Harmonic Navigator: A Gesture-Driven, Corpus-Based Approach to Music Analysis, Composition, and Performance

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] Department of Music, College of Charleston, 66 George Street, Charleston, SC 29424, USA
vassilandonakisy@cofc.edu

Abstract

We present a novel, real-time system for exploring harmonic spaces of musical styles, to generate music in collaboration with human performers utilizing gesture devices (such as the Kinect) together with MIDI and OSC instruments / controllers. This corpus-based environment incorporates statistical and evolutionary components for exploring potential flows through harmonic spaces, utilizing power-law (Zipf-based) metrics for fitness evaluation. 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 Classical Music Archives 14,000+ MIDI corpus, among others. The user interface supports real-time exploration of the balance between predictability and surprise for musical composition and performance, and may be used in a variety of musical contexts and applications.

Introduction

Visual representation of musical structure is a quest that goes back to the beginning of Western music. By allowing "out-of-time" design, it has defined the development of art music in many ways. It has led to more intricate musical structures and better control of temporal and timbral space. For example, the development of a grid-based music notation system made the preservation and communication of musical structures possible. Moreover, visual representation using geometrical shapes quickly found its way onto theoretical treatises to describe pitch relationships and tuning systems. This, most notably, incudes the circle of fifths and the *Tonnetz*, which is a conceptual lattice diagram describing frequency relationships in just intonation systems, as well as more recent approaches such as Fred Lerdahl's *Tonal Pitch Space*, which employs visual models to explore pitch-space trajectories and compute patterns of tonal tension.

Figure 1. *One of the co-authors "driving" an early prototype of the system.*

Additionally, the universal acceptance of equal temperament tuning has allowed for more abstraction, manipulation and transposition of intervallic relationships. This is because intervals, which measure distance between pitches, can thus be expressed as exact multiples of a basic unit, the semitone. Consequently, organization of the vertical aspect of musical texture lends itself particularly well to visual representation.

Visual representation has traditionally been confined to two-dimensional representation systems. Two notable exceptions are Tod Machover's "Harmonic Driving", an interactive application using a video game interface to explore harmonic space (Paradiso 1999); and Dmitri Tymoczko's "Geometry of Musical Chords" (Tymoczko 2006), which provides a basis to the work presented herein. In this paper, we explore going beyond static (or passive, analytical) geometry to a dynamic interactive space, and thus allow the user to interact with harmony in real time.

The term "harmony" is used throughout this paper to describe vertical sonority, whether it appears in the form of a traditional triadic "chord", as codified in common practice, or any group of pitches sounding simultaneously as a result of independent musical gestures, commonly found in non traditional musical textures.

Harmonic Navigator (Navigator, for sort) 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 harmonic spaces 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 (see figure 1). The target audience for the Navigator includes composers and performers with basic musical training (minimally, first year music theory at the university level). There are several possible user scenarios. One range of possibilities includes using the Navigator to study or analyze well-defined harmonic languages, such as functional tonality, impressionist modal styles, or more abstract harmonic practices. Another range of possibilities includes using the system as a harmonic experimentation tool for composing within a specific harmonic language. This would facilitate seeking novel yet stylistically appropriate harmonies for diverse musical projects.

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. The system is initialized using a particular, stylistically appropriate corpus of music, from which the system extracts harmonies and learns harmonic transition probabilities. Currently, we utilize the Classical Music Archives 14,000+ MIDI corpus, along with a few smaller corpora. The Navigator combines Markov models, genetic algorithms and power-law based metrics for fitness evaluation. 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 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 the musical corpora used for training. Section 4 describes the user interface. It focuses on the visual representation of the harmonic space and the transition possibilities available, as well as the input gesture language used to navigate and select those possibilities. 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.

Background

Within the last 50 years there have been numerous applications of computing and artificial intelligence in analysis, generation, composition, and performance of music. While these results are sometimes judged by how close (or far) they model a human expert (strong AI), the real contribution lies in how they may enhance human creativity and facilitate artistic exploration and expression.

GenJam generates jazz improvisations for real time performance (Biles 2003). GenJam is trained using an interactive genetic algorithm which determines fitness through a human mentor. The trained population is used to "trade fours" with a human performer. While the performer is playing, GenJam listens to his last four measures, maps the measures to individuals in the GA population and then mutates these individuals to generate its own solo.

The *Corpus-Based Harmonic Progressions Generator* (Eigenfeldt and Pasquier 2010) 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.

Experiments in Music Intelligence (EMI) is the most comprehensive work in automated computer music generation to-date (Cope 2004). 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.

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 (Pachet 2004). 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 (Manaris, et al. 2007) 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 (Manaris, et al., 2011a) 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 (e.g., Manaris et al. 2005, 2011b).

Our system's user interface is inspired by Dasher. *Dasher* is an efficient text-entry interface for motor-impaired users, driven by natural continuous pointing gestures (Ward, Blackwell, & MacKay 2000; Ward & MacKay, 2002). The user navigates through a space of probable character sequences using mouse movements to select the next character to "type". As the user selects characters, the display is dynamically updated with new characters (see figure 2). Dasher uses a probabilistic model to determine what characters should appear next, given what has been typed so far. The most probable alternatives are given more screen area to facilitate selection. As predicted by Fitt's law, this approach improves data entry speed, while reducing error rate.

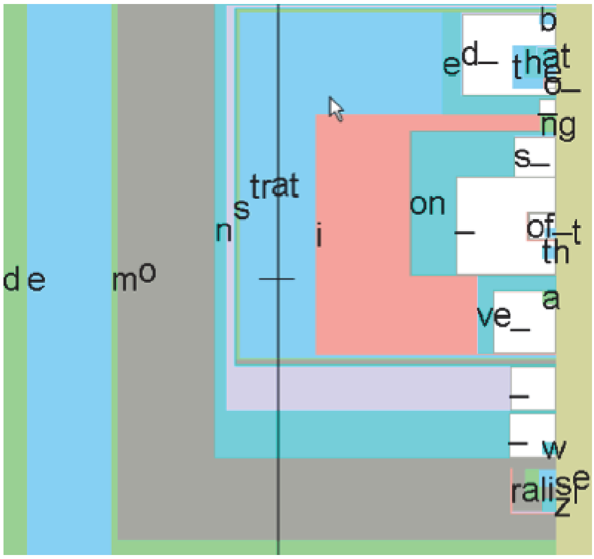

Figure 2. *The Dasher GUI with the user having typed "demonstrat" so far. The most probable sequence is "demonstration". Other alternatives include "demonstrative", and "demonstrated".*

Harmonic Navigator is implemented in Jython and Java using custom GUI and MIDI libraries. It incorporates computational elements from NevMuse and Monterey Mirror to allow human performers to navigate, through a Dasher-inspired interface, the space of musical harmonies using a Kinect gesture interface.

In summary, while the Navigator has been influenced by the other applications described above, it differs in many ways. Most of the GA based applications are interactive, i.e., require humans to judge the fitness of the results. This is time consuming; also, it may produce inconsistent results due to user fatigue. Our system uses power-law metrics to automatically determine fitness. It also takes an interactive approach to give users an environment to express their own creativity in selecting harmonic progressions, while also presenting suggestions from its generative algorithms.

Harmonic Space 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" (Rosen, 1971). 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), or even modulations to new tonal centers (Schoenberg 1954).

In this context, the Navigator allows a user who is familiar with the musical style at hand to meaningfully interact with upcoming harmonies (much like a Dasher user), 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.

Musical Training Corpora

We are currently using 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, as well as 371 MIDI pieces of the Riemenschneider collection of Bach chorales (a total of approximately 15,600 music pieces).

Stylistic integrity is paramount when selecting pieces to be used with our system. 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.

Additional experiments are being conducted with musical corpora in MP3. Our team has developed an automated audio-to-MIDI transcription system, based on the Constant Q Transform algorithm (Brown and Puckette 1992), which is being used in the Armonique musical search engine (Manaris, et al. 2011b). Another system we are experimenting with is WIDI (www.widisoft.com). Although these solutions are not perfect (i.e., they introduce transcription errors and musical "noise", such as extra, "ghost" notes), we are exploring various filtering techniques to extract relevant harmonic data for use with the Navigator.

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 small to be part of harmonic voicings (e.g., 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 perform voice separation using our implementation of the Chew and Wu (2004) algorithm. This algorithm separates a piece into contiguous sequences of melodic phrases, by connecting melodic fragments using a shortest-distance approach. Evaluation experiments indicate a high success rate in identifying individual voices from MIDI material (ranging from 88.98% to 99.75% depending on the particular metric).

Melodic phrases are canonicalized, by converting the MIDI pitch data into a canonical form that disregards key information. 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 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.

Data 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:

[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).

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

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, including (Tymoczko 2006). The advantage of the latter 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).

Gesture Interface

The development of gesture devices (such as the Kinect) has opened new avenues for hands-free, gesture-based user interfaces. Currently, many implementations of gesture languages are based on the point-and-click (mouse) metaphor. However, we need to move beyond point-and-click, in order to utilize the full interaction-design potential of this novel gesture-interface paradigm (Francese, et al. 2012).

User Interface Design

The Navigator's interface presents available harmonies as a dynamic navigable space, similarly to Dasher. While Dasher presents follow-up characters in a 2D left-to-right navigation space, we utilize a 3D front-to-back approach.

The interface presents users with a *harmonic palette*, from which to choose a follow-up harmony (see figure 3). The palette contains a number of circles, each representing a harmony. The current harmony is located in the center of the display. Follow-up harmonies are determined by the current harmony (as dictated by the training corpus), and are placed in a clockwise fashion, around a clock face labeled with the 12 tones. (Since pieces are normalized, C is 0.) Harmonies are represented by vertically stacked numbers stating harmonic intervals. This is consistent with (and provides the same information as) the vertical placement of notes on a staff.[4]

Similarly to Dasher, the size (radii) of follow-up circle-harmonies corresponds to transition probabilities from the current harmony (the larger, the more probable).

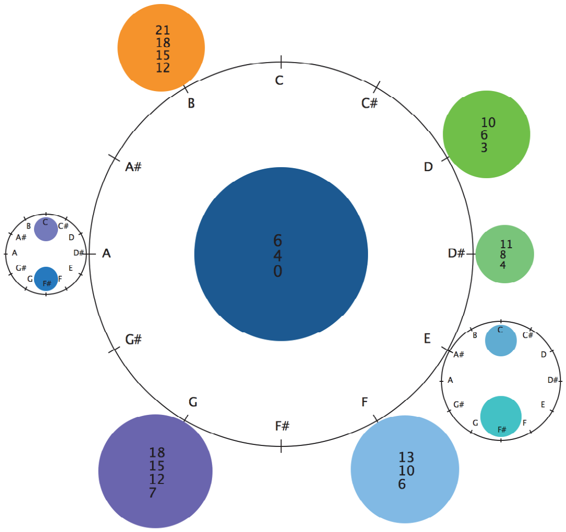

Figure 3. *The Harmonic Navigator user interface (colors adjusted for grayscale printing).*

When there are multiple follow-up harmonies with the same root pitch (e.g., see E and A root pitches, in figure 3), they are arranged around a smaller clock face. The size (radius) of this clock face corresponds to the sum of the enclosed harmonies' probabilities. Hovering the cursor over this clock face zooms in to display a larger version of the clock face, which presents more information about the contained harmonies, and allows the user to select one. The placement of these harmonies inside the smaller clock face is determined by the second pitch in the harmony.

In the case where multiple harmonies have the same second pitch, these harmonies will also be placed inside an even smaller clock. This hierarchical (recursive, fractal-like) grouping continues until all harmonies can be represented individually. Similarly to binary search, this hierarchical decomposition/organization of the harmony palette allows the user to quickly "home in" on a desired harmony. The user efficiency gained by this approach (over a linear, non-hierarchical placement of harmonies around a single clock face) is $O(log_{12}\ n)$, where n is the number of follow-up harmonies (and base 12 given the decomposition of the search space into 12 regions, at each level).

Similarly to Monterey Mirror, a genetic algorithm runs continuously (in the background) to suggest interesting harmonic flows. The follow-up harmony (or harmonies) selected by the genetic algorithm is (are) identified by a special color (e.g., gold) settable by the user. We are exploring other possibilities (e.g., a thicker border).

In accordance with HCI guidelines (e.g., Dix et al. 1998), we use color redundantly, i.e., to emphasize information already present at the interface. Color assignment of circle-harmonies is based on intervallic tension. Intervallic tension is already visible on the interface, through the displayed harmonic intervals

[4] Arguably, in a more direct (usable) manner, given our target users.

(especially for expert musicians). Color assignment makes that information more visible to non-experts.

Intervallic tension of a chord is determined by two factors. One is the intervallic content of the chord - a chord with more tense intervals has a higher tension factor, and thus sounds more dissonant. The relaxation vs. tension of the chord is mapped to cool vs. warm colors on a color wheel (i.e., blues are cool (relaxed) and reds or yellows are warm (tense)).[5]

Once again, it should be noted that this UI was designed for users with, at least, basic training in functional tonality (first-year college harmony, or equivalent). However, as we collect usability feedback from more users, this UI may evolve (e.g., to provide more or, even, less information).

Kinect-Based Gesture Language

We identified three main user tasks that the Navigator's interface needs to support for harmonic navigation. These are "explore the current harmony palette", "select a follow-up harmony", and "backtrack" (i.e., unselect current harmony and return to the previous palette).

Our current prototype is implemented using a Kinect controller. Given this, we have devised the following gesture language to implement the above user tasks. (We are also exploring gesture languages for other controllers, such as the Leap Motion sensor and OSC control via smartphone.) The Kinect gesture language utilizes only one hand via three gestures (freeing the second hand for other activities, such as interacting with MIDI and OSC controllers):

- **Hand Movement in the X-Y Plane** – Moving the hand left-to-right and up-to-down moves the cursor around the display. This action supports exploration of the current harmony palette (e.g., hovering over a secondary clock face to enlarge it).

- **Hand Push** – Pushing towards a follow-up harmony selects it. This moves the selected circle-harmony to the center, begins sounding the corresponding harmony (via MIDI), and displays the next harmony palette. This action supports moving forward in the harmonic space.

- **Hand Wave** – Waving over the current circle-harmony (center of the display), stops sounding it, and returns to the previous harmonic palette (to possibly try something else). This action supports moving backward in the harmonic space.

In particular, moving backwards allows the user to step back to previous harmony selection points, and try other alternatives. While this may seem peculiar during live performance, it may be utilized creatively (not unlike sound looping, and/or "scratching" by DJs). On the other hand, this is quite natural for composition tasks (i.e., "should I use this harmony or that?" or "what harmonic choices would I have here, had I gone to a relative minor three chords ago?").

Statistical Modeling

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 Kinect 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. We have found that, in practice, Markov orders of 1, 2 or 3 work well. Higher orders may result in memorization of harmonic sequences, and thus reduce the potential for discovery of novel harmonic ideas.

Evolutionary Recommendations

The Navigator utilizes a genetic algorithm to discover interesting harmonic choices (and present them), as the user navigates a harmonic space. Aesthetic interest is guided by a fitness function using power-law metrics (see below). The fitness function may also incorporate a melodic contour, as in (Eigenfeldt and Pasquier 2010), to provide a high-level structure and impose melodic constraints to the exploration. In this case, harmonies would be chosen by the GA based on how well they fit the provided contour. This may be useful to prune large harmonic spaces (i.e., spaces generated from large training corpora). However, it may be too restrictive for smaller corpora, where there may not be enough harmonic material to fit the melodic contour.

The GA continuously computes the remaining harmonic flows from the current harmony. Every time the user

[5] At the time of this writing, we are still exploring various color schemes.

makes a new selection (and thus changes the harmonic context), the genetic pool is updated and the GA is restarted. This allows new possibilities to be explored and presented. Harmonies in the GA's elite part of the population are presented at the UI identified by a special color (as mentioned earlier).

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). Instead, they are evolved and evaluated for fitness using power-law metrics (and, potentially, a melodic contour). However, these flows are consistent with the Markov model, i.e., they are built only of harmonic transitions found in the training corpus.

Power-Law Metrics

Power laws are statistical models of proportions exhibited by various natural and artificial phenomena. Zipf's law is a special type of a power law, where the probability of a certain event occurring, ***P(f)***, is related to the frequency of occurrence, ***f***, by the following equation:

$$P(f) = 1/f \qquad (1)$$

This equation describes a unique property of Zipfian phenomena, where the 2nd most frequent event appears 1/2 as many times as the 1st most frequent event; the 3rd most frequent event appears 1/3 as many times; the 4th appears 1/4 as many times, and so on. This intriguing relationship (regularity) is found in many human and natural phenomena.

Music, in particular, exhibits near-Zipfian proportions across many dimensions (e.g., pitch, duration, harmonic intervals, distance of repeated notes, etc.). Moreover, power-law behavior (e.g., numeric deviations from the ideal Zipfian proportions) across a multitude of attributes has been shown to be of aesthetic significance (Manaris, et al. 2005, 2007, 2011b). Power-law proportions (i.e., vectors of power-law measurements across numerous dimensions) extracted from musical pieces have been used successfully in previous research to identify contextual information of a given piece, such as author and genre.

Power-law metrics capture longer-term correlations in harmonic flow structure. On the other hand, Markov models capture shorter-term correlations.

Genotype Representation

Each genotype represents a particular harmonic flow (i.e., a sequence of chord choices), which can be generated by the Markov model. The genetic population is initialized by using the Markov model.

The genetic algorithm utilizes mutation and crossover operations to always evolve individuals (harmonic sequences) that are consistent with the Markov model (i.e., they could have been derived by the Markov model itself):

- 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 common subsequence (pivot point) in two individuals (parents) and swaps the corresponding parts. The length of the pivot subsequence is consistent with the Markov order, i.e., the two children could have been generated by the Markov model.

The genetic algorithm uses power-law metrics to derive a measurement from a target piece. Fitness is determined by the proximity of the measurements of evolved individuals to the target piece's measurement.

Conclusion and Future Work

We have presented the Harmonic Navigator, a gesture-based interactive system for exploring harmonic spaces of distinct (or composite) musical styles, and for dynamically generating music in collaboration with human performers. We have also presented one possible user interface.

The Harmonic Navigator may be used to explore compositional ideas in harmonic spaces derived from various musical corpora. These corpora could be used to derive traditional harmonies to be evaluated on a consonance/dissonance scale (Hindemith 1945). This approach could also be expanded to more dense harmonies, which can be similarly evaluated on a consonance/dissonance scale (Legname 1998). The only constraint is that the corpora contain enough musical pieces (for harmonic variety) and are stylistically consistent (although merging of interesting styles that share common harmonic traits is supported).

Another possibility is to revitalize traditional classroom training in tonal harmony via tonal harmony games. "Players" could interactively assign appropriate tonal function and hierarchy to each important pitch in a melody: tonic, predominant or dominant (Berry 1976), and then select from a variety of available chords in various inversions.

Finally, the Harmonic Navigator could be used in musical happenings, together with MIDI and OSC controllers, as well as traditional instruments, to create harmonic contexts for improvised performances. This can also inspire musical games (e.g., the system could be driven through audience participation) to engage, inspire, and possibly challenge musicians in various performance environments.

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

Berry, W.T. 1976. *Structural Functions in Musi*c, Prentice Hall.: 40-57.

Biles, J. A. 2003. GenJam in Perspective: A Tentative Taxonomy for GA Music and Art Systems. *Leonardo* 36(1): 43-45.

Brown, J.C. and Puckette, M.S. 1992. An Efficient Algorithm for the Calculation of a Constant Q Transform. *Journal of the Acoustical Society of America* 92(5): 2698–2701.

Chew, E. and Wu, X. 2004. Separating Voices in Polyphonic Music: A Contig Mapping Approach. *Proceedings of 2nd International Symposium on Computer Music Modeling and Retrieval (CMMR-04)*: 1-20.

Cope, D. 2004. *Virtual Music: Computer Synthesis of Musical Style*. MIT Press.

Dix, A., Finlay, J., Abowd, G, and Beale, R. 1998),. *Human-Computer Interaction*,. 2nd ed., Prentice Hall: 198.

Francese, R., Passero, I., and Tortora, G. 2012. Wiimote and Kinect: Gestural User Interfaces add a Natural Third Dimension to HCI, *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI-12)*: 116-123.

Hindemith, P. 1945. *The Craft of Musical Composition*, Schott: 87-89.

Legname, O. 1997. Density Degree of Intervals and Chords. *20th Century Music* 4(11): 8-14.

Lerdahl, F. 2001. *Tonal Pitch Space*. Oxford University Press.

Manaris, B., Romero, J., Machado, P., Krehbiel, D., Hirzel, T., Pharr, W., and Davis, R.B. 2005. Zipf's Law, Music Classification and Aesthetics. *Computer Music Journal* 29(1): 55-69. MIT Press.

Manaris, B., Roos, P., Machado, P., Krehbiel, D., Pellicoro, L., and Romero, J. 2007. A Corpus-Based Hybrid Approach to Music Analysis and Composition. *Proceedings of 22nd Conference on Artificial Intelligence (AAAI-07)*: 839-845. Vancouver, BC.

Manaris, B., Hughes, D., and Vassilandonakis, Y. 2011a. Monterey Mirror: Combining Markov Models, Genetic Algorithms, and Power Laws. *Proceedings of 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011)*: 33-40. New Orleans, LA, USA.

Manaris, B., Roos, P., Krehbiel, D., Zalonis, T., and Armstrong, J.R. 2011b. Zipf's Law, Power Laws and Music Aesthetics, in T. Li, M. Ogihara, G. Tzanetakis (eds.), *Music Data Mining*: pp. 169-216. CRC Press - Taylor & Francis.

Pachet, F. 2004. Beyond the Cybernetic Jam Fantasy: The Continuator. *IEEE Computer Graphics and Applications* 24(1): 31-35.

Paradiso, J. A. 1999. The Brain Opera Technology: New Instruments and Gestural Sensors for Musical Interaction and Performance. *Journal of New Music Research* 28(2): 130-149.

Rosen, C. 1971, *The Classical Style; Haydn, Mozart, Beethoven*. W.W. Norton & Co,: 26.

Schoenberg, Arnold 1954, *Structural Functions of Harmony*. W.W. Norton & Co: 192-196.

Tymoczko, Dmitri 2006. The Geometry of Musical Chords. *Science* 313: 72-74. Sheridan Press.

Ward, D., Blackwell, A., and MacKay, D. 2000. Dasher - a Data Entry Interface Using Continuous Gestures and Language Models. *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST 2000)*: 129-137. San Diego, CA.

Ward, D., and MacKay, D. 2002, Fast Hands-free Writing by Gaze Direction. *Nature* 418: 838-840.

Wessel, D. and Wright, M. 2002. Problems and Prospects for Intimate Musical Control of Computers. *Computer Music Journal* 26(3): 11-22.