A Novelty Search and Power-Law-Based Genetic Algorithm for Exploring Harmonic Spaces in J.S. Bach Chorales

Bill Manaris[1], David Johnson[1], and Yiorgos Vassilandonakis[2]

[1] Computer Science Department, College of Charleston,
66 George Street, Charleston, SC 29424, USA
manarisb@cofc.edu, dsjohnso1@g.cofc.edu
[2] Music Department, College of Charleston,
66 George Street, Charleston, SC 29424, USA
vassilandonakisy@cofc.edu

Abstract. We present a novel, real-time system, called Harmonic Navigator, for exploring the harmonic space in J.S. Bach Chorales. This corpus-based environment explores trajectories through harmonic space. It supports visual exploration and navigation of harmonic transition probabilities through interactive gesture control. These probabilities are computed from musical corpora (in MIDI format). Herein we utilize the 371 J.S. Bach Chorales of the Riemenschneider edition. Our system utilizes a hybrid novelty search approach combined with power-law metrics for evaluating fitness of individuals, as a search termination criterion. We explore how novelty search can aid in the discovery of new harmonic progressions through this space as represented by a Markov model capturing probabilities of transitions between harmonies. Our results demonstrate that the 371 Bach Chorale harmonic space is rich with novel aesthetic possibilities, possibilities that the grand master himself never realized.

Keywords: Novelty Search, Genetic Algorithm, Markov Model, Harmony, Generative Music.

1 Introduction

Harmonic Navigator (the Navigator, for short) is the latest system from a multi-year interdisciplinary effort exploring artificial intelligence techniques in analysis, composition, and performance of musical works. It is an interactive platform for exploring the harmonic space of distinct or composite musical styles (controlled by the musical corpora loaded into the system). It may be used to compose new harmonic sequences, as well as to support real-time performance by dynamically generating music in collaboration with human performers. The target audience for the Navigator includes composers and performers with basic musical training (minimally, first year music theory at the university level).

J. Romero et al. (Eds.): EvoMUSART 2014, LNCS 8601, pp. 95–106, 2014.

The Navigator generates harmonic sequences through interactive user control. It combines a GUI visualization of harmonic possibilities (i.e., how typical it is for certain harmonies to appear next in a given harmonic sequence), and real-time, gesture-based user input (see Fig. 1). The system is initialized using a particular, stylistically appropriate corpus of music, from which the system extracts harmonies and learns harmonic transition probabilities. Herein, we utilize the Riemenschneider collection of 371 J.S. Bach Chorales. We also have access to the Classical Music Archives 14,000+ MIDI corpus, along with a few additional smaller corpora, for further experimentation.

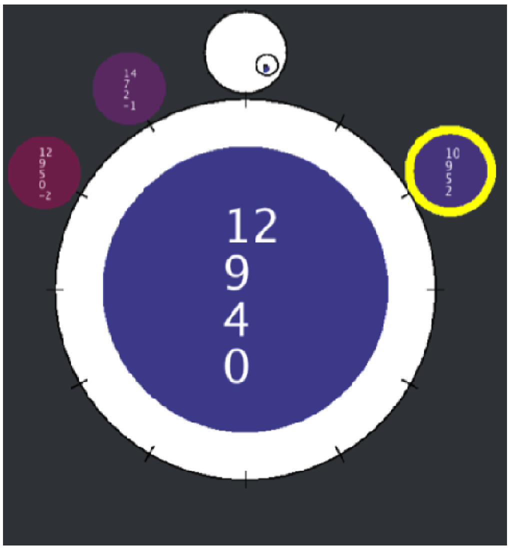

Fig. 1. Harmonic Navigator user interface

The Navigator combines Markov models, genetic algorithms, novelty search, and power-law metrics. In particular, the Markov models are used to learn harmonic transitions. A genetic algorithm (GA) is used to suggest interesting possibilities as the user navigates the harmonic space. The GA is guided using the novelty search approach. Finally, power-law based metrics are used to assess the aesthetic value of generated individuals (harmonic progressions) according to some predetermined target piece (e.g., one of the Bach chorales, or something totally different stylistically, e.g., Led Zeppelin's "Stairway to Heaven"). While the Novelty Search algorithm is encouraging creation of novel individuals, the power-law metrics are used to determine if enough individuals have been generated that meet some basic aesthetic criteria, to be used as the termination condition.

The architecture of the Navigator consists of two main services, a harmonic generator and a gesture engine (see Fig. 2). The gesture engine ensures that an implemented gestural device supports the required user tasks as explained below in the User Interface section. The harmonic generator provides the user interface with all services necessary to generate harmonic flows via the GA and the Markov Model.

The paper is organized as follows: Section 2 discusses background research and related projects, Section 3 focuses on how we extract and represent harmonic data, as well as alternate musical corpora that may be used for training, and Section 4 describes the user interface. It focuses on the visual representation of the harmonic space and the transition possibilities available. The next two sections describe how we combine Markov models and genetic algorithms to search for interesting harmonic sequences. Finally, we present closing remarks and ideas for future work.

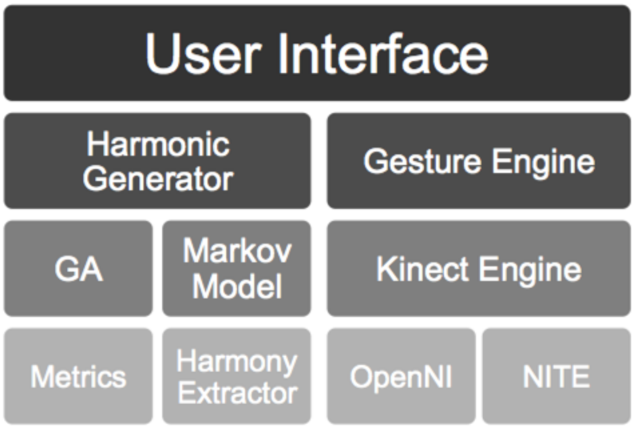

Fig. 2. Harmonic Navigator architecture using a Microsoft Kinect

2 Background

Within the last 50 years there have been numerous applications of computing and artificial intelligence in analysis, generation, composition, and performance of music.

Experiments in Music Intelligence (EMI) is the most comprehensive work in automated computer music generation to-date [1]. EMI performs feature analysis on a corpus of musical works and uses results from this analysis to train Markov models. Using these models, EMI can then automatically compose pieces in a style similar to the corpus, arguably some better than others. EMI works off-line and has been used to generate numerous pieces in the style of various composers.

The *Corpus-Based Harmonic Progressions Generator* [2] mixes stochastic selection via Markov models and user influence to generate harmonic progressions in real time. The user enters information to specify harmonic complexity and tension, as well as a bass-line contour. This is used by the system to influence the selection of harmonies from the trained Markov models, and to generate a harmonic progression.

Continuator is an interactive music performance system which accepts musical input from a human performer. It completes musical material in the same style as the user input [11]. Using a musical corpus, the system trains several Markov models (of various orders), structured in decreasing order of information retrained. Human performer input is matched against the various Markov models starting with the most informed one, and continuing (by reducing the length of the user input) until a match is found. The corresponding Markov model is used to generate a musical continuation. This makes the system sometimes generate perfect reproductions of earlier musical input, and other times less accurate repetitions (introducing interesting variations).

NEvMuse [6] is an experiment in using genetic programming to evolve music pieces based on examples of desirable pieces. *NEvMuse* uses artificial music critics (employing power-law metrics) as fitness functions. The approach was evaluated by training artificial neural networks to predict the popularity of 2000 musical pieces with 90.70% accuracy. The system was used to autonomously "compose" variations of J.S. Bach's Invention #13 in A minor (BWV 784). Similarly to *NevMuse*, the Navigator's genetic algorithm uses power-law metrics to determine fitness.

Monterey Mirror [7] uses Markov models, genetic algorithms and power-law metrics to generate musical phrases in real-time, based on musical input from a human performer. Markov models are used to capture short-term correlations in melodic material. The genetic algorithm is then used to explore the space of probable note combinations, as captured by the Markov model, in search of novel, yet similar melodic material. Similarity is measured using power-law metrics, which capture long-term correlations in melodic material, i.e., the statistical balance between expectation and surprise across various musical parameters [5, 10].

Harmonic Navigator is implemented in Jython and Java using custom GUI, MIDI, and OSC libraries. It incorporates computational elements from *NevMuse* and *Monterey Mirror* to allow human performers to navigate the space of musical harmonies using a gesture-based interface. Different aspects of this system have been presented elsewhere. The corpus-based approach with an emphasis on music analysis, composition, and performance is presented in [8]. The user interface aspects of the system appear in [9]. Herein we focus on the evolutionary aspects on the system.

3 Data Representation

From a music theory standpoint, harmony has played a key role in formal construction and narrative in both tonal and non-tonal systems. Furthermore, harmonic context not only defines musical texture by contextualizing lines (melodies) and creating consonance/dissonance hierarchy, but also, across time, it outlines formal trajectories, controls pacing and phrasing, as well as levels of tension and release. Well-established pitch systems have strong harmonic syntax, which dictates how strongly or loosely vertical sonorities (chords) can be connected in sequence. This syntax in some styles (e.g., common practice functional tonality) is as strong as in natural language, with chords assigned specific functions (much like nouns being subjects or objects in a phrase), hierarchy, weight and even punctuation. Thus, a musical phrase may close with a cadence (a codified chord sequence), "the basis of all Western musical form starting with Gregorian chant" [12]. Within such a pitch system, the listener is guided and can orient themselves by the harmonic flow, which also sets up expectations for what is to follow and how long a specific phrase will be. Of course, this opens the door for introducing surprises, such as deceptive cadences (e.g., V-vi), mode mixture derived harmonies, or modulations to new tonal centers [13].

In this context, the Navigator allows a user who is familiar with the musical style at hand to meaningfully interact with upcoming harmonies, selecting chords that have strong probability of connecting well to the existing sequence of chords (i.e., the harmonic sequence context), or possibly go to unexpected harmonic places. Additionally, we provide a GA process for generating harmonic recommendations based on novelty search and power-law metrics (as described below).

3.1 Musical Corpora

We are currently using the 371 MIDI pieces of the Riemenschneider collection of Bach chorales. Additionally, we have access to the Classical Music Archives (CMA) corpus, which consists of 14,695 classical MIDI encoded pieces. This corpus is augmented with 500+ MIDI pieces from other music genres including Jazz, Rock, Country, and Pop (a total of approximately 15,600 music pieces), for additional experimentation.

Stylistic integrity is paramount when selecting pieces to be used with the Navigator. For instance, combining 12-tone pieces with Renaissance pieces would most likely create a harmonic space with two disjoint subspaces. This is undesirable, if not pointless. On the other hand, combining, say, modal Jazz with impressionist pieces (e.g., Debussy) might create a somewhat coherent harmonic space, which may be interesting to investigate and navigate.

3.2 Harmony Extraction

Harmony extraction is a difficult problem even in MIDI musical corpora. Since our task is statistical in nature (i.e., we calculate probabilities of harmonic transitions), we try to simplify and normalize the information at hand.

For each MIDI piece, we calculate the average note duration across the whole piece and set a duration threshold to remove notes that are too short in duration to be part of structural harmonic voicings (e.g., ornamental pitches such as passing or neighbor tones). This is a parameter that can be set by the user, during training of the system, and is experimentally determined based on the musical style.

Next, we normalize melodic phrases by converting the MIDI pitch data into a standard representation that independent of musical key. To do so, we calculate the key of the piece by creating a histogram of pitch-durations. We assume the most frequent pitch class (total durations) is the key (regardless of octave).[1] Next, from the various instances of the pitch class across octaves (e.g., C3, C4, C5),[2] we select the most frequent one (e.g., C4) and set it as the *base tone* (i.e., canonical pitch 0). All other

[1] Even for pieces with modulations, the pitch class with the longest accumulated duration (across the piece) defines the tonic center of that piece.

[2] C4 represents the pitch C in the 4th octave (i.e., MIDI pitch 60).

pitches (MIDI numbers) in the piece are replaced by the difference between their MIDI pitch number and the MIDI pitch number of the base tone. For example, if a piece begins with C4, D4, B3 (and assuming C4 is the base tone), it is converted to 0, 2, -1.

Then we extract harmonies. For our purposes, a harmonic interval is any interval between two notes that overlap in time.

3.3 Harmony Representation

For each harmonic interval, we store (i_1, i_2, …, i_n), where i represents the interval from the piece's base tone. For example, consider the following two harmonies from a piece with base tone C4:

Harmony 1: (C3, C4, E4, G4, B4)
Harmony 2: (D4, FS4, A4, CS5)[3]

Using the above process, these harmonies would be represented as:

Harmony 1: (-12, 0, 4, 7, 11)
Harmony 2: (2, 6, 9, 13)

where the first number, -12, represents the root of the first harmony (C4), which has -12 distance from the piece's base tone. The other four numbers, (0, 4, 7, 11), are the corresponding pitch distances (intervals) from the piece's base tone. This representation is complete, i.e., it can represent all harmonies (even dissonant ones), and is consistent with music theory. The advantage of this representation is that it defines an aesthetically-relevant notion of harmonic distance across chords. Also, it allows us to define interval-based Zipf metrics for use in the fitness function of our genetic algorithm (discussed later).

4 User Interface Design

The Harmonic Navigator interface presents available harmonies as a dynamic navigable space. It offers two primary modes of interaction: a gesture-based harmonic transition selector, called the *harmonic palette*, and a harmonic-flow scrubber, which presents a global view of a flow being generated. The first UI provides a tree-level view, and thus allows for localized control and inter-harmony navigation (see Fig. 1). The second UI provides a forest-level view, and supports scrubbing and editing actions. Both views use colors to indicate harmonic density (or tension) calculated using Legname's density degree theory [3]. This approach measures the complexity of a

[3] FS4 is the pitch F sharp, 4^{th} octave (i.e., MIDI pitch 66).

harmonic interval's sine wave representation to estimate the consonance/dissonance of the harmonic interval.

Similarly to Monterey Mirror, the genetic algorithm runs continuously (in the background) to suggest interesting harmonic flows. A special bright ring around a follow-up harmony signifies that this is the harmony recommended by the genetic algorithm. Since the genetic algorithm is running continuously, it is possible for the recommended harmony to change (if the genetic algorithm discovers a better choice), while the user is contemplating. As the user navigates forward (or backward) through the harmonic space, the genetic algorithm restarts from this particular point in the space looking for aesthetic recommendations on how the current harmonic flow may be completed. Again, the user has a choice - they may accept the recommendation or ignore it. Either way, the genetic algorithm adapts and continues in parallel with the user's navigation through the space.

The system's user interface is described in more detail elsewhere [9]. Demos of the system's user interface include:

- a demo of using the Harmonic Palette interface via a Microsoft Kinect to control navigation; and
- a demo of the Harmonic Flow interface for scrubbing GA generated harmonic progressions.

These are available at http://www.cs.cofc.edu/~manaris/navigator .

5 Markov Model

Harmonic Navigator uses Markov models to construct an n-dimensional matrix of how often one subsequence of harmonies resolves into a given harmony. Once trained, the system can be used (via the gesture interface) to generate various harmonic sequences (or harmonic flows) that are derived from the training corpus.

It is possible for the system to recreate an exact training sequence (i.e., recreate the harmonies in a musical piece used for training). This is more probable for smaller training sets, as a Markov model would mostly memorize separate (disconnected, independent) sequences. However, as the training set grows and introduces ambiguity (i.e., training sequences overlap across different places), generating exact training sequences becomes exponentially improbable. This is where the power of the Navigator lies, i.e., to facilitate exploration and discovery of novel harmonic flows that are probabilistically plausible (at least, at a local level) and are stylistically consistent with the training corpus. Given our reliance on power-law metrics for fitness, we have observed that Markov orders of 1, 2 or 3 work well with this approach. Higher orders may result in memorization of harmonic sequences, and thus reduce the potential for discovery of novel harmonic ideas.

6 Evolutionary Algorithm

To guide users through the harmonic space, the Navigator utilizes a hybrid Novelty Search and power–law based genetic algorithm. Novelty Search is used to guide the

GA through the space while a power-law based fitness function is used to store the elite individuals and terminate the search.

The size of our corpora generates a large harmonic space with nearly endless possibilities to explore. The fitness function may also incorporate a melodic contour, as in [2] to provide a high-level structure and impose melodic constraints to the exploration. In this case, harmonies are chosen by the GA based on how well they fit the provided contour in addition to the power-law based metrics. Using a melodic contour helps prune the space for more efficient navigation. In cases of smaller corpora, where there may not be enough harmonic material, using a melodic contour may be too restrictive.

It should be noted that these GA-generated flows are not necessarily the most probable ones (i.e., as would be produced by the Markov model). However, the flows are consistent with the Markov model, i.e., they are built only of harmonic transitions found in the training corpus.

6.1 Power-Law Metrics

During the last decade, our group has explored Zipf's law and other power laws in music information retrieval, computational aesthetics, and artificial creativity (see http://sger.cs.cofc.edu). Due to limited space, here we provide an overview of power-law metrics and cite earlier publications.

Power laws are statistical models of proportions found in natural and artificial phenomena. Human language, music and brain waves are a few such phenomena.

Zipf's law is one of many power laws. It corresponds to phenomena (e.g., language in books), where the probability, ***P(f)***, of a certain event occuring (e.g. a word in a book) is proportional to its frequency of occurrence, ***f***, as follows:

$$P(f) = 1/f \tag{1}$$

For instance, if the most frequent word in a book (e.g., "a") appears *x* times (say, ~1000 times), the 2nd most frequent word appears *x/2* times (~500 times); the 3rd most frequent word appears *x/3* times (~333 times); the 4th appears *4/x* times (~250 times); and so on.

We have found that music exhibits power-law proportions across many dimensions, including pitch, duration, harmonic intervals, and distance of repeated notes, among others. Evaluation experiments indicate that power-law proportions (measured across many dimensions per musical piece) correlate with human aesthetics (i.e., aspects of how this piece is perceived by humans) [5, 6, 10]. In particular, we have used power-law metrics to classify musical pieces with high accuracy, in terms of composer, style, and pleasantness, among other possibilities.

For the harmonic navigation task, we have developed three new power-law metrics:

- **Chord Metric** - calculates the power-law proportions of chords across a harmonic flow.

- **Chord Distance Metric** - first, measures the distance of identical chords across a harmonic flow; then, calculates the power-law proportions of those distances.
- **Density Degree Metric** - first, calculates the density degrees of chords across a harmonic flow; then, calculates the power-law proportions of those density degrees.[4]

As stated earlier, our approach combines Markov models and power-law metrics. Markov models capture shorter-term correlations. Power-law metrics capture longer-term correlations, and overall balance in harmonic flow structure. The combination of the two techniques (i.e., Markov models to provide raw genetic material, and power-law metrics for fitness) allows to effectively search the harmonic space for flows (chord progressions) which make aesthetic sense both at the tree level (e.g., voice leading) and at the forest level (harmonic development).

6.2 Genotype Representation

The genetic population is initialized randomly through the Markov model. Each genotype represents a particular harmonic flow (a sequence of chord choices).

Throughout evolution, genotypes always remain consistent with the Markov model (i.e., there is a set of valid transitions in the Markov model which can generate a given genotype). Therefore, the crossover and mutation operations must maintain this consistency. In essence, the GA explores the space of all harmonic transitions captured by the Markov model, in search of the best possible harmonic flows (genotypes).

- The **mutation operator** randomly selects a point in the genotype and replaces the remainder with a partial flow generated from the Markov model.
- The **crossover operator** randomly selects a pivot point (a common harmonic subsequence) in two individuals (parents). Then, it cross-swaps the subsequences before and after the pivot. The length of the pivot subsequence is consistent with the Markov order. (If not such pivot point exists, the parents are deemed incompatible, and they do not produce offspring.)

To decide which genotypes survive, we calculate a novelty value of each genotype (see section 6.3), as well as, a more traditional fitness value (see section 6.4).

6.3 Novelty Search

Novelty search is used to guide the GA through the possible paths of the harmonic space. Rather than using a standard objective-based fitness function, which may constrain search results by moving towards local maxima, novelty search rewards individuals with new behavior. This helps to open up the search and visit places that an objective based search would not have otherwise [4].

---

[4] For more information on constructing such metrics, see [11].

To implement novelty search within an existing GA, the objective based fitness value is replaced by a novelty value. This novelty value is higher for genotypes that are further away from (a) the rest of population, and (b) an archive of previously found novel genotypes. The archive ensures the path of the search does not return to previously visited spaces.

Within our system, the novelty of an individual (harmonic flow) is measured as follows. First, we calculate the behavior of the generated harmonic flow. To do so, we construct a vector consisting of the density degrees of all harmonies in the flow. Next, we average the distance (mean squared error) between the individual's vector and those of the k-nearest neighbors of the population, where k is a constant determined via experimentation. The greater the average distance, the sparser of a region the individual resides in, hence a more novel individual.

If the average distance of the individual meets a specified threshold (also determined via experimentation) it is considered *novel* and is added to the archive. The most novel individuals of each population are used to reproduce the population.

6.4 Melody Guided Fitness

While the novelty metric (described in the previous section) is used to decide which individuals survive during evolution, we use an objective-based fitness measurement to decide, during evolution, which individuals to output to the user. Also, we use this measurement to decide when to terminate the search. So, in essence, we "piggyback" on the novelty search as it explores the harmonic space for novel harmonic flows, and we observe. Any novel harmonic flow that passes the objective-based fitness test, we save.

The objective-based fitness utilizes a target piece, which is provided by the user as input to the Navigator. The target piece is used as an example of balance and aesthetics we would like to emulate. It is possible to provide a set of target pieces. Objective fitness is calculated by comparing the distance (mean squared error, or MSE) between the power-law metrics of an individual and the corresponding metrics of the target.

Next, we determine how well the genotype fits the input melody. This is done as follows:

- we juxtapose the individual to the melodic contour;
- we calculate the density degrees of the harmonies in the individual;
- we add the normalized pitch values from the melody to each corresponding harmony in the individual; and
- we calculate the density degrees of the new harmonies.

The density degrees of the new progression are then compared to the original density degrees by calculating the distance (MSE) between the two vectors. A progression that closely fits the melodic contour will have a small MSE because adding the melody pitch to each harmony of the progression will have a small effect on the harmonic density.

For example, if the melody's pitch is already contained within the harmony then the density degree will not change (the MSE will be zero). But, if the addition of the melody's pitch generates a tense harmony, then the density degree is much higher (a larger MSE).

Adding the weighted MSE values of each measurement generates the final fitness value. As the GA is running (using Novelty Search to reproduce), the fitness value is used **only** to determine which progressions to save. If an individual exceeds a given objective-based fitness threshold, it is saved, while the GA continues searching. The GA terminates after enough individuals have been saved, or after a max number of generations.

Additional results, generated music, and demos of the Navigator's user interface are available at http://www.cs.cofc.edu/~manaris/navigator .

7 Conclusion and Future Work

We have presented the Harmonic Navigator, an interactive system for exploring harmonic spaces of distinct (or composite) musical styles, and for dynamically generating music in collaboration with human performers. The Navigator uses a genetic algorithm to help guide users through the space, while allowing them to have control by providing a target piece and melodic contour. By combining novelty search and objective based fitness, the system explores the harmonic space more thoroughly, while identifying novel harmonic progressions with desirable aesthetics.

Our results indicate that, while the generated harmonic progressions may not be 100% perfect (as this depends on the specific musical corpus used, input melody, and target piece), they may be good enough for music composition and performance tasks. For instance, we have found that the harmonic space of 371 J.S. Bach Chorales, used herein, is rich with novel aesthetic possibilities, possibilities that the grand master himself never realized.

The Harmonic Navigator may be used to explore compositional ideas in harmonic spaces derived from various musical corpora. Additionally, it may be used to revitalize traditional classroom training in tonal harmony via tonal harmony games. Finally, it may be used in musical happenings, together with MIDI and OSC controllers, as well as traditional instruments, to create harmonic contexts for improvised performances.

Acknowledgements. This work has been supported in part by NSF (grants IIS-0736480, IIS-0849499 and IIS-1049554) and a donation from the Classical Music Archives. The authors would like to acknowledge Dana Hughes for correcting MIDI transcription errors in the 371-piece Riemenschneider corpus of Bach chorales. Also, Dana Hughes and Kenneth Hanson for their contributions to the custom Jython libraries used in this project.

References

1. Cope, D.: Virtual Music: Computer Synthesis of Musical Style. MIT Press, Cambridge (2004)
2. Eigenfeldt, A., Pasquier, P.: Realtime Generation of Harmonic Progressions Using Controlled Markov Selection. In: 1st International Conference on Computational Creativity (ICCC-X), pp. 16–25. ACM Press, New York (2010)
3. Legname, O.: Density Degree of Intervals and Chords. 20th Century Music 4(11), 8–14 (1997)
4. Lehman, J., Stanley, K.: Abandoning Objectives: Evolution through the Search for Novelty Alone. Evolutionary Computation Journal 19(2), 189–223 (2011)
5. Manaris, B., Romero, J., Machado, P., Krehbiel, D., Hirzel, T., Pharr, W., Davis, R.B.: Zipf's Law, Music Classification and Aesthetics. Computer Music Journal 29(1), 55–69 (2005)
6. Manaris, B., Roos, P., Machado, P., Krehbiel, D., Pellicoro, L., Romero, J.: A Corpus-Based Hybrid Approach to Music Analysis and Composition. In: 22nd Conference on Artificial Intelligence (AAAI 2007), pp. 839–845. AAAI Press, Palo Alto (2007)
7. Manaris, B., Hughes, D., Vassilandonakis, Y.: Monterey Mirror: Combining Markov Models, Genetic Algorithms, and Power Laws. In: 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011), pp. 33–40. IEEE Press, Piscataway (2011)
8. Manaris, B., Johnson, D., Vassilandonakis, Y.: Harmonic Navigator: A Gesture-Driven, Corpus-Based Approach to Music Analysis, Composition, and Performance. In: 2nd International Workshop on Musical Metacreation (MUME 2013), 9th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, pp. 67–74. AAAI Press, Palo Alto (2013)
9. Johnson, D., Manaris, B., Vassilandonakis, Y.: Harmonic Navigator: An Innovative, Gesture-Driven User Interface for Exploring Harmonic Spaces in Musical Corpora. In: Kurosu, M. (ed.) HCI 2014, Part II. LNCS, vol. 8511, pp. 58–68. Springer, Heidelberg (2014)
10. Manaris, B., Roos, P., Krehbiel, D., Zalonis, T., Armstrong, J.R.: Zipf's Law, Power Laws and Music Aesthetics. In: Li, T., Ogihara, M., Tzanetakis, G. (eds.) Music Data Mining, pp. 169–216. CRC Press, Boca Raton (2011)
11. Pachet, F.: Beyond the Cybernetic Jam Fantasy: The Continuator. IEEE Computer Graphics and Applications 24(1), 31–35 (2004)
12. Rosen, C.: The Classical Style; Haydn, Mozart, Beethoven, p. 26. W.W. Norton & Co., New York (1971)
13. Schoenberg, A.: Structural Functions of Harmony, pp. 192–196. W.W. Norton & Co., New York (1954)