TOWARD AN INTELLIGENT EDITOR FOR JAZZ MUSIC *

G.TZANETAKIS, N.HU, AND R.B. DANNENBERG[†]

Computer Science Department, Carnegie Mellon University
5000 Forbes Avenue,
Pittsburgh, PA 15213, USA
E-mail: gtzan@cs.cmu.edu

The majority of existing work in Music Information Retrieval (MIR) has been concerned with the similarity relations between different pieces of music rather than their internal structure and content. Although a limited number of automatic techniques for analyzing the internal structure and content of musical signals (such as segmentation and structural analysis) have been proposed, there has been little work in integrating these techniques into a common working environment for music understanding. In addition, because of the emerging nature of MIR research, it is necessary to provide interactive tools for experimentation with new algorithms and ideas. As a first step in this direction, a prototype "intelligent" editor for Jazz music has been implemented and is used as a platform for exploring various analysis algorithms and how they can be integrated into a working interactive system. Jazz standards were chosen as a test bed for the developed system because they have many variations, rich internal structures, and raise interesting research challenges. In this paper, we describe the main motivations behind the editor, the algorithms and tools that are currently supported, and our plans for future work.

1. Introduction

The major focus of this work is the automatic segmentation, and structural analysis of Jazz music. In order to explore existing techniques and develop new ones it is necessary to provide interactive experimentation tools. As a first step in this direction, an "intelligent" editor for Jazz music has been implemented. This editor provides a prototype interactive system that allows experimentation with a variety of analysis techniques and their interaction. The two key characteristics of our approach are: *Interactivity*: allow manual specification and editing of all the results, *Integration*: information flows between all the different analysis components. As an example of how these characteristics are manifested in our system, for a section where automatic beat detection is not very accurate the user can manually tap the beat and then the segmentation and structural analysis can be performed based on the manual tapping information. The three main analysis techniques that will be described in this paper are: *Segmentation* which is the

[†] Work partially supported by grant 0085945 of the National Science Foundation.

*Published as: George Tzanetakis, Ning Hu, and Roger B. Dannenberg, "Toward an Intelligent Editor for Jazz Music," in *Digital Media Processing for Multimedia Interactive Services: Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services,* London, UK, 9-11 April 2003. Ed. Ebroul Izquierdo. Singapore: World Scientific Press 2002. pp. 332-333.

detection of changes in instrumental texture (such as the change from orchestra to solo piano in a piano concerto), *Structural Analysis* which the detection of repetition and form (such as ABA) and *Beat Detection* which is the automatic determination of rhythm and tempo information. Although there has been existing work in all these three areas, there has been little work in integrating them in a working interactive system that targets a specific type of music. Jazz standards were chosen as a test bed for the system because they have many variations, rich internal structures, and raise interesting research questions.

2. Related Work

An early work describing the need to have audio editors that are aware of musical content is [1]. Although many of the concepts proposed in the paper are important, the actual system was very limited and constrained by the hardware of that time (1982). Algorithms for segmentation are described in [2, 3]. Beat detection algorithms are covered in [4, 5, 6, 7].

3. Segmentation

Currently the methodology described in [2] is utilized for segmentation. The main idea behind this segmentation method is that changes of sound "texture" correspond to abrupt changes in the trajectory of feature vectors representing the music file. Based on this idea, the following four steps are used to estimate potential segmentation boundaries:

1. A time series of feature vectors V_t is calculated by iterating over the sound file.

2. A distance signal $d(t) = \|V_t - V_{t\_1}\|$ is calculated between successive frames of sound.

3. The derivative $d(t)/\, dt$ of the distance signal is taken. The derivative of the distance will be low for slowly changing textures and high during sudden transitions. The peaks roughly correspond to texture changes.

4. Peaks are picked using simple heuristics and are used to create the segmentation of the signal into time regions. As a heuristic example, adaptive thresholding can be used. A minimum duration between successive peaks is used to avoid small regions.

The resulting segment boundaries, in addition to being directly useful for "intelligent" skipping, can also be used to constrain the structural analysis or be quantized to beat or measure boundaries based on the beat detection.

4. Structural Analysis

Structural analysis [8] uses pattern discovery within a piece of music to identify structure. It can be used to present audio data in musical terms: “here’s where the band plays the bridge on the final chorus.” For example, a form used in popular music and Jazz is AABA, where an 8 measure “A” section is repeated, followed by the “B” section (called the “bridge”) and then ”A” is played again. To discover such patterns, a similarity matrix is contracted containing comparisons between every pair of notes (each note includes the pitch, starting time and duration information), or frames (each frame represents an equal interval of time) in the piece. In Figure 2, the similarity matrix is constructed from the note-based representation of the melody contour transcribed from music by monophonic pitch estimation. The similarity matrix reveals repetitions within the music. These repetitions are then clustered as shown in the upper right of Figure 2. Finally, the data is used to generate an “explanation” of the piece as shown in the lower right of Figure 2. Structural analysis experiments have been conducted with other kinds of data representations, such as chord progression data from polyphonic transcription [8], the spectral data using *Chroma* [12] from the audio [8], and from the MIDI file.

To construct the similarity matrix from the MIDI file, the MIDI piano roll is segmented into a sequence of frames with equal duration. The piano roll within each frame is condensed into a set of pitch classes within one octave (pitch mod 12). The similarity of frame A and B (set of pitch classes), is defined as:

$\sigma(A, B) = |A \cap B| - |A \cup B - A \cap B|$, where $|X|$ is the cardinality of set X.

Two similarity matrixes for the Jazz standard “Satin Doll” but based on different representations are shown in Figure 1. The one at the left is from the MIDI file, and the one at the right is from the *Chroma* representation of the audio synthesized from the same MIDI file. Since the similarity matrix is symmetric, the matrixes were shown with vertical axis representing the time and horizontal axis representing the time lag between every pair of frames. That way the blue vertical lines which represent the repeated sequences are easier to observe. The similarity matrixes in Figure 2 are similar, but the similarity matrix from MIDI has clearer and more elaborate structure than that from *Chroma*, which should be expected as MIDI contains more structure information. However, in some special cases the situation reverses, for example, when the piece has a strong rhythm track that confuses the current simple MIDI ‘condensing’ algorithm. A greedy hill-climbing algorithm [8] is used to recognize the repeated patterns from the similarity matrix. Basically it attempts to automatically distinguish those blue vertical lines similarly to what our eyes do easily.

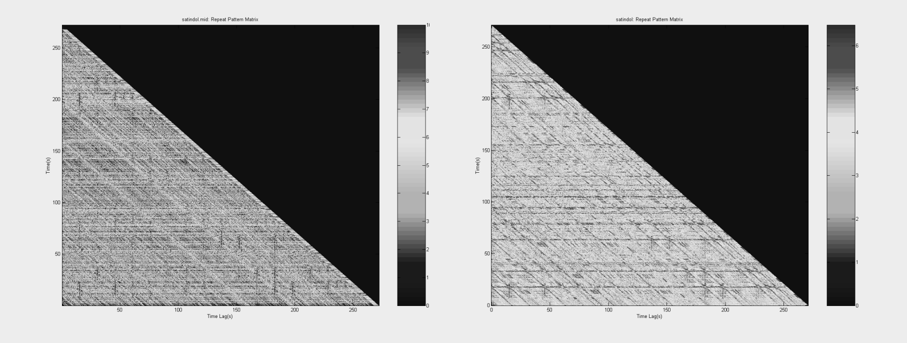

Figure 1. Similarity matrixes (left from MIDI file, right from Chroma data synthesized from MIDI) for “Satin Doll”

5. Beat Detection

The output of the beat detection component is a set of beat locations over the piece and a confidence score for each of them. In addition to manual beat specification using the mouse key three automatic beat extraction algorithms are supported. In [4], a bank of comb-filters is used to extract self-similarity at various levels. A filter bank based on the Discrete Wavelet Transform (DWT) followed by multiple channel envelope extraction and periodicity detection using autocorrelation is used in [6]. Another method based on clustering inter-onset-intervals (IOIs) that are calculated with event detection is described in [5]. Experiments to compare the relative performance of these three algorithms to manual data are under way.

6. The Editor

The editor supports a plug-in architecture for each analysis component and in addition allows the manual editing and specification of results. For example, it is possible to manually tap the beat and then use that information to constrain a segmentation algorithm. *Audacity* [9] `http://audacity.sourceforge.net/` is used as the basis of our system. In addition, the *Marsyas* [10] viewer `http://www.cs.princeton.edu/˜gtzan/marsyas.html` has also been modified to support the same functionality. Figure 1 shows the waveform of an excerpt from the piece “Naima” performed by John Coltrane. Beneath the waveform is the segmentation of the piece. On the right side of the figure, the pairs of similar clusters and the structural analysis of the piece are shown.

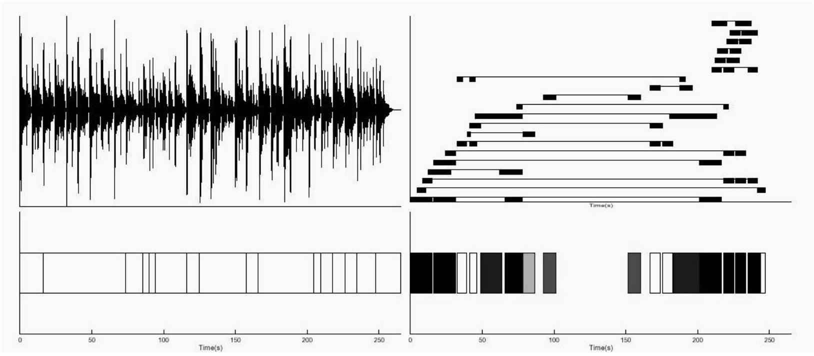

Figure 2. Segmentation (left) and Structural Analysis (right) for "Naima"

7. The Dataset

The dataset consists of three types of files: audio, audio-from-midi and midi. The audio-from-midi files are generated using the *Timidity* software synthesizer. They can be used to evaluate audio analysis algorithms, by first performing the analysis on the audio representation and comparing the results with the more accurate results that can be obtained from the MIDI symbolic representation. Currently the dataset used consists of 37 audio files, 23 midi and 23 audio-from-midi files. Each file contains a performance of a particular Jazz standard from a collection of 20 titles. That way multiple performances of the same title are represented in the dataset. In addition, we are looking forward to using the RWC Music database described in [11] when it becomes available.

8. Future Work

This paper describes work in progress and there are many interesting directions of future work. A comparative evaluation study of different segmentation, structural analysis, beat detection algorithms and their combination is currently under way. In addition, we are planning to explore additional functionality such as query-by-humming for patterns/licks, instrument identification for solos, chord progression detection and others.

REFERENCES

1. C. Chafe, B. Mont-Reynaud and L.Rush, Toward an Intelligent Editor for Digital Audio: Recognition of Musical Constructs, *Computer Music Journal,* 6(1), 30-41, (1982).
2. G. Tzanetakis and P. Cook, Multifeature Audio Segmentation for Browsing and Annotation, *Proc. Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA),* New Paltz, USA, IEEE, (1999).
3. J. J. Aucouturier and M. Sandler, Segmentation of Musical Signals using Hidden Markov Models, *Proc. 110th Audio Engineering Society (AES) Convention,* Amsterdam, The Netherlands, (2001).
4. E. Scheirer, Tempo and Beat Analysis of Acoustic Musical Signals, *Journal of the Acoustical Society of America,* 103(1), 588-601, (1998).
5. S. Dixon, An Interactive Beat Tracking and Visualization System, *Proc. Int. Computer Music Conference (ICMC),* Habana, Cuba, 215-218, ICMA, (2002).
6. G. Tzanetakis and P. Cook, Musical Genre Classification of Audio Signals, *IEEE Transactions on Speech and Audio Processing,* 10(5) July (2002).
7. M. Goto and Y. Muraoka, Music Understanding at the Beat Level: Real-time Beat Tracking of Audio Signals, *Computational Auditory Scene Analysis,* D. Rosenthal and H. Okuno, Eds, 157-176, Lawrence Erlbaum Associates, (1998).
8. R. Dannenberg and N.Hu, Pattern Discovery Techniques for Music Audio, *Proc. Int. Symposium on Music Information Retrieval (ISMIR),* Paris, France, (2002).
9. D. Mazzoni and R. Dannenberg, A Fast Data Structure for Disk-based Audio Editing, *Compute Music Journal,* 26(2), 62-76, (2002).
10. G. Tzanetakis and P. Cook, Marsyas: A Framework for Audio Analysis, *Organized Sound,* 4(3), (2000).
11. M. Goto, et al. RWC Music Database: Popular, Classical and Jazz Music Databases, *Proc. Int. Symposium on Music Information Retrieval (ISMIR),* Paris, France, 287-288 (2002).
12. Bartsch, M. and Wakefield, G.H., To Catch a Chorus: Using Chroma-Based Representations For Audio Thumbnailing, *Proceedings of the Workshop on Applications of Signal Processing to Audio and Acoustics*, (2001), IEEE.