Statistical Modeling and Retrieval of Polyphonic Music

Erdem Unal
Panayiotis G. Georgiou
and Shrikanth S. Narayanan
Speech Analysis and Interpretation Laboratory
University of Southern California
Los Angeles, California 90089
unal@usc.edu, {georgiou,shri}@sipi.usc.edu

Elaine Chew
Music Computation and Cognition Laboratory
University of Southern California
Los Angeles, California 90089
echew@usc.edu

Abstract— **In this article, we propose a solution to the problem of query by example for polyphonic music audio. We first present a generic mid-level representation for audio queries. Unlike previous efforts in the literature, the proposed representation is not dependent on the different spectral characteristics of different musical instruments and the accurate location of note onsets and offsets. This is achieved by first mapping the short term frequency spectrum of consecutive audio frames to the musical space (The Spiral Array) and defining a tonal identity with respect to center of effect that is generated by the spectral weights of the musical notes. We then use the resulting single dimensional text representations of the audio to create n-gram statistical sequence models to track the tonal characteristics and the behavior of the pieces. After performing appropriate smoothing, we build a collection of melodic n-gram models for testing. Using perplexity-based scoring, we test the likelihood of a sequence of lexical chords (an audio query) given each model in the database collection. Initial results show that, some variations of the input piece appears in the top 5 results 81% of the time for whole melody inputs within a 500 polyphonic melody database. We also tested the retrieval engine for small audio clips. Using 25s segments, variations of the input piece are among the top 5 results 75% of the time.**

I. Introduction

Due to advances in computer and network technologies, development of efficient data storage and retrieval techniques have received much attention in recent years. Music Information Retrieval (MIR) is one example of technologies that focus on identifying desired music data within large music collections. The query input to such systems may be of various types, such as modes of natural human interactions (humming, singing, recorded audio samples) or metadata (lyrics, genres, artists.) Given the metadata, retrieval can be straightforward; string matching algorithms that are used in web search engines are capable of these kinds of tasks. On the other hand, when the input query is in the form of audio, signal processing algorithms and music knowledge based techniques need to be incorporated. Query by Example is the problem under discussion in this work, where a system must match an audio query (a polyphonic signal) to similar audio samples that are stored in a database.

A considerable amount of research has focused on the transcription of music audio signal to MIDI or piano roll type representations for accurate understanding of the tonal structures of a polyphonic melody. Numerous researchers have modeled sound events in order to detect musical notes and their onset and offset times. Amongst them, Raphael [1], Pertusa & Inesta [2], Smargadis & Brown [3], Ryynanen & Klapuri [4], and Poliner & Ellis [5] have employed machine learning algorithms such as Hidden Markov Models, Bayesian networks, and Support Vector Machines, which perform well for mono-timbral transcription tasks, such as piano music transcription, where only one single instrument is allowed. These results are promising, however, their extension to a general solution for non-instrument specific polyphonic transcription remains in question.

The common solution to music audio matching and retrieval is to perform symbol-to-symbol comparison within a database to find the most similar, or the exact, matches to the input. Since the main features that are being used for the matching task are features (or symbols) that are extracted from the transcription process, the performance of the transcription directly impacts the performance of the matching and retrieval. In fact, retreival systems can be tolerant against some levels of uncertainty, so that the retrieval problem might be independent of the performance of accurate audio to note transcription.

Initial efforts in polyphonic music retrieval used MIDI transcriptions for modeling melodies. Doraisamy & Ruger [6] used MIDI transcriptions of musical pieces for comparing audio data; n-grams were built from different sets of features that were extracted from MIDI transcriptions, and the cosine rule was adopted for the ranked retrieval.

Pickens et. al. [7] considered the query by example problem as a whole and proposed a general solution. They used existing polyphonic transcription systems in the literature to collect melodic note features from mono-timbral (piano only) music audio. The transcription was then mapped to a harmonic domain, namely a harmonic model that was designed for representing the n length database entries with $24^n \times 24$ matrices corresponding to the distributions of the 24 lexical triads (three-note chords) for the concurrent states. In later studies, Lavrenko & Pickens [8] used random fields to model polyphonic music pieces from MIDI files. Using random fields, they automatically induced new high level features from the melodies, such as consonant and dissonant chords, progressions and repetitions, to efficiently model polyphonic music information.

II. Hyphothesis & Overview

In our work, we use a similar strategy for solving the query by example problem. Our representation schema is slightly different from Pickens et. al.'s in that we prefer single dimensional representations of chord sequences, and we are not directly limited by the performance of the initial audio-to-symbol transcription. Our aim for transcription is a mid-level representation, which is independent of the exact note onsets and offsets, and also independent of the spectral effects of different musical instruments. We try to show that our optimization criterion in this work is not transcription accuracy but the retrieval performance.

We use fixed length audio frames for frequency analysis. The audio frequeny domain is the main feature set we have for melodic representation. We post-process the short term frequency vector to acquire a distribution of the 12 distinct pitch classes (A to G#); their weights are given by the amplitudes of the corresponding fft feature vector. The pitch class vector is than mapped to Chew's Spiral Array representation [9].

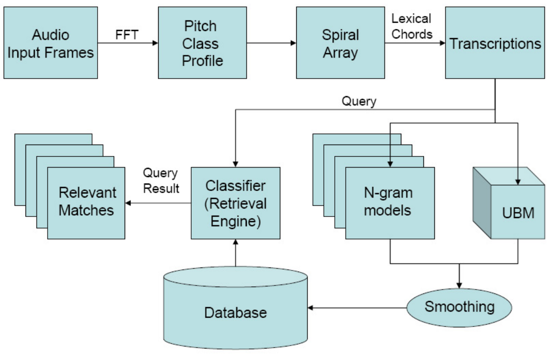

Fig. 1. System Overview

Fig. 1 gives a system overview. From the estimated chord time series representation, we create n-gram language models for the melodies in the database. We then mix these models with a corresponding Universal Background Model for normalization purposes. Given a query, a time series of lexical chords that represents the input melody, we perform perplexity-based scorings for each of the smoothed n-gram models in the database, and attain an N-best list from the entire dataset. Our final goal is to observe how well this transcription accuracy-independent mid-level representation performs on expressive variations of the input melody using an N-best metric.

The rest of the paper is planned as the following. Section III discusses the melodic representation that we use for representing polyphonic music samples. In this section, the algorithm for extracting the pitch class feature vector from fft analysis is explained. In Section IV we briefly discuss the Spiral Array model, and we present how we map the pitch class vector to the Spiral Array for extracting harmonic chord instances at corresponding time windows. Next in Section V, we report the results of our retrieval experiments and conclude the paper in Section VI with further discussion and future work.

III. Melodic Representation

Because of the complex nature of polyphonic music audio, a direct mapping from audio to musical notes is not straightforward. As mentioned in Section I, researchers have attempted to solve the polyphonic transcription problem using a variety of techniques, but their success has been primarily limited to mono-timbral experiments. For this reason, we choose a mid-level representation that can be generalized for any kind of instrumental music audio. Similar to a previous approach by Pickens et. al., we select 24 lexical chords as the representation grammar, spanning all major and minor triads in one full octave. As shown in Fig. 1, we first segment the audio into small frames, then stamp each frame with one of the 24 lexical chords. For chord estimation, we map the frequency spectrum of the corresponding frame to the Spiral Array representation, the details of which will be addressed in the next section.

We use 125ms non-overlapping hamming audio windows and apply the fft algorithm to gather the frequency spectrum of each consecutive frame. Fig. 2 shows the fft spectrum of a random frame in the Molto Allegro movement of Mozart's *Symphony No.40*.

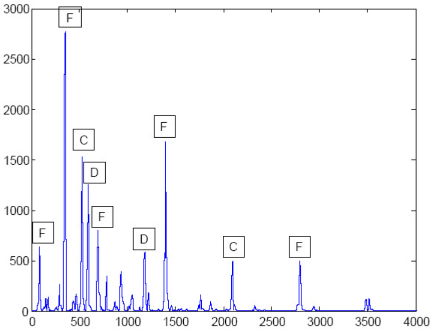

Fig. 2. Frequency spectrum of a random frame generated by fft with note annotations based on peak detection

In the frequency spectrum shown, some peaks are marked with note information on the active pitch (not all are annotated). These peaks are automatically selected by applying a simple curve fitting algorithm on the fft vector. We consider a pitch range from 27.5 Hz (A0) to 3520 Hz (A7). We use a short length sliding window for the fft vector. For the samples inside the window, we fit a parabolic function to the sample points. If the parabolic function is concave and the maximum (the point at which $f'(x) = 0$) lies within the selected fft window, then a peak is assumed to exist. The active pitch in this location is the one that corresponds to the maximum point of the windowed fft vector.

For each detected peak, we record its active pitch, and its amplitude. After all the possible peaks in the frequency spectrum are extracted, the information is accumulated in a vector called the pitch class profile (PCP) which contains information on the energy and the notes detected. The corresponding PCP for the above frame (Fig. 2) is shown in Fig. 3. One can see from the figure that, F, C and D are the most dominant notes in this particular audio window. Now, given the weight profile of the active pitches, our goal is to assign the most meaningful triad to this particular frame, and with the remaining frames, to extract a one dimensional representation for the whole melody. We used the Spiral Array model to achieve this goal.

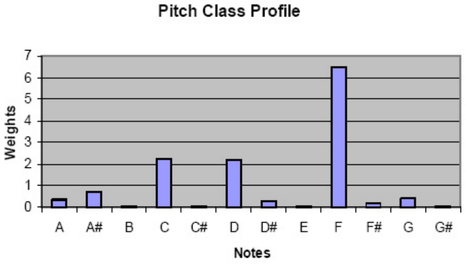

Fig. 3. Corresponding PCP vector for the frequency spectrum of Fig. 2 that shows weights of 12 distinct pitches

IV. Spiral Array

The Spiral Array is a geometric model for tonality that defines representations for pitches, chords, and keys in a three dimensional space. The Spiral Array has been used for key finding [10], [11] and music similarity analysis [12]. We use the Spiral Array to estimate musical chords (major and minor triads) for each frame set, and thus construct a one dimensional harmonic representation of the musical audio in the time domain. According to the model, as shown in Fig. 4, notes that are a Perfect 5th apart from each other are adjacent one to another on the spiral, and pitches that are a Major 3rd apart are vertical neighbors. Please see Chew [9] for other specifications. On the spiral, 24 lexical chords are represented as triangles, an example of which is shown in Fig. 4. A Cmaj chord, consisting of C, the reference pitch, G, the perfect fifth, and E, the major 3rd for which, specific weights are assigned to generate a representative position for the particular triad, marked by a ”star” in the figure above. All such representation points for the 24 major and minor triads are computed inside the spiral.

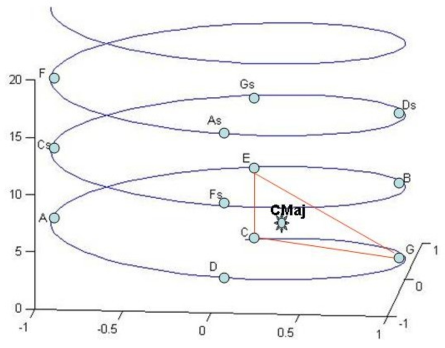

Fig. 4. Spiral Array: pitch locations and the C-Major triad chord

A. Mapping from PCP to the Spiral Array

We use the fft amplitude values from the PCP vector as weights for the corresponding pitch positions on the Spiral Array to calculate a center of effect (CE), a point inside the geometric structure, for the particular PCP. Here, an appropriate selection of pitch locations is required, since pitches within the PCP may have multiple mappings onto the spiral. For example, for $F^{\sharp}$ we should select either $F^{\sharp}$ on the spiral or $G^{\flat}$, which are physically the same, but theoratically different. The pitch spelling algorithm ensures accurate selections of such inconsistencies by simply selecting the closest location for the appropriate pitch value with respect to the CE. Refer to [13] for a more detailed analysis. For identifying the triad associated with this CE, we search for the nearest chord representation. The nearest neighbor chord gives the label for the particular audio window. By successively applying the same algorithm to the remaining frames, we construct a one dimensional text transcription of the audio melody.

B. Modeling

An n-gram is a statistical model of subsequences of n items within a larger sequence, and is in common use in natural language processing applications to model word sequence statistics. We use n-grams to statistically deduce information of the harmonic behavior of polyphonic melodies. As seen from Fig. 1, we store polyphonic melodies in our database in the form of n-gram models in order to quantify the likelihood that a given query sequence of chords is generated by one of the stored melodic models.

To enable the efficient use of this strategy, normalization of the n-gram models is required. We first need to create a Universal Background Model (UBM) to compensate for the variations in text lengths. A UBM is built by concatenating all the available text transcriptions into one single document, and creating an n-gram for this particular collection. By mixing the UBM with each individual melodic n-gram model using a low weight, the required smoothing is also performed. Finally, the collection of the smoothed melodic n-grams constitutes our database.

C. Perplexity-based Evaluation

Perplexity is a common way of evaluating the complexity language models (i.e., its branching factor). In this work we used perplexity to evaluate our melodic models against a given query chord sequence. The perplexity measure gives us the likelihood that the query was generated by a specific probability distribution, namely one of the melodic n-gram models. By the calculated perplexity scores, our retrieval engine gives an N-best list of most likely melody candidates. For creating the n-gram models, performning smoothing by the UBM, and model evaluation, we used the SRILM toolkit [14].

V. Retrieval Experiments Setup

This section describes the evaluation of our proposed model.

A. Data

We downloaded 500 MIDI files from the web for our main melody database, and converted them to wav files. These samples include approximately 150 selections from classical pieces by composers such as Bach, Beethoven, Mozart, and Chopin. The remainder of the 350 examples are variations of the initial 150 samples. Some pieces have only one variation, and some up to seven different variations. These variations include different expressive performances, different orchestrations of the same piece, and variations on an original theme.

B. Retrieval Tests

We performed two sets of retrieval tests. First we use as query one of the selections to see if one of its variations is returned in the N-best list. For instance, when using Mozart's *Symphony No.40* as a test sample, we select "version 0" amongst all relevant documents in the database as the input query, and checked to see if the resulting score table contained versions $1, 2, 3, \ldots, m$ in the N-best list. Setting N to different numbers, we apply the same strategy to all variation groups in the database, and the results are reported in Table I. Correct retrieval occurs when one of the target models is in the N-best list, otherwise the results is classified as an incorrect retrieval.

TABLE I

Retrieval Test Results for whole melody inputs where a correct match is defined when the N-best list includes the target polyphonic audio

	Length of the N-best List			
	N=1	N=5	N=10	N=20
Correct	903	966	982	1008
Incorrect	287	224	208	182
Total	1190			
Accuracy	75%	81%	83%	86%

As expected, from Table I, we can see that the retrieval result improves when we increase the tolerance region in the N-best list. It is critical to set N to a reasonable level, since in practical use, the number of samples in the database can be extremely large. Even for a small scale database, an 81% retrieval accuracy for $N = 5$ is promising, considering the different types variations that are being tested.

For the second set of the tests, we randomly extracted 15s, 25s and 35s audio clips from the sample pieces, and used them as the queries to our retrieval engine. Here, we aim to examine the effect of different input lengths on perplexity-based scoring. For this test, we only used the baseline variation ("variation 0") of all melodies in the database as the source of our short clips. Results for $N = 5$ are reported in Table II. As can be seen from Table II, for 25s clips, we achieved 75% retrieval accuracy within the top 5 of the results list. When the length of the input query increases, the retrieval accuracy improves as expected; this is because more data in the input sequence provides more meaningful comparisons in retrieval. This relative change may be more significant in the cases where the number of samples in the database is large.

TABLE II

Retrieval test results for short clips of audio

	Length of the query		
	15s.	25s.	35s.
Correct	372	385	391
Incorrect	140	127	121
Total	512		
Accuracy	73%	75%	76%

VI. Conclusion

In this paper, we have presented a mid-level representation scheme for polyphonic music audio that is independent of the note level transcription performance. Since the salient pitches form the most important features in the defining of musical chord identity, we claimed that the effect of spectral differences of different instruments can be ignored. This is achieved by mapping the audio spectrum to the Spiral Array model, which accurately tracks the tonal behavior of the melodies for sequential modeling. We used perplexity analysis to understand how likely a query sequences can be generated by melodic models. We also tested the system for different query lengths to observe the effects of query length on the perplexity-based scoring. Results for the retrieval tests showed that, on average, around 80% of the time, we can expect to retrieve a reasonable variation of the polyphonic query in the 5-best list using the proposed mid-level lexical chord representation.

As future work, we plan to test our representation scheme and the retrieval system on a larger database. We would like to test the scalability of the proposed system to large-scale databases in terms of accuracy and computation time. We would also like to expand our chord vocabulary to include 7th chords. This will help the system to represent a wider range of musical samples.

Another future direction could be to build generic models for the pieces that include all its variations. Here the application will be slightly different, since we will have one generic model for each melody as the query target. Instead of trying to retrieve relevant variations, we would then aim to match to a single model for each input. This approach could result in more meaningful retrieval in the face of extremely large databases.

Acknowledgment

This work was funded in part by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152, in part by the National Science Foundation Information Technology Research Grant NSF ITR 53-4533-2720, and in part by ALi Microelectronics Corp. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation and ALi Microelectronics Corp.

References

[1] C. Raphael, "Automatic Transcription of Piano Music," in *Proc. ISMIR International Conference on Music Information Retrieval*, Paris, France, 2002.

[2] A. Pertusa and J. M. Inesta, "Polyphonic Music Transcription Through Dynamic Networks and Spectral Pattern Identification," in *Proc. IAPR International Workshop on Artificial Neural Networks in Pattern Recognition*, Florence, Italy, 2003.

[3] P. Smargadis and J. C. Brown, "Non-Negative Matrix Factorization for Polyphonic Music Transcription," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2003.

[4] M. Ryyananen and A. Klapuri, "Polyphonic Music Transcription Using Note Event Modeling," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2005.

[5] G. E. Poliner and D. P. W. Ellis, "A Discriminative Model for Polyphonic Piano Transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2007.

[6] S. Doraisamy and S. Ruger, "A Comparative and Fault-tolerance Study of the Use of n-grams with Polyphonic Music." in *Proc. International Conference on Music Information Retrieval*, Paris, France, 2002.

[7] J. Pickens, J. B. G. Monti, M. Sandler, T. Crawford, M. Dovey, and D. Byrd, "Polyphonic Score Retrieval Using Polyphonic Audio Queries: A Harmonic Modeling Approach," *Journal of New Music Research*, vol. 32, 2003.

[8] V. Lavrenko and J. Pickens, "Polyphonic Music Modeling with Random Fields," in *Proc. ACM Multimedia 2003*, Berkeley, CA, 2003.

[9] E. Chew, "Towards a mathematical model of tonality," Ph.D. dissertation, Massachusetts Insitute of Technology, Cambridge, MA, 2000.

[10] ——, "Modeling Tonality: Applications to Music Cognition." in *Proc. CogSci2001 23rd Annual Meeting of the Cognitive Science Society*, Edinburg, Scotland, 2001.

[11] C.-H. Chuan and E. Chew, "Polyphonic Audio Key-Finding Using the Spiral Array CEG Algorithm," in *Proc. IEEE-ICME International Conference on Multimedia and Expo*, Amsterdam, Netherlands, 2005.

[12] A. Mardirossian and E. Chew, "Key Distributions as Musical Fingerprints for Similarity Assessment," in *Proc. IEEE-MIPR International Workshop on Multimedia Information Processing and Retrieval*, Irvine,CA, 2005.

[13] E. Chew and Y.-C. Chen, "Real-time pitch spelling using the spiral array," *Computer Music Journal(CMJ)*, vol. 29, 2005.

[14] A. Stolcke, "Srilm – an Extensible Language Modeling Toolkit," in *Proc. International Conference on Spoken Language Processing*, Denver, CO, 2002.