Supporting Compositional Creativity Using Automatic Style-Specific Accompaniment

Ching-Hua Chuan
Department of Mathematics and Computer Science
Barry University
Miami Shores, FL
chchuan@mail.barry.edu

ABSTRACT

In this paper, we described an automatic style-specific accompaniment system and an interactive user interface designed to support creative music composition. With the use of both music theoretical knowledge and statistical learning, the system provides users with an easy start by suggesting a refined accompaniment based on the examples given by users. The interactive user interface allows users to further explore the suggested composition via the manipulation of graphic icons.

Author Keywords

Style-specific accompaniment, neo-Riemannian transforms, statistical learning.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous, H.5.5 Sound and Music Computing.

INTRODUCTION

A growing number of software tools are now available for professionals and amateurs interested in music composition. Some packages focus on providing functionalities such as sound manipulation tools for creating new "instruments," while others increase compositional freedom through the use of sketchpads and other alternative interfaces for planning musical ideas. While these tools certainly stimulate the composition process, non-professional music lovers and inexperienced users often find it difficult to compose songs that achieve the same level of quality as the music they regularly enjoy, giving them a feeling of not knowing where to start in order to realize their musical ideas. When produced segments sound far from the music they have in mind, users may become discouraged and discard their unfinished creations.

In this paper we describe an *automatic style-specific accompaniment* (ASSA) system that makes song writing accessible to both experts and novices. Our effort is inspired by the observation that many people without formal musical training are capable of singing karaoke at a functional or advanced level, but have difficulty creating sophisticated chord arrangements for the melodies that they sing with such ease. To assist them in creating complete compositions, we have designed a system that takes a user-created melody as input and automatically generates a sequence of chords to harmonize the melody, while ensuring that the combination of melody and accompaniment sounds stylistically similar to user-supplied song examples. In this way, users can define the musical style they desire for the composition without using formal musical terminologies, and can create their compositions by exploring style-specific versions suggested by ASSA.

RELATED WORK

Music composition has been a popular domain for studies of creativity for many centuries. In recent decades, the developments of artificial intelligence and machine learning have supported researchers' efforts to apply computational power to analyze the compositional creativity of prominent composers. For example, techniques such as genetic algorithms and Hidden Markov Models have been utilized for modeling Baroque style four-part harmonization [1, 11]. More recently, rule-based logic is used in [13] to model blues musicians' creativity in their improvisation.

Computational power is also the basis for creativity support tools for music composition. Unlike music production software that only provides tools for notating music and recording sound, software such as Hyperscore [8] and Sonic Sketchpad [7] offer alternative methods (e.g., drawing) for composing music without requiring users familiarity with music notation or instrument characteristics.

Our proposed ASSA system and ASSA Visual Editor user interface use computational power to model professionals' compositional creativity and to generate support tools. The details of the system and the user interface are described below.

AUTOMATIC STYLE-SPECIFIC ACCOMPANIMENT

To assist amateur song writers in creating distinctive harmonization, we proposed a hybrid system capable of generating style-specific accompaniments to user-created melodies based on small numbers of examples written by professionals [5]. ASSA combines music theoretic knowledge with statistical learning to model the accompaniment choices made by professionals in their identifiable pieces, and then predicts their compositional

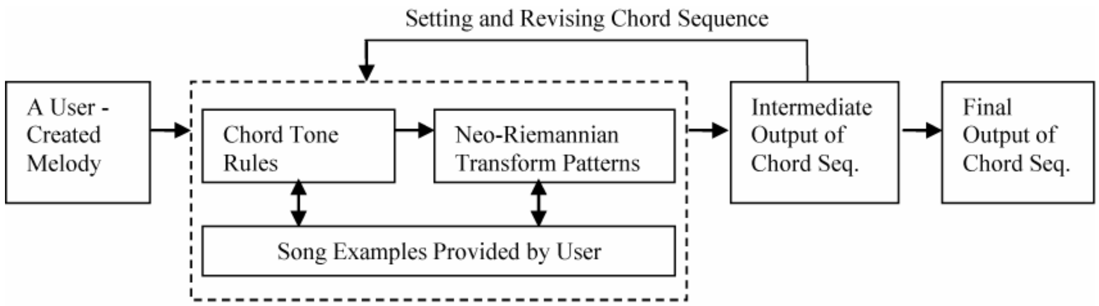

Figure 1. Overview of stages in ASSA modeling accompaniment decisions.

decisions on new, user-created melodies to create foundations for stimulating further creativity.

Figure 1 shows the overview of ASSA and data flow in the system. The core of the system, presented within the dashed box, consists of two modules that model professionals' musical vocabulary in terms of chord tone rules and neo-Riemannian transform patterns. Chord tone rules are constructed using statistical learning to learn the relationship between a melody and harmonizing chords in given examples. Neo-Riemannian transform patterns are generated to describe the transition between adjacent chords, based on a neo-Riemannian operational framework [4]. The system first produces chords for melody segments in which notes tend toward stable harmonization, then revises chord sequences according to probability distributions of neo-Riemannian patterns. Notice that in ASSA, users can control the quality and style of output chord sequence for their own melodies by supplying the system with different song examples.

The following subsections describe the major components of ASSA in details.

Chord Tone Rules

The relationship between melodic notes and chordal harmonies can be expressed in terms of binary classification: if the note is part of the chord structure, then the note is classified as a chord tone; otherwise it is labeled a non-chord tone. A non-chord tone can be further categorized in terms of notes that precede it and follow it — for example, whether such notes are one or one-half step higher or lower, and if they belong to the same scale as the non-chord tone [10]. The varieties of non-chord tone utilized in compositions are part of a unique musical vocabulary of composers, and have been used to analyze and identify individual composers' styles [9].

In ASSA, instead of adopting the rules used in music theory textbooks to describe non-chord tone types, we applied machine learning techniques with the given song examples to derive chord/non-chord tone rules. We believe that rules constructed by machine learning can capture accompaniment styles more precisely and make our proposed system more flexible for learning music in different genres.

Toward this end we utilized decision trees to generate chord tone rules for ASSA. To apply decision trees we identified 73 attributes to describe melodic notes. The attributes represent properties such as pitch class, note duration, relationships among notes in the same bar, the way a note is approached, metrical features, etc. An example of chord tone rules generated by decision trees is shown in Figure 2.

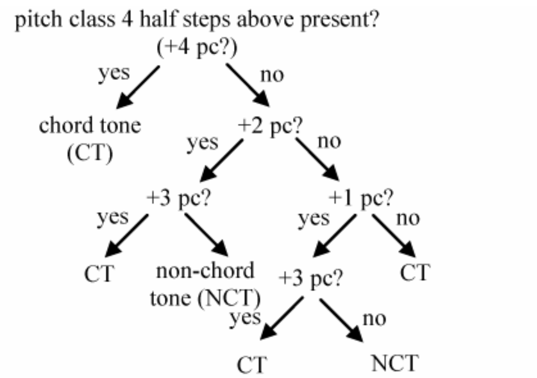

Figure 2. An example of chord tone rules derived by decision trees for British band Keane's *She Has No Time.*

Neo-Riemannian Transform Patterns

Neo-Riemannian transforms have recently been used by music theorists to analyze harmonic patterns and voice-leading in pop-rock music [4]. The four fundamental operations in neo-Riemannian transforms are: I (Identify), L (Leading-tone exchange), P (Parallel), and R (Relative). Single or compound neo-Riemannian operations can be used to describe transitions between two chords — for example, the transition from G major triad to C major triad can be represented as a compound operation RL.

Using neo-Riemannian transforms as the framework for describing chord transitions served as a solution to our biggest challenge in designing ASSA, i.e., only a limited number of examples are available for the statistical learning process. Neo-Riemannian transforms allow for extending the number of possible chord patterns based on very limited numbers of examples, while ensuring that extended patterns are stylistically consistent with the original. The extension process can be explained using the example in the preceding paragraph: the transition from a G major triad to C major triad can be extended to other transitions (e.g., C major to F major), as long as the pattern follows an RL neo-Riemannian operation.

Chord Sequence Setting and Revision

To generate a chord sequence, we first identified melody segments in which chord tones show strong tendencies toward triads[1] as checkpoints. A segment may be a bar consisting of chord tones with their third tone present, or perhaps their third and fifth tones present. By using the selected checkpoints, we can determine chords for the segments with strong evidences of harmony choice independently of the segments with less evidence of harmonicity.

The setting up of checkpoints divides the chord sequence generation task into smaller sections of chord series setting. Instead of finding a chord sequence for the entire melody at once, we generated a suitable series of chords between each pair of the adjacent checkpoints. For segments with less harmonic evidence, we used the probability distribution from neo-Riemannian transform patterns to determine chords based on adjacent chord selections. Each chord series is modeled as a Markov chain, with the likelihood of the series in question being calculated from conditional probabilities learned from song examples. In the final step, all chord series are combined and the most possible combination is selected to generate final chord sequences for the entire melody.

Evaluation of ASSA

To evaluate the effectiveness of ASSA, we examined the degree to which generated accompaniments were consist and specific to the style of given examples. We used one or more songs in a music album as examples for training the system, then used the trained model to generate an accompaniment for the melody of another song (the test song) by the same artist from the same album. To identify similarities between generated accompaniments and originals, we conducted a Turing test in which we played two versions of each test song (one with the original accompaniment and the other with the generated one) and asked a small group of volunteers to identify the accompaniment creator [2]. In addition to this subjective test, we also developed general quantitative methods for evaluating and visualizing the results of machine-generated style-emulating accompaniments [6]. Based on these metrics, ASSA reports that, on average, 25% of the generated chords are exactly the same as the original, and 77% are closely related to the original for the songs in five pop-rock albums.

VISUALIZATION

Our objective for the style-emulation task is to visually represent the musical distance between a computer-generated accompaniment and an artist's original composition. The greater the distance in the visualization, the more distinct the generated accompaniment is perceived as being from the original.

Figure 3 presents an example of the proposed visualization. On the left-upper corner is a chord map, consisting of nine cells representing the closest matches for a given a chord in the original accompaniment. On the right-upper corner is a neo-Riemannian space, showing the relation between the generated and original chords in a specified melody segment in terms of distance and orientation of neo-Riemannian operations. The melody-chord distribution difference graph depicts distinctions between the two accompaniments in relation to the melody. At the bottom of the screen, a second distance graph shows neo-Riemannian distances between individual pairs of generated and original chords for entire songs.

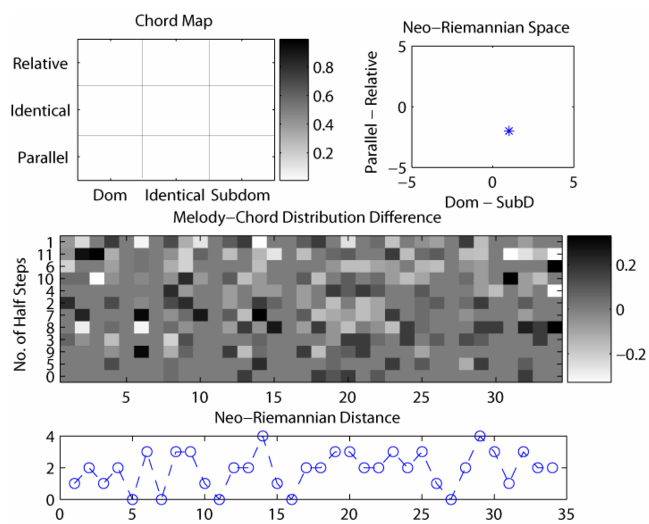

Figure 3. Visualization of a generated accompaniment.

Animation

To visually emphasize the auditory perception, we animated the visualization when the melody, the original accompaniment, and the generated accompaniment are played. The audio portion of the animation makes use of instruments, placement, and rhythm for differentiating between the original and generated accompaniments — for instance, an acoustic guitar for playing generated accompaniments and an alto saxophone for the originals. The audio for the generated accompaniment is placed on the left side and the original on the right side of a stereo field. To play the melody with the generated accompaniment, all the user has to do is turn the balance control to the left during the animation. An animation example is provided on [3].

Interactive User Interface

Following the visualization outlined above, we designed an interactive user interface, called the ASSA Visual Editor,

[1] A triad is a chord consisting of three notes, named *root*, *third*, and *fifth* according to the intervals between them.

for modifying generated accompaniments (Figure 4). A list of accompaniments provided by ASSA for experiments is shown on the left upper corner of the interface. On the center of the interface is an editing panel consisting of two types of visualization: melody-chord distribution difference above, and neo-Riemannian distance below. The editing panel allows for modifying accompaniments by changing the colors and quantities of cells in the top section or changing the position of connected dots in the bottom section. In this manner, users can work on their compositions by modifying a supporting example without using any formal musical terminologies or notation. Notice that no label is depicted for the cells or dots, as we aim to provide users with more freedom to experiment. When an accompaniment is played, the corresponding visualization section is highlighted and the statistical information for the relationship between the accompaniment and original is shown in the statistics panel. The information is updated as the song is played and the accompaniment modified.

A prototype of ASSA Visual Editor can be downloaded from [3].

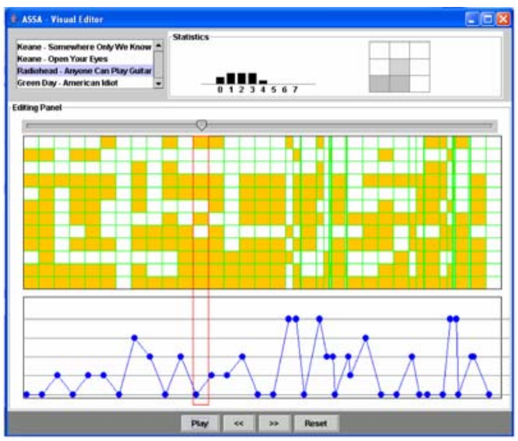

Figure 4. The GUI of ASSA visual editor (assaViEd).

CONLUSIONS AND DISCUSSION

In this paper, we described an automatic style-specific accompaniment system and interactive user interface aimed at supporting music composition creativity. The ASSA system combines music theoretical knowledge and statistical learning in a manner that allows users to explore a range of complex compositional ideas (e.g., changing chordal tones in relation to the melody) by modifying the color and position of graphic icons.

Many principles are currently being proposed for the design of creativity support tools [12] — for example, low thresholds, high ceilings, and wide walls. In the future, we will study user responses to ASSA and the proposed interface, focusing on degree of difficulty for using the program, perceptions of limitations and requirements for assistance among users while composing, and degree of achievement in terms of how their compositions match their musical intentions.

REFERENCES

1. Allan, M., Williams, C.K.I. Harmonising chorales by probabilistic inference. *Proc. of the Neural Information Processing Systems Conference*, Vancouver, 2004.
2. Automatic Style-Specific Accompaniment (Turing test). http://www-scf.usc.edu/~ise575/c/projects/chuan/.
3. Automatic Style-Specific Accompaniment – Visualization and assaViEd User Interface. http://euclid.barry.edu/~chchuan/assa/.
4. Capuzzo, G. Neo-Riemannian theory and the analysis of pop-rock music. *Music Theory Spectrum*, vol. 26, no, 2, pp. 177-199, 2004.
5. Chuan, C.-H., Chew, E. A hybrid system for automatic generation of style-specific accompaniment. *Proc. of the 4th Inter. Joint Workshop on Computational Creativity*, London, 2007.
6. Chuan, C.-H., Chew, E. Evaluating and visualizing effectiveness of style emulation in musical accompaniment. *Proc. Of the 9th Inter. Conf. on Music Information Retrieval*, Philadelphia, 2008.
7. Coughlan, T., Johnson, P. Interaction in creative tasks: ideation, representation and evaluation in composition. *Proc. of ACM Conference on Human Factors in Computing Systems*, Montréal, Canada, 2006.
8. Farbood, M. M., Pasztor, E., Jennings, K. Hyperscore: a graphical sketchpad for novice composers. *IEEE Emerging Technologies*, Jan/Feb, 2004.
9. Huang, C.-Z., Chew, E. Palestrina Pal: a grammar checker for music compositions in the style of Palestrina. *Proc. Of the 5th Conference on Understanding and Creating Music*, Italy, 2005.
10. Kostka, S., Payne, D. *Tonal Harmony*. McGraw-Hill: New York, 2003.
11. Phon-Amnuaisuk, S. Tuwson, A, Wiggins, G. Evolving music harmonization. *Proc of the 4th International Conference on Artificial Neural Nets and Genetic Algorithms*, Portoroz, 1999.
12. Shneiderman, B and et al. Creativity support tools: report from a U.S. National Science Foundation Sponsored Workshop. *International Journal of Human-Computer Interaction*, vol. 20, no. 2, pp. 61-77, 2006.
13. Vempala, N., Dasgupta, S. A computational model of the music of Stevie Ray Vaughan. *Proc. of the 6th Creativity and Cognition Conference*, Washington, 2007.