and Wessel, D. “Control Improvisation with Application to Music”, EECS Department, University of California, Berkeley, 2013.

[8] Dubnov, Shlomo. and Assayag, Gérard. Cont, Arshia “Audio Oracle: A new algorithm for fast learning of audio structures”, *International Computer Music Conference,* Denmark, 2007.

[9] Dubnov, Shlomo. and Assayag, Gérard. “Memex and Composer Duets: computer-aided composition using style mixing”, *OM Composer's Book 2,* France, 2008.

[10] Rueda, C. and Assayag, G. and Dubnov, S. “A Concurrent Constraints Factor Oracle Model for Music Improvisation”, IRCAM, France, 2006.

[11] Valle, Rafael. “Gradual Control of Harmonicity in the Context of Frequency Modulation”, *International Computer Music Conference,* Australia, 2013.

Score

Debussy and Gesualdo Mashup

Yver, vous n'estes qu'un vilain,
Luci serene e chiare

Music Seed


13
13
16
16
19
19
22
22
25
25