owner
Oct 21, '08, 10:50 AM
Added Text

owner
Oct 21, '08, 10:50 AM
Deleted: d

owner
Oct 21, '08, 10:50 AM
Added Text

owner
Oct 21, '08, 10:50 AM
Deleted: o

owner
Oct 21, '08, 10:52 AM
Added Text

owner
Oct 21, '08, 10:50 AM
Deleted: al

owner
Oct 21, '08, 10:51 AM
Added Text

owner
Oct 21, '08, 10:51 AM
Deleted: music

owner
Oct 21, '08, 10:51 AM
Deleted: assists

owner
Oct 21, '08, 10:51 AM
Added Text

owner
Oct 21, '08, 10:52 AM
Added Text

owner
Oct 21, '08, 10:53 AM
Added Text

owner
Oct 21, '08, 10:53 AM
Added Text

owner
Oct 21, '08, 10:53 AM
Deleted: i

owner
Oct 21, '08, 10:53 AM
Added Text

owner
Oct 21, '08, 10:54 AM
Deleted: for

owner
Oct 21, '08, 10:54 AM
Added Text

owner
Oct 21, '08, 10:54 AM
Deleted: ing

owner
Oct 21, '08, 10:55 AM
Deleted: such as tools for

Supporting Compositional Creativity Using Automatic Style-Specific Accompaniment

[redacted]
Department of Mathematics and Computer Science
Barry University
Miami Shores, FL
[redacted]

ABSTRACT

In this paper, we describe an automatic style-specific accompaniment system and an interactive user interface that supports music composition creativity. With the use of both theoretical knowledge and statistical learning, the system provides users with an easy start by suggesting a refined accompaniment based upon the examples given by users. The interactive user interface allows users to further explore their compositions by manipulating the graphic icons on the interface.

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

Various software tools have been invented to assist professionals and amateurs in music composition. Some software focuses on providing advanced functionalities that manipulate sounds and create new instruments. Other software aims to increase compositional freedom by creating alternative interfaces (such as sketchpad) for planning musical ideas. While these tools certainly provide stimulation in the composition process, it is difficult for non-professional music lovers to compose songs that reach the same level as the music they listen to under these designs. Many inexperienced users feel lost with an abundance of choices and do not know where to begin in realizing their musical ideas. If, during the composition process, the produced segment sounds far different from the music in mind, users may become discouraged and discard their unfinished creation.

In this paper, we describe an automatic style-specific accompaniment (ASSA) system that makes song writing accessible to both experts and novices. This work is inspired by the fact that, while many people without formal musical training can sing karaoke well, they have difficulty creating sophisticated chord arrangements for the melodies which they sing with such ease. To assist them in creating a complete composition, we 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 the songs supplied by the user. 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 further exploring the style-specific version suggested by ASSA.

RELATED WORK

Music composition has been a popular domain in creativity study for hundreds of years. In recent decades, with the development of artificial intelligence and machine learning, researchers began applying computational power to describe the compositional creativity of prominent composers in their master pieces. 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 to model blues musicians' creativity in their improvisation [13].

Computational power is also used to generate creativity support tools for music composition. Apart from music production software that only provides tools for notating music and recording sound, software such as Hyperscore and Sonic Sketchpad offer alternative methods of creativity like drawing to compose music without the requirement of knowing music notation or playing instruments [8,7].

The ASSA system and user interface ASSA Visual Editor presented in this paper apply computational power in modeling professionals' compositional creativity, as well as generating support tools for amateurs. 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 that is capable of generating style-specific accompaniment to a user-created melody, given only a few examples written by professionals [5]. The system combines theoretical knowledge and statistical learning to model the accompaniment decisions that have made by professionals

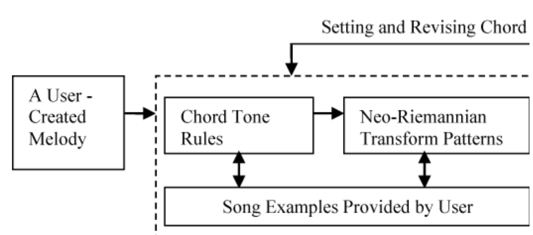

Figure 1. Overview of stages in ASSA modelin

owner
Oct 21, '08, 11:22 AM
Added Text

owner 10/21/08 11:23 AM

Figure 1 should be placed in a way so that it doesn't break up the text so much

owner
Oct 21, '08, 11:24 AM
Deleted: learn

owner
Oct 21, '08, 11:24 AM
Added Text

owner
Oct 21, '08, 11:30 AM
Deleted: in

owner
Oct 21, '08, 11:30 AM
Added Text

owner
Oct 21, '08, 11:30 AM
Deleted: s

owner
Oct 21, '08, 11:31 AM
Added Text

owner
Oct 21, '08, 11:31 AM
Added Text

owner
Oct 21, '08, 11:31 AM
Added Text

owner
Oct 21, '08, 11:31 AM
Deleted: es

owner
Oct 21, '08, 11:31 AM
Added Text

owner
Oct 21, '08, 11:31 AM
Added Text

owner
Oct 21, '08, 11:32 AM
Added Text

owner
Oct 21, '08, 11:32 AM
Added Text

owner
Oct 21, '08, 11:32 AM
Deleted: composers'

owner
Oct 21, '08, 11:32 AM
Added Text

owner
Oct 21, '08, 11:33 AM
Added Text

owner

in their identifiable pieces, and then predicts their compositional choices on a newly (user) created melody to layout a foundation for stimulating the user's musical 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' unique musical vocabulary in terms of chord tone rules and neo-Riemannian transform patterns. Chord tone rules are constructed by using statistical learning to discover the relationship between a melody and the chords that are chosen to harmonize the melody in the given examples. Neo-Riemannian transform patterns are generated to describe the transition between adjacent chords, based on a music theoretical framework of neo-Riemannian operations [4]. The system first produces chords for the segment of melody where notes show stable harmonization tendency, and then revises the chord sequence according to the probability distribution on neo-Riemannian patterns. Notice that in ASSA, users can control the quality and style of output chord sequence for their own melody by simply supplying the system with different song examples.

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

Chord Tone Rules

The relationship between a melodic note and the chord harmonizing that note can be considered as the result of binary classification: if the note is a member of the chord, then the note is classified as a chord tone; otherwise, it is labeled as a non-chord tone. According to the way in which a note is approached and left, a non-chord tone can be further categorized into several types [10]. The types of non-chord tones utilized in compositions are often considered as the unique musical vocabulary of various composers, and have been used to analyze and identify their styles [9].

In ASSA, instead of adopting the rules that describe non-chord tone types in music theory textbooks, we derive chord/non-chord tone rules by applying machine learning techniques on the given song examples. We believe that rules constructed by machine learning can capture the accompaniment style more exactly to the given examples, and thus make the system more flexible for learning pieces in different music genres.

We used decision trees to generate chord tone rules in ASSA. To apply decision trees, we designed 73 attributes to describe the characteristics of a melodic note. The attributes represent properties such as pitch class, note duration, the relation between a note and the other notes in the same bar, the way in which a note is approached and left, metrical features, etc. An example of chord tone rules generated by decision trees is shown in Figure 2.

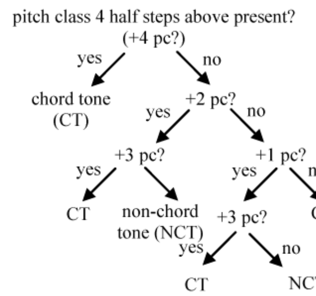

Figure 2. An example of chord tone rules derived by decision trees for British band Keane's *She Has No Time*.

Neo-Riemannian Transform Patterns

In recent decades, Neo-Riemannian transforms have been used by music theorists to analyze harmonic patterns and voice-leading within pop-rock music [4]. There are four fundamental operations in neo-Riemannian transforms, including: I (Identify, same chord), L (Leading-tone exchange), P (Parallel), and R (Relative). We can use a single or compound neo-Riemannian operation to describe the transition 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 helps us overcome the biggest challenge in designing a system like ASSA, i.e., only a limited number of examples are available for the statistical learning process. Neo-Riemannian transforms provide us with the ability to extend the number of possible chord patterns with only a few examples, while ensuring that those extended patterns remain in a style consistent with the original. We can explain the extension process by using the same example mentioned in the previous paragraph: the observation of a transition from G major triad to C major triad can be extended to other patterns, such as from C major to F major, as long as the transition in those patterns follows the RL neo-Riemannian operation.

Chord Sequence Setting and Revision

To generate a chord sequence, we first assign the segments of melody where reported chord tones show a strong tendency of being a triad[1] as checkpoints. Such a segment could be a bar that consists of chord tones having their Third present, or chord tones having their Third and Fifth present. By using the selected checkpoints, we can select chords from the segments with strong evidences of harmony 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 the segments with less harmonic evidence, we used the probability distribution in neo-Riemannian transform patterns to determine chords,

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

owner
Oct 21, '08, 11:51 AM
Added Text

owner
Oct 21, '08, 11:53 AM
Added Text

owner
Oct 21, '08, 11:53 AM
Added Text

owner
Oct 21, '08, 11:55 AM
Deleted: ,

owner
Oct 21, '08, 11:55 AM
Added Text

owner
Oct 21, '08, 11:55 AM
Deleted: and t

owner
Oct 21, '08, 11:55 AM
Added Text

owner 10/21/08 3:17 PM
What could that series do?

owner
Oct 21, '08, 11:51 AM
Deleted: that

owner
Oct 21, '08, 11:51 AM
Deleted: At las

owner
Oct 21, '08, 11:52 AM
Deleted: t

owner
Oct 21, '08, 11:52 AM
Added Text

owner
Oct 21, '08, 11:53 AM
Added Text

owner
Oct 21, '08, 11:54 AM
Added Text

owner
Oct 21, '08, 11:54 AM
Deleted: possible

owner
Oct 21, '08, 11:55 AM
Deleted: most

owner
Oct 21, '08, 11:54 AM
Added Text

owner
Oct 21, '08, 11:56 AM
Added Text

owner
Oct 21, '08, 11:56 AM
Deleted: e

based upon the chord selections in their neighbors. We then modeled each chord series as a Markov chain. The likelihood of that series can be calculated from the conditional probabilities learned from song examples. Finally, we combined all chord series and selected the combination most likely to generate the final chord sequence for the entire melody.

Evaluation of ASSA

To evaluate the effectiveness of ASSA, we examined the degree of consistency and specificity that a generated accompaniment presents with respect to the style in the given examples. We take one or a few songs in an album as examples to train the system, and use 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 understand how similar a generated accompaniment is to the original, we conducted a Turing test in which we played two versions of the test song, one with the original accompaniment and the other with the generated one, and asked participants to identify the creator of the accompaniment [2]. Besides subjective judgments, we proposed general quantitative methods for evaluating and visualizing the results of machine-generated style-emulating accompaniment [6]. Based on the proposed 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

For the style emulation task, our objective is to represent the musical distance and relation between a computer-generated accompaniment and the artist's original piece. The larger the distance (as shown in the visualization), the more perceptually distinct the generated accompaniment can be expected to be 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 nine closest chords within a chord in the original accompaniment. On the right-upper corner is neo-Riemannian space, showing the relation between the generated and original chords in a particular melody segment in terms of distance and orientation of neo-Riemannian operations. The melody-chord distribution difference graph compares difference between the two accompaniments in relation to the melody, respectively. At the bottom, the neo-Riemannian distance graph indicates the neo-Riemannian distance between each pair of generated and original chords for the entire song.

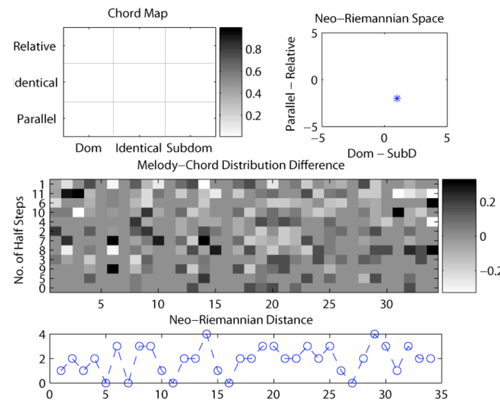

Figure 3. Visualization of a generated accompaniment.

Animation

To emphasize our auditory perception visually, we animated the visualization as the melody, the original accompaniment, and the generated accompaniment. For the audio part in the animation, we used instrument, placement, and rhythm for differentiating the generated accompaniment from the original. For example, we chose acoustic guitar for playing the generated chords, while using alto saxophone for the original. The sound of the generated accompaniment is placed on the left and the original is on the right in the stereo field. Therefore, if the balance control is tuned to the left when the animation is played, only the melody with the generated accompaniment will be heard. An animation example is provided on [3].

Interactive User Interface

Follow the visualization outlined above, we designed an interactive user interface to let users modify a generated accompaniment. The interactive user interface, called ASSA Visual Editor, is shown in 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 the editing panel, consisting of two types of visualization: melody-chord distribution difference on the top and neo-Riemannian distance at the bottom. In the editing panel, users can modify an accompaniment by changing the color (fill or empty) of cells on the top section, or changing the position of connected dots at the bottom. In this way, users can start their compositions by modifying a supporting example without using any formal musical terms. 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 section of the visualization is highlighted, while the statistical information about the relation between the accompaniment and the artist's original is illustrated in the statistics panel. The information is updated as the song is sounded and when the accompaniment is modified. A prototype of ASSA Visual Editor can be downloaded from [3].

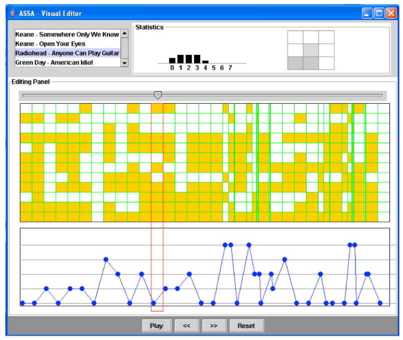

Figure 4. The GUI of ASSA visual editor (assaViEd).

CONLUSIONS AND DISCUSSION

In this paper, we described an automatic style-specific accompaniment system and an interactive user interface to support music compositional creativity. The system combines theoretical knowledge and statistical learning, allowing users to concentrate on higher level decisions and focus on creative ideas. Based on the design of the system, we provided an interactive user interface to let users explore other complex compositional ideas, such as changing note members of a chord in relation to melody by modifying the color and position of graphic icons.

There are many principles proposed for designing creativity support tools [12], such as low threshold, high ceiling, and wide walls. In the future, we will study users' responses on ASSA and its interface, focusing on the degree of difficulty for using the program, the assistance and limitation users feel when compositing, and the degree of achievement in their composition towards their musical goals in mind.

owner
Oct 21, '08, 12:09 PM
Deleted: music

owner
Oct 21, '08, 12:10 PM
Deleted: to

owner
Oct 21, '08, 12:10 PM
Added Text

owner
Oct 21, '08, 12:10 PM
Deleted: ,

owner
Oct 21, '08, 3:19 PM
Added Text

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.