Polyphonic Score Retrieval Using Polyphonic Audio Queries: A Harmonic Modeling Approach

Jeremy Pickens[†], Juan Pablo Bello[‡], Giuliano Monti[‡],
Tim Crawford[♯], Matthew Dovey[♭], Mark Sandler[‡], Don Byrd[♮]
[†]Center for Intelligent Information Retrieval
Department of Computer Science, University of Massachusetts, Amherst
jeremy@cs.umass.edu
[‡]Department of Electronic Engineering, Queen Mary, University of London
juan.bello-correa@elec.qmul.ac.uk
giuliano.monti@elec.qmul.ac.uk
mark.sandler@elec.qmul.ac.uk
[♯]Music Department, King's College, London
tim.crawford@kcl.ac.uk
[♭]Oxford University
matthew.dovey@las.ox.ac.uk
[♮]Indiana University, Bloomington
donbyrd@indiana.edu

ABSTRACT

This paper extends the familiar "query by humming" music retrieval framework into the polyphonic realm. As humming in multiple voices is quite difficult, the task is more accurately described as "query by audio example", onto a collection of scores. To our knowledge, we are the first to use polyphonic audio queries to retrieve from polyphonic symbolic collections. Furthermore, as our results will show, we will not only use an audio query to retrieve a known-item symbolic piece, but we will use it to retrieve an entire set of real-world composed variations on that piece, also in the symbolic format. The harmonic modeling approach which forms the basis of this work is a new and valuable technique which has both wide applicability and future potential.[1]

1. INTRODUCTION

Music information retrieval is a rapidly growing field. As more music collections come online, the demand to search these collections increases. Music collections, or sources, exist in one of two basic formats: audio and symbolic. To complicate matters, music queries exist in both formats as well. A comprehensive music retrieval system should be able to allow queries in either format to retrieve music pieces in either format. The problem lies in the fact that the features readily available from audio files (MFCCs, energy) do not correspond well with the features available from symbolic files (note pitches, note durations) It is a "vocabulary mismatch" problem.

Our system will bridge the gap between audio and symbolic music using transcription algorithms together with harmonic modeling techniques. In this manner we allow users to present queries in the audio format and retrieve pieces of music which exist in the symbolic format. This is one of the earliest goals of music retrieval, and until now it has only been possible within the monophonic domain. We extend the realm of possibility into the remarkably more difficult polyphonic domain, and show this through successful retrieval experiments for both known-item and variation queries. The ability to use polyphonic audio queries to retrieve pieces of music from a polyphonic symbolic collection is a major step forward in the field.

The remainder of this paper proceeds as follows: In Section 2 we give a brief review of the problem domain and existing literature. Section 3 locates this paper within the larger framework of the "language" modeling approach to Information Retrieval. Section 4 contains an overview of our system. In Section 5 we explain our audio music transcription techniques. In Section 6 we explain our harmonic modeling techniques, while in section 7 we show how two models are compared for dissimilarity. Finally, Sections 8 and 9 contain our experimental design, results, discussion and conclusion.

2. BACKGROUND AND RELATED WORK

To date, research in the field of ad hoc music retrieval has experienced two fundamental divisions. The first division is one of representation. Music may either be presented as a performance or as instructions to the performer. A performance is an audio file, in a format such as WAV or MP3. Instructions to the performer exist in a symbolic format, either as a MIDI file (www.midi.org) or in Conventional Music Notation (CMN) format [1], both of which express some manner of instructions about what notes should be played, when, for how long, and with what instrument or dynamic.

This division between actualized performance and instructions for a performance manifests itself in the types of features readily extractable from digital forms of audio and symbolic music. Those retrieving audio tend to work with features such as MFCCs, LPCs, centroids, or energy, while those retrieving symbolic sources use actual note pitch and/or duration, as these values are known.

The second division in music IR is one of complexity, or monophony versus polyphony. Monophonic music has at most one note playing at any given time; before a new note starts the previous note must have ended. Polyphonic music has no such restrictions. Any note or set of notes may begin before any previous note or set of notes has ended, which proves difficult for any clear, unambiguous sense of sequentiality. Therefore, techniques which work for monophonic music, such as string matching or n-gramming, are more difficult to apply to the polyphonic domain. Furthermore, reasonably accurate conversions from audio to symbolic music is generally seen as a solved (or at least manageable) problem for monophonic music, but still a fairly inaccurate, unsolved problem for polyphonic music.

[1]This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-9905842. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

Polyphonic music in general is more complex and difficult to work with. Indeed, some of the earliest works in music retrieval remained entirely within the monophonic domain [16, 25]. These "query by humming" systems allow the query to be presented in audio format, and then converted to symbolic format to be used for query on a monophonic symbolic collection. Gradually, systems which allowed monophonic queries upon a polyphonic collection, a more difficult prospect, were introduced [5, 21, 35]. The query is still monophonic, so conversion of the query between audio and symbolic formats remains possible. The collection to be searched may therefore be audio or symbolic, as the query may easily be converted in either direction to match. But again, this is only possible because the query is monophonic.

Most recently, polyphonic queries upon a polyphonic collection have become possible. Yet because of the complex nature of polyphonic music and the difficulty of accurate conversion, researchers tend not to mix the audio and symbolic domains. Research has either focused on polyphonic audio queries upon polyphonic audio collections [14, 31, 34], or polyphonic symbolic queries upon polyphonic symbolic collections [6, 11, 10, 26, 29]. We know of no prior work which tackles polyphony, audio, and symbolic music all in the same breath.

Of the papers mentioned above, the one that most closely resembles our work is Purwins et al [31]. These authors have devised a method of estimating the similarity between two polyphonic audio music pieces by fitting the audio signals to a vector of key signatures using real-valued scores, averaging the score for each key fit across the entire piece, and then comparing the averages between two documents. As do we, these authors use Krumhansl distance metrics [20] to assist in the scoring. One of the main differences, however, is that these authors attempt to fit an audio source to a 12-element vector of keys, while we fit a symbolic source to a 24-element vector of major and minor triads. Furthermore, by averaging their key-fit vector across the entire piece, their representation is analogous to our 0^{th}-order Markov models. Our paper utilizes not only 0^{th}-order models, but 1^{st} and 2^{nd}-order models as well. Moreover, the Purwins paper was not specifically developed as a music retrieval task, and thus has no retrieval-related evaluation. We present comprehensive known-item as well as recall-precision results.

Finally, a paper by Shmulevich et al [33] also uses some of the same techniques presented here, such as Krumhansl's distance metrics and the notion of smoothing, the latter which will be presented in section 6.2. The domain to which these techniques are applied are monophonic, but Shmulevich's work nevertheless demonstrates that harmonic analysis and probabilistic smoothing can be valuable components of a music retrieval system.

3. LANGUAGE MODELING APPROACH

Language Modeling (LM) has received much attention recently in the text information retrieval community. It is only natural that we wish to leverage some of the advantages of LM and apply it to music. Ponte explains some of the motivations for this framework:

> [A language model is] a probability distribution over strings in a finite alphabet (page 9)... The approach to retrieval taken here is to infer a language model for each document and to estimate the probability of generating a query according to each model. The documents are then ranked according to these probabilities (page 14)...The advantage of using language models is that observable information, i.e., the collection statistics, can be used in a principled way to estimate these models and do not have to be used in a heuristic fashion to estimate the probability of a process that nobody fully understands (page 10)...When the task is stated this way, the view of retrieval is that a model can capture the statistical regularities of text without inferring anything about the semantic content (page 15)." [30]

Even though our retrieval task is polyphonic music rather than text, we are duplicating the LM framework by creating statistical models of each piece of music in a collection and then ranking the pieces by those statistical properties. Thus, while it might be more appropriate to name this work "statistical music modeling", we still say that we are taking the language modeling *approach* to information retrieval. So rather than attempting a formal analysis of the harmonic structure of music, we instead "capture the statistical regularities of [music] without inferring anything about the semantic content".

Nothing illustrates this more than our choice, explained in section 6, to characterize the harmony of a piece of music at a certain point as a *probability distribution* over chords, rather than as a single chord. Selecting a single chord is akin to inferring the semantic meaning of the piece of music at that point in time. While useful for some applications, we feel that for retrieval, this semantic information is not necessary, perhaps even harmful if the incorrect chord is chosen. Rather, we let the statistical patterns of the music speak for themselves.

To our knowledge, the first LM approach to music IR was done in the monophonic domain [28]. Other recent techniques, which also take the LM approach (though without always explicitly stating it), apply 1^{st}-order Markov modeling to monophonic note sequences [32, 17]. Further work extends the modeling to the polyphonic domain, using both 0^{th} and 1^{st}-order Markov models of raw note simultaneities to represent scores [4].

4. SYSTEM OVERVIEW

The goal of this system is to take polyphonic audio queries and return polyphonic symbolic pieces of music, highly ranked, which are relevant to the given query. This is done in a number of stages, as outlined in Figure 1.

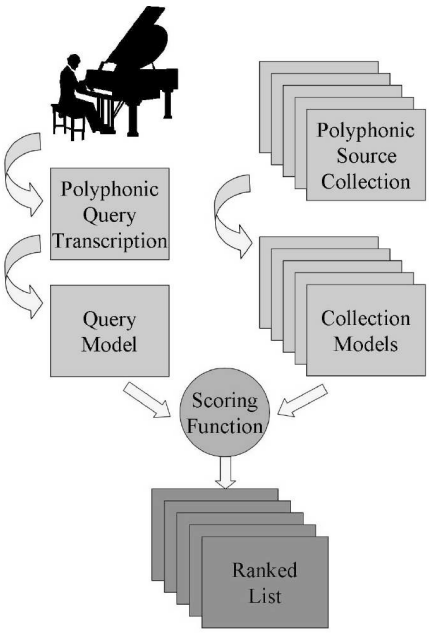

Figure 1: System Overview

Offline and prior to query time, the entire source collection (the set of polyphonic scores which are to be searched) is passed through the harmonic modeling module, described in Section 6. Each piece of music, each document, is then "indexed", or stored, as a model. At query time, the system is presented with polyphonic audio, such

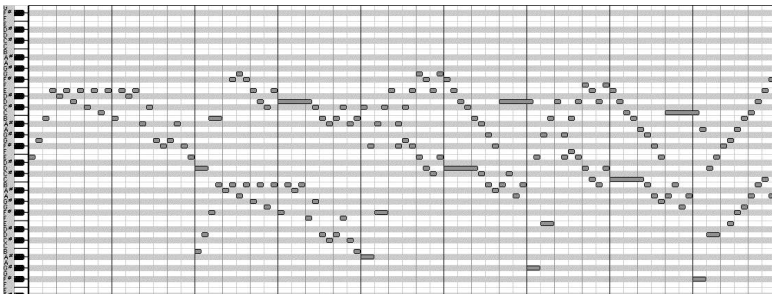

Figure 2: Bach Fugue #10 Original Score

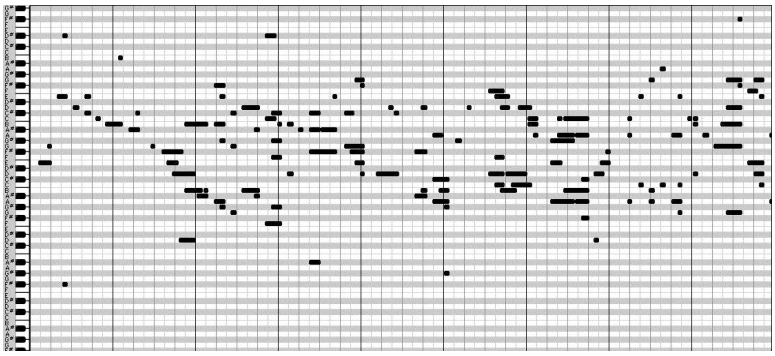

Figure 3: Bach Fugue #10 from Polyphonic Transcription II algorithm

as a digitized recording of a piano piece from an old LP. The query is first passed through the audio transcription module, described in Section 5. The transcription from this module is passed to the harmonic modeling module, and a model for the query is created.

Finally, a scoring function is used to compare the query model with each of the document models, and give each query-document pair a dissimilarity value. Documents are then sorted, or ranked, by that value, with the least dissimilar at the top of the list.

5. AUDIO TRANSCRIPTION

Automatic music transcription is the process of transforming a recorded audio signal into a representation of its musical features. We will limit our definition to the estimation of onset times, durations and pitches of the notes being played. This task becomes increasingly complicated when dealing with polyphonic music because of the multiplicity of pitches, inconsistent durations, and varied timbres. Most monophonic transcription techniques are therefore not applicable. In fact, despite several methods being proposed with varying degrees of success [9, 19, 23, 24], automatic transcription of polyphonic music remains an unsolved problem.

We offer two figures as an example of this transcription procedure. Figure 2 is the original score of Bach's Fugue #10 from Book I of the Well-tempered Clavier, presented here in piano-roll notation. A human musician then performs this piece, and the audio signal is digitized. Figure 3 is the transcription of this digitized audio from one of our algorithms. It is with this imperfect transcription that we still achieve excellent retrieval results.

We locate the audio transcription task within the context of Computational Auditory Scene Analysis (CASA). In this context, systems try to explain the analysed signal following a set of perceptual rules and sound models. These rules suggest how to group the elements from the signal time-frequency representation into auditory objects (i.e. musical notes). In polyphonic music, events overlap both in the time and the frequency domain, meaning that transcription systems should be able to analyse the signal in both domains in order to return an accurate representation of the scene. From this approach we propose two different methods. Both techniques will be used, separately, to produce queries, and retrieval results for each transcription technique will be given. We do this to show that our harmonic modeling algorithm is robust to varying transcriptions and their associated errors.

5.1 Polyphonic Transcription I

Our first method is an extension and reworking of a technique used for monophonic transcription in Monti [27]. Fourier analysis is used to represent the signal in the frequency domain. An auditory masking threshold is calculated using a perceptual model. Only spectral maxima above such a threshold are chosen to represent the signal. The Phase-Vocoder technique is used to calculate the instantaneous frequencies of the peaks, by interpolating the phase of two consecutive frames. The analysis is optimised for the steady state part of the notes.

Once the representation of the signal is given as a set of spectral peaks, the system groups the peaks according to their frequency position and time evolution. The grouping rules are: harmonic relation in the frequency domain and common onset in the time domain. For the implementation of these rules, which group peaks into objects (notes) we used the Blackboard model [12]. This model has shown great flexibility and modularity, which is important when implementing additional rules.

The system starts selecting the lowest available frequency peak and, assuming it to be a note's fundamental, looks for harmonic support among the other peaks. The support of a note hypothesis is given by a fuzzy rate depending on the fundamental frequency position and energy, and the harmonic support in the spectrum. If the note is confirmed as an hypothesis, its harmonic peaks are eliminated from the hypothesis space so they cannot be chosen as new fundamental hypotheses. However, they still may contribute to other notes' hypotheses since the partials of the notes composing a chord often overlap in western music.

The algorithm iterates while there are peaks in the spectrum. Hypotheses qualify as note objects, only if they last in time for a mini-

mum number of (activation) frames. Once a note is recognised the system predicts its evolution in the spectrum, and in future analysis the existing notes are verified before searching for new notes. If the spectrum reveals change in the frequencies' positions or amplitude the system formulates new note hypotheses corresponding to the new events detected. Using this method, octave errors are eliminated, but at the cost of failing to detect octave intervals when played simultaneously. The system extracts onsets, offsets and MIDI pitches from the audio and writes them in a MIDI file for listening and retrieval tasks.

5.2 Polyphonic Transcription II

Our second system is an extension of work found in Bello [2, 3]. We again begin by apply Fourier analysis on overlapping frames in the time-domain. The phase-vocoder technique is also used to estimate the exact instantaneous frequency value for each bin in the frequency-domain representation. However in this approach all frequency peaks are used, regardless of their perceptual conditions.

Two levels of hypotheses are considered here. On each analysis frame, all musical notes within the evaluated range (from 65 to 2kHz) are considered to be 'frame' hypotheses. Associated with each of these frame hypotheses a filter is developed in the frequency domain. To do this we assume that a note with fundamental frequency f_k must (theoretically) present frequency partials located according to:

$$f_{m,k} = m \cdot f_k \sqrt{1 + (m^2 - 1) \cdot \beta_k} \qquad (1)$$

where β_k is the inharmonicity factor (note and instrument dependent) [13], and $m = 1 \ldots M$, with M such that $f_{M,k} \leq f_s/2$. The filter associated with f_k behaves like a comb filter with lobes centered at the expected partials' frequencies and bandwidths equal to half the tone-distance between the hypothetical note and its closest neighbour (a quarter or half a tone depending on the note).

The frame's frequency-domain is processed through this filter-bank, producing a group of spectrums associated with each of the frame-hypotheses. The hypotheses are rated according to the ratio between the filtered spectra energy and the energy of the original spectrogram. Hypotheses with high ratings are classified as 'note' hypotheses and followed over time. If continuity and envelope conditions are satisfied, then the note is recognised as a note-object of the signal.

Note that in this approach, no onset detection is performed on the audio signal. Timing information depends on the behaviour of the instantaneous rating of each possible note. A smoothing window is used to group events that are very close in time.

An important difference from the previous approach is that frame hypotheses are evaluated independently, allowing any interval to be detected. This brings as a consequence the detection of octave intervals and the proliferation of octave-related errors. As with the previous transcription algorithm, the system extracts onsets, offsets and MIDI pitches from the audio and writes them in a MIDI file for listening and retrieval tasks.

6. HARMONIC MODELING

A harmonic model is our term for a Markov Model in which the states of the model are musically salient, harmonic entities. The process of transforming polyphonic music into a harmonic model divides into three stages. In the first stage, *harmonic description*, the music document to be modeled is broken up into sequences of note sets, and each of those note sets are fit to a probability vector. Each of these note sets is assumed to be independent of the neighboring sets. This assumption, while necessary for the modeling, is not always accurate, in particular because harmonies in a piece of music are often defined by their context. The second stage of the harmonic modeling process is therefore a *smoothing* procedure, designed to account for this context. Finally, the third stage is the process by which *Markov models* are created from the smoothed harmonic descriptions. Stages one and three are covered in greater detail in [29], while stage two is a new technique first described in this paper.

6.1 Harmonic Description

Recall from Section 1 that polyphonic music has no innate, one-dimensional sequence. Arbitrary notes or sets of notes may start before the current note or set of notes has finished playing. It therefore becomes necessary for us to artificially impose sequentiality. This is accomplished by ignoring the played duration for every note in a score, and then selecting at each new note onset all the notes which also begin at that onset. These event-based sets are then reduced, mod 12, to octave-equivalent pitch classes and given the name *simultaneity*.

We define a *lexical chord* as a codified pitch template. Of the 12 octave-equivalent (mod 12) pitches in the Western canon, we select some n-sized subset of those, call the subset a *chord*, give that chord a name, and add it to the lexicon. Not all possible chords belong in a lexicon; with $\binom{12}{n}$ possible lexical chords of size n, and 12 different choices for n, we must restrict ourselves to a musically-sensible subset. The chord lexicon will furthermore make up the state space of our Markov model, in addition to providing the basis for the harmonic description.

The chord lexicon used in this paper is the set of 24 major and minor triads, one each for all 12 members of the chromatic scale: C Major, c minor, C♯ Major, c♯ minor ... B♭ Major, b♭ minor, B Major, b minor. No distinction is made between enharmonic equivalents (C♯/D♭, A♯/B♭, E♯/F, and so on). Assuming octave-invariance, the three members of a major triad have the relative semitone values n, $n + 4$ and $n + 7$; those of a minor triad n, $n + 3$ and $n + 7$.

During the 1970s and 1980s the music-psychologist Carol Krumhansl conducted a ground-breaking series of experiments into the perception and cognition of musical pitch [20]. By using the statistical technique of multi-dimensional scaling on the results of experiments on listeners' judgements of inter-key relationships, she produced a table of coordinates in four-dimensional space which provides the basis for the lexical chord distance measure we adopt here. The 'distance' between triads a and b can be expressed as the four-dimensional Euclidean distance between these coordinates. We do not reproduce these distances here, but denote the distance as $Edist(a, b)$.

Now that these definitions are clear, we may proceed with the harmonic description algorithm. The basic idea is that when calculating the score of a simultaneity s on a lexical chord c, this score is influenced by all the other lexical chords p in which s participates. Thus, every lexical chord has an effect on every other lexical chord.

An analogy might help: The amount of gravitational force that two bodies (such as the earth and moon) exert on each other is proportional to the product of their masses, and inversely proportional to a function of the distance between them. By analogy, each of our 24 lexical chords is a body in space, and each exerts some influence on all others. Thus, if the notes of a G major triad are observed, not only does G major get the most mass, but we also assign some probability mass to E minor and B minor, a bit less to C major and D major, even less to A minor and F♯ minor, and so on.

So the amount of influence exerted by each chord in the lexicon on the current chord is proportional to the number of pitches shared between the simultaneity s and each lexical chord p, and inversely proportional to the inter-triad distance from each p to c. Since, in general, 'contributions' of near neighbors in terms of inter-key distance are preferred, we use that fact as the basis for computing a suitable context:

$$Context(s,c) = \sum_{p \in lexicon} \frac{|s \cap p|}{Edist(p,c)+1} \quad (2)$$

This context score is computed for every chord c in the lexicon (each point in the distribution), and then the entire distribution is normalized by the sum total of all context scores. While it is clear that the harmony of all but the crudest music cannot be reduced to a mere succession of major and minor triads, as this choice of lexicon might be thought to assume, we believe that this is a sound basis for a probabilistic approach to harmonic description, as more complex chords (such as 7^{th} chords) are in fact accounted for by the contributions of their notes to the overall probabilistic context.

6.2 Smoothing

While the method above takes into account contributions from neighboring triads, it only does so within the current simultaneity, the current timestep. Harmony, as musicians perceive it, is a highly contextual phenomenon which depends not only on the harmonic distances at the current timestep, but is also influenced by the previous timesteps: the harmonies present in the recent past are assumed to be a good indication of the current harmony. Thus, a simultaneity with only one note might provide a relatively flat or uniform distribution across the lexical chord set, but when that simultaneity is taken in historical context, the distribution becomes more accurate.

We have developed a naive, yet effective, technique for taking into account this event-based context by examining a window of n simultaneities and using the values in that window to give a better estimate for the current simultaneity. This is given by the following equation, where s_t is the simultaneity at timestep t:

$$Smoothed(s_t,c) = \sum_{i=1}^{n} \frac{1}{i} \left(\sum_{p \in lexicon} \frac{|s_{t-i+1} \cap p|}{Edist(p,c)+1} \right) \quad (3)$$

When the smoothing window n is equal to 1, this equation degenerates into the one from the previous section. When n is greater than one, the score for the lexical chord c at the current timestep is influenced by previous timesteps in proportion to the distance (number of events) between the current and previous timestep. As in the unsmoothed version, the smoothed context score is computed for every chord c in the lexicon and then the entire distribution is normalized by the sum total.

6.3 Markov Modeling

It should be clear by now that the primary difference between our harmonic description algorithm and most other such algorithms is the choice to create probabilistic *distributions* across the lexical chord set, rather than *reductions* of each simultaneity to a single, most salient lexical chord. The figure below is a toy example of a harmonic description, using an example lexicon of three chords, P, Q, and R. With this probabilistic harmonic description, we now create a Markov model.

	Timestep (Simultaneity)				
Lexical Chord	1	2	3	4	5
P	0.2	0.1	0.7	0.5	0
Q	0.5	0.1	0.1	0.5	0.1
R	0.3	0.8	0.2	0	0.9

Markov models are often used to capture statistical properties of a state sequence over time. We want to be able to predict future occurrences of a state by the presence of sequences of previous states. In our harmonic approach, we have chosen lexical chords as the states of the model. For an n^{th}-order model, a $24^n x 24$ matrix is constructed, with the 24^n rows representing the *previous state* space, and the 24 columns representing the *current state* space. An $(n+1)$ sized window slides over the sequence of lexical chord distributions and Markov chains are extracted from that window. The count of each chain is added to the matrix, where the cross of the first n states is the previous state, and the $(n+1)^{th}$ state is the current state. Finally, when the entire observable sequence has been counted, each row of the matrix is individually summed and the elements of each row normalized by the sum total for that row.

One problem is that Markov modeling only works on 1-dimensional sequences of observable states, while our harmonic description is a sequence of 24-point probability distributions. Our solution is to assume independence between points in each distribution at each timestep, so that an exhaustive number of independent, one-dimensional paths through the sequence may be traced. (This exhaustive paths approach is abstractly similar to one suggested by Doraisamy and Rüger [10].) Each path, thus constructed, is not counted as a full observation. Instead, observations are proportional; the degree to which each path is observed is a function of the amount by which all elements of the path are present. Since independence between neighboring simultaneities was assumed, this becomes the product of the values of each state which comprises the path. For example, suppose we construct a 2^{nd}-order model from the sequence of distributions, above. Then one of the many observed state sequences we would see in timesteps 1 to 3 is "QRR". The count of this observation is 0.08 = (0.5 * 0.8 * 0.2).

7. SCORING FUNCTION

Our goal is to produce a ranked list for a query across the collection. We wish to rank those pieces of music at the top which are most similar to the query, and those pieces at the bottom which are least similar. This is the task of the scoring function. We have chosen as this function the Kullback-Liebler (KL) divergence, a measure of how different two distributions are, over the same event space. The divergence is always zero if two distributions are exactly the same, or a positive value if the distributions differ. We denote the KL divergence between query model q and music document model d as $D(q||d)$. "The KL divergence between $[q]$ and $[d]$ is the average number of bits that are wasted by encoding events from a distribution $[q]$ with a code based on the not-quite-right distribution $[d]$" [22].

In our Markov model, each previous state, each row in the $24^n x 24$ matrix, is a complete distribution. We therefore compute a divergence score for each row in the model, and add the value to the total divergence score for that query-document pair. This is given by the following equation, where q_i and d_i represent each previous state. It is imperative that the same modeling procedure and size that is used for the document models is also used for the query model.

$$D(q||d) = \sum_{q_i \in q, d_i \in d} \left(\sum_{x \in X} q_i(x) \log \frac{q_i(x)}{d_i(x)} \right) \quad (4)$$

However, there is a problem in that sometimes a document model can have estimates of zero probability. This is especially true of shorter music documents, in which a lot of the possible transitions are never observed. The divergence score in such cases ($q_i(x) \log \frac{q_i(x)}{0}$)) automatically goes to infinity. This small problem in just a single value could therefore throw off our entire score for that document. We therefore must create some small but principled non-zero value for every document model zero value. There are many ways to do this, but we have done so by "backing off" to a general music model, using the value of that previous state node from the general model whenever we encounter a zero value in any particular document model.

A general music model is created by averaging the models over the entire set of document models in the collection. In principle, there could still remain zero values in the general music model, depending on the size and properties of the collection. In our experiments,

Table 1: Average Ranks for Transcription I

Bach Preludes			
	mm0	mm1	mm2
Window 1	4.83	23.11	219.41
Window 2	4.83	4.83	13.98
Window 3	4.76	3.52	4.30
Window 4	4.83	3.17	**3.04**
Random = 1575			

Bach Fugues			
	mm0	mm1	mm2
Window 1	4.04	35.08	192.08
Window 2	3.63	5.69	10.58
Window 3	3.31	5.19	3.52
Window 4	3.23	4.02	**2.38**
Random = 1575			

Table 2: Average Ranks for Transcription II

Bach Preludes			
	mm0	mm1	mm2
Window 1	8.91	28.72	223.87
Window 2	7.85	5.04	16.72
Window 3	7.54	**3.83**	6.85
Window 4	7.35	4.87	7.96
Random = 1575			

Bach Fugues			
	mm0	mm1	mm2
Window 1	6.08	24.88	142.92
Window 2	5.33	4.77	10.23
Window 3	6.10	3.75	3.63
Window 4	5.79	3.58	**2.60**
Random = 1575			

however, we found this almost never to be the case. Also, it should be observed that when the query model has a zero probability in any cell, there is no problem. The KL divergence for that point is $0 \log \frac{0}{d_i(x)}$, which is zero.

8. EXPERIMENT DESIGN AND RESULTS

For our retrieval experimentation, we adopt the Cranfield evaluation model[2] [8]. This requires three crucial components: (1) Source collection, (2) Query, and (3) Relevance judgements which label each item in the source collection as either relevant or not relevant to the query. In all our experiments, the source collection remains the same. However, we vary the queries and the relevance judgements, as described below.

8.1 Source Collection

The basic test collection on which we tested our retrieval method was assembled from data provided by the Center for Computer Assisted Research in the Humanities (CCARH) [18]. It comprises around 3000 files of separate movements from polyphonic fully-encoded music scores by a number of classical composers (including Bach, Beethoven, Handel, and Mozart) of varying keys, textures (i.e. average numbers of notes in a simultaneity) and lengths (numbers of simultaneities). To this basic collection we add, for the purposes of the present paper, three additional sets of polyphonic music data, for a total collection of approximately 3,150 pieces of music. Collectively, we denote these Twinkle, Lachrimae and Folia variations as the TLF sets:

T 26 individual variations on the tune known to English speakers as 'Twinkle, twinkle, little star' (in fact a mixture of mostly polyphonic and a few monophonic versions);

L 75 versions of John Dowland's 'Lachrimae Pavan', collected as part of the ECOLM project (www.ecolm.org) from different 16th and 17th-century sources, sometimes varying in quality (numbers of 'wrong' notes, omissions and other inaccuracies), in scoring (for solo lute, keyboard or five-part instrumental ensemble), in sectional structure and in key;

F 50 variations by four different composers on the well-known baroque tune 'Les Folies d'Espagne'.

8.2 Experiment One: Known Item

The idea for the first experiment comes from a desire to test the robustness of our harmonic modeling. We therefore assembled from the Naxos audio collection the 24 Preludes and Fugues of Book I of Bach's Well-tempered Clavier. The score versions of these piano-based, human-played audio files are present within our source collection, from the CCARH data. So each audio-transcribed Prelude or Fugue becomes a query, and the score from which the audio file was ostensibly played becomes the one "known item" relevant document in the collection.

The question is whether this degraded, transcribed query (Figure 3) can retrieve, at a high rank relative to all other music in the collection, the original "perfect" score (Figure 2). For this particular example, Figure 2 was retrieved at a rank of 1^{st}, from our collection of 3,150 pieces of music.

As good as this result is, accurate evaluation deals with averages to get a true indication of system performance. The results of this experiment are found in Tables 1 and 2. For each set of queries (either the 24 Preludes or 24 Fugues) the known item was retrieved at some rank, where first is the best possible value. These ranks were then averaged across all queries in the set. Results are given for 0^{th} to 2^{nd}-order Markov models, each of which has been smoothed over a window of size $n = 1$ to $n = 4$. For comparison, a system which performed random ranking would place the known item, on average, approximately $1,575^{th}$.

Discussion: Our results show that the known item searches are extremely successful. Through a combination of higher-order Markov models and larger smoothing windows, we were able to retrieve the true symbolic version of the piece using the audio-transcribed, degraded query at an average rank of a little over 3 for the Bach Preludes, and a little over 2 for the Bach Fugues. While there is still room for improvement, it should prove difficult to produce an *average* which is better than 2nd or 3rd.

Though results vary slightly from the Transcription I to the Transcription II algorithms, equally good results were achieved using each. Our harmonic modeling technique is robust enough to handle two significantly different transcription algorithms.

8.3 Experiment Two: Variations

For the second experiment, we wish to determine whether our harmonic modeling approach is useful for retrieving variations on a piece of music, rather than just the original. Recall that in addition to the CCARH data, our source collection contains three sets of variations. For this experiment, the audio version one variation is selected and the score versions of all the variations are judged "relevant" to the audio query, even though their actual *similarity* may vary considerably. A good retrieval system would therefore return all variations toward the top of the 3,150 item list, and all

[2]See also http://ciir.cs.umass.edu/music2000/evaluation.html

non-variations further down. This is repeated for all audio pieces in the set. For example, Figure 4 contains a few of the "Twinkle" variations. When the audio version of Variation 3 is used as the query, we expect not only the score version of Variation 3 to be ranked highly, but the score version of Variation 11 and the score version of the Theme to be ranked highly as well. (The "Theme" is, of course, one of the many variations.)

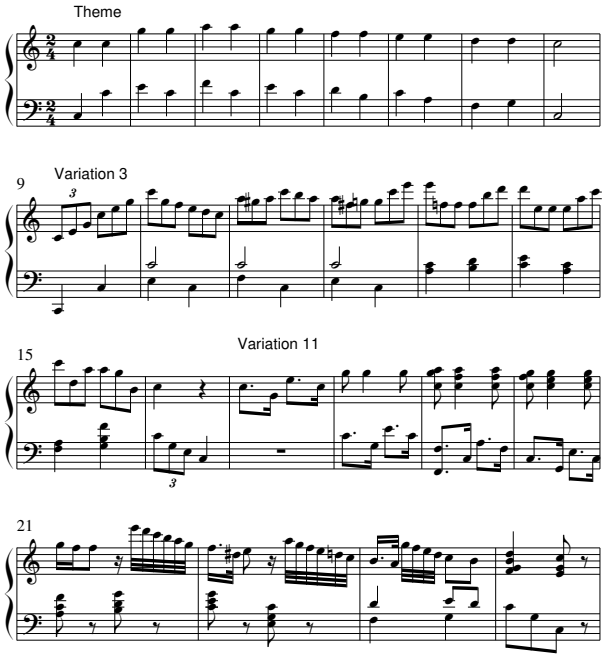

Figure 4: Excerpts from the "Twinkle" variations

Because of the size of these sets and our limited resources, we were not able to get human performances of all these variations. Instead, we converted the queries to MIDI and used a high-quality (30 Megabyte) piano soundfont to create an audio "performance". This apparent weakness in our evaluation is countered by two facts: (1) These audio queries are still polyphonic, even if synthesized, and automatic transcription of overlapping and irregular-duration tones is still quite difficult. (2) Many of the variations on a piece are themselves quite different from a potential query, as we see in Figure 4, and good retrieval is still a difficult task. Even if the perfect score of a variation were used as a query, rather than the imperfect (though perhaps slightly better because of the synthesized audio), quality retrieval is not guaranteed. While we hope to work with a human-produced audio collection for this retrieval experiment someday, as we have done with the known-item Naxos data above, we feel the gist of the evaluation has not been compromised.

Presentation of the known-item results were straightforward. With one relevant document in the entire collection, one need only report the rank (or average rank across all queries) of this document. The problem with multiple relevant documents is how best to visualize the ranked list. Typically this is done using 11-pt interpolated recall-precision graphs, with *precision* (number of relevant documents over total retrieved at a point in the ranked list) given at various level of *recall* (number of relevant documents retrieved over the total number of relevant documents in the query set). However, space constrains us. Instead, we present two values which hopefully characterize the data: mean average precision and mean precision at the top 5 retrieved documents.

Average precision is computed by calculating the precision for a single query (retrieved relevant over total retrieved) every time another variation (relevant document) is found, then averaging over all those points. This score is then averaged over all queries in the set, to create the mean average precision. It is a single value popular in Information Retrieval studies because it allows easy comparison of different systems.

However, some users are more interested in the precision of a system at the top of the ranked list. If the user does not care about finding every single variation but only cares about finding any variation, then the average precision is not as important as the precision at the top of the ranked list. We therefore compute the precision for a single query after retrieving the top 5 documents. If 1 of those documents is relevant (a variation), then the precision is 0.2, or 20%. If none of them are, the precision is 0%. If all of them are, the precision is 100%. We then average this value over all queries in the set, to get the mean precision at the top 5 retrieved documents.

Tables 3 and 4 contain the mean average precision results, while Tables 5 and 6 contain the average precision at the top 5 retrieved documents. These values are given for the three TLF query sets, for 0^{th} to 2^{nd}-order Markov models, each of which has been smoothed over a window of size $n = 1$ to $n = 4$, averaged over all queries in each of the TLF query sets. Unlike the known-item results, where the lower numbers were better because they represented average rank, the values for these variations experiments represent precision. Higher numbers are better.

For each query set we give, as a baseline, the expected value a random ranking algorithm would produce, for a document collection of size and with relevant document count equal to those of the various query sets. For example, the Twinkle set only has 26 variations, so a random ranking of the collection yields a mean precision at the top 5 documents of 0.0077. The Lachrimae set has 75 variations, so it is only natural that with more relevant documents in the collection, a random ranking of those documents will include more relevant documents toward the top of the list. Indeed, the mean precision at 5 docs of the random algorithm on the Lachrimae set is 0.0213.

Discussion: Using an audio-transcribed query to retrieve variations on a piece of music is a much harder problem. We do not consider this a solved problem by any means, but we are encouraged by the results we see. First, it is clear that our harmonic modeling algorithm is doing something correctly, as it yields significant improvement over the random algorithm. Second, we once again see the trend that higher order Markov models and more harmonic smoothing yield better results. Higher and longer does not monotonically indicate better performance, but the trend is nonetheless apparent.

We also note that some query sets are more difficult than others. Not only did we have more success on the Folia variations than on the Twinkle variations, but after listening to the actual pieces, it is clear than human judges would have more difficulty picking out the Twinkle variations than they would the Folia variations. Nevertheless, even for these more difficult Twinkle variations, almost 3 of the 5 top ranked documents are, on average, relevant variations. We feel this is a respectable result.

9. CONCLUSION

It is now clear that retrieval of polyphonic scores using polyphonic audio is possible. By "taking apart" (transcribing) an audio music query and harmonically modeling the musically-salient pitch features we are bridging the gap between audio and symbolic music retrieval, and doing so within the difficult polyphonic domain.

That we have restricted ourselves in this paper to piano (a single timbre) is not a limitation as much as it is an indication of future potential. We did not have to perfectly recognize every single note in a piece of music in order for the harmonic modeling to be successful. Therefore, future audio transcription methods which attempt to

Table 3: Variations Transcription I, Mean Average Precision

Twinkle				Lachrimae				Folia			
	mm0	mm1	mm2		mm0	mm1	mm2		mm0	mm1	mm2
Window 1	0.164	0.130	0.168	Window 1	0.168	0.064	0.033	Window 1	0.375	0.216	0.136
Window 2	0.168	0.163	**0.179**	Window 2	0.168	0.140	0.094	Window 2	0.379	0.365	0.219
Window 3	0.168	0.122	0.131	Window 3	0.164	0.172	0.158	Window 3	0.378	**0.479**	0.334
Window 4	0.172	0.135	0.101	Window 4	0.162	0.179	**0.191**	Window 4	0.384	0.445	0.390
Random = 0.0052				Random = 0.0112				Random = 0.0087			

Table 4: Variations Transcription II, Mean Average Precision

Twinkle				Lachrimae				Folia			
	mm0	mm1	mm2		mm0	mm1	mm2		mm0	mm1	mm2
Window 1	0.145	0.111	0.150	Window 1	0.172	0.056	0.030	Window 1	0.333	0.172	0.105
Window 2	0.145	0.149	**0.156**	Window 2	0.174	0.136	0.096	Window 2	0.337	0.315	0.178
Window 3	0.145	0.095	0.117	Window 3	0.173	0.177	0.162	Window 3	0.331	**0.422**	0.284
Window 4	0.130	0.104	0.083	Window 4	0.172	0.181	**0.195**	Window 4	0.328	0.389	0.329
Random = 0.0052				Random = 0.0112				Random = 0.0087			

Table 5: Variations Transcription I, Precision at top 5 retrieved pieces

Twinkle				Lachrimae				Folia			
	mm0	mm1	mm2		mm0	mm1	mm2		mm0	mm1	mm2
Window 1	**0.592**	0.323	0.462	Window 1	0.496	0.067	0.056	Window 1	0.692	0.104	0.212
Window 2	0.577	0.500	0.515	Window 2	0.501	0.317	0.096	Window 2	0.680	0.444	0.200
Window 3	0.577	0.431	0.485	Window 3	0.477	0.520	0.451	Window 3	0.704	**0.884**	0.544
Window 4	0.585	0.485	0.415	Window 4	0.456	0.531	**0.616**	Window 4	0.740	0.804	0.816
Random = 0.0077				Random = 0.0213				Random = 0.02			

Table 6: Variations Transcription II, Precision at top 5 retrieved pieces

Twinkle				Lachrimae				Folia			
	mm0	mm1	mm2		mm0	mm1	mm2		mm0	mm1	mm2
Window 1	0.485	0.285	0.408	Window 1	0.461	0.040	0.032	Window 1	0.628	0.056	0.112
Window 2	0.515	**0.539**	0.431	Window 2	0.440	0.216	0.059	Window 2	0.672	0.404	0.144
Window 3	0.531	0.346	0.446	Window 3	0.427	0.523	0.419	Window 3	0.628	**0.788**	0.480
Window 4	0.439	0.392	0.331	Window 4	0.440	0.499	**0.619**	Window 4	0.608	0.728	0.732
Random = 0.0077				Random = 0.0213				Random = 0.02			

transcribe the even more difficult polytimbral, polyphonic domain may do so with the confidence that the transcription need not be perfect in order to get good retrieval results.

The same technique which gives us robust, error-tolerant retrieval of known-item queries (Section 8.2) is also useful for retrieving variations (Section 8.3). Indeed, at one level of abstraction, a composed variation can be thought of as an "errorful transcription" of the original piece. Our harmonic modeling approach succeeded in capturing a degree of invariance, a degree of similarity, across such "transcriptions". The technique, though far from perfect, is an important first step for polyphonic (audio and symbolic) music retrieval.

10. FUTURE WORK

We feel one useful direction for this work is to bypass the transcription phase and go directly from audio features to a harmonic description. This will make the modeling phase slightly more difficult, but there might be advantages to bypassing the transcription, as the transcription is only used to create harmonic descriptions. This would bring us closer to some harmonic-recognition work being carried out by others in the pure audio domain such as by Carreras et al [7], or Fujishima [15].

A second direction is to modify the harmonic description smoothing algorithm. We propose in the future to adopt either a (millisecond) time-based or a (rhythmic) beat-based window smoothing approach, rather than the event-based approach we use in this paper. We will sum the harmonic contributions in the way described above across simultaneities within the window in inverse proportion to their time or beat-based distance from the current simultaneity, with additional weightings provided according to metrical stress, note duration or other factors that might be considered helpful. Indeed, harmonic smoothing, properly executed, might be a way of integrating the problematic, not-quite-orthogonal dimensions of pitch and duration within a polyphonic source. Better time-based smoothing might also yield a richer harmonic description, because it gives less weight to transient changes in harmony arising from non-harmonic notes such as passing tones or appoggiaturas.

A third direction deals with passage level retrieval. Rather than modeling entire documents, it might be useful to model portions of documents, particularly if those portions are musically salient. Finally, the issue of standardized test collections remains important. We are interested in participating in such experiments, to compare our system with others that will be developed in the future.

11. ACKNOWLEDGEMENTS

We would like to thank Eleanor Selfridge-Field, Craig Sapp, and Bret Aarden for their patient assistance with the CCARH data, which we used as our primary source collection. We would like to thank Naxos for the use of their Bach Prelude and Fugue audio recordings. Finally, Samer Abdallah deserves credit as an early inspiration for some of the harmonic description assumptions made in this paper.

12. REFERENCES

[1] AMNS. Nightingale music notation software, 2001. http://www.ngale.com.

[2] J. P. Bello, L. Daudet, and M. B. Sandler. Time-domain polyphonic transcription using self-generating databases. In *Proceedings of the 112th Convention of the Audio Engineering Society*, Munich, Germany, May 2002.

[3] J. P. Bello and M. B. Sandler. Blackboard system and top-down processing for the transcription of simple polyphonic music. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFx-00)*, Verona, Italy, December 7-9 2000.

[4] W. Birmingham, R. B. Dannenberg, G. H. Wakefield, M. Bartsch, D. Bykowski, D. Mazzoni, C. Meek, M. Mellody, and W. Rand. Musart: Music retrieval via aural queries. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR)*, pages 73–81, Indiana University, Bloomington, Indiana, October 2001.

[5] W. Birmingham, B. Pardo, C. Meek, and J. Shifrim. The musart music-retrieval system. In *D-Lib Magazine*, February 2002. Available at: www.dlib.org/dlib/february02/02contents.html.

[6] J. Bloch and R. Dannenberg. Real-time accompaniment of polyphonic keyboard performance. In *Proceedings of the 1985 International Computer Music Conference*, pages 279–290, Vancouver, 1985.

[7] F. Carreras, M. Leman, and M. Lesaffre. Automatic harmonic description of musical signals using schema-based chord decomposition. *Journal of New Music Research*, 28(4):310–333, 1999.

[8] C. W. Cleverdon, J. Mills, and M. Keen. *Factors Determining the Performance of Indexing Systems, Volume I - Design, Volume II - Test Results*. ASLIB Cranfield Project, Cranfield, 1966.

[9] S. E. Dixon. On the computer recognition of solo piano music. *Mikropolyphonie*, 6, 2000.

[10] S. Doraisamy and S. M. Rüger. An approach toward a polyphonic music retrieval system. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR)*, pages 187–193, Indiana University, Bloomington, Indiana, October 2001.

[11] M. Dovey. An algorithm for locating polyphonic phrases within a polyphonic piece. In *Proceedings of AISB Symposium on Musical Creativity*, pages 48–53, Edinburgh, April 1999.

[12] R. S. Engelmore and A. J. Morgan. *Blackboard Systems*. Addison-Wesley Publishing, 1988.

[13] N. Fletcher and T. Rossing. *The Physics of Musical Instruments*. Springer Verlag, 1991.

[14] J. Foote. Arthur: Retrieving orchestral music by long-term structure. In *Proceedings of the 1st International Symposium for Music Information Retrieval (ISMIR)*, Plymouth, Massachusetts, October 2000. See http://ciir.cs.umass.edu/music2000.

[15] T. Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *Proceedings of the 1999 International Computer Music Conference*, pages 464–467, Beijing, China, 1999.

[16] A. Ghias, J. Logan, D. Chamberlin, and B. Smith. Query by humming - musical information retrieval in an audio database. In *Proceedings of ACM International Multimedia Conference (ACMMM)*, pages 231–236, San Francisco, CA, 1995.

[17] H. H. Hoos, K. Renz, and M. Görg. Guido/mir - an experimental music information retrieval system based on guido music notation. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR)*, pages 41–50, Indiana University, Bloomington, Indiana, October 2001.

[18] http://www.musedata.org. The musedata collection, 2000. Center for Computer Assisted Research in the Humanities (Stanford, CA).

[19] A. P. Klapuri. Automatic transcription of music. Master's thesis, Tampere Univeristy of Technology, 1998.

[20] C. L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York, 1990.

[21] K. Lemström and J. Tarhio. Searching monophonic patterns within polyphonic sources. In *Proceedings of the RIAO Conference*, volume 2, pages 1261–1278, College of France, Paris, April 2000.

[22] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2001.

[23] M. Marolt. Transcription of polyphonic piano music with neural networks. In *Information technology and electrotechnology for the Mediterranean countries. Vol. 2, Signal and image processing : Proceedings, MEleCon 2000, 10th Mediterranean Electrotechnical Conference*, Cyprus, May 29-31 2000.

[24] K. Martin. A blackboard system for automatic transcription of simple polyphonic music. Technical Report Technical Report No. 385, Perceptual Computing Section, MIT Media Laboratory, 1996.

[25] R. J. McNab, L. A. Smith, D. Bainbridge, and I. H. Witten. The new zealand digital library melody index. In *D-Lib Magazine*, May 1997. Available at: www.dlib.org/dlib/may97/meldex/05witten.html.

[26] D. Meredith, G. Wiggins, and K. Lemström. Pattern induction and matching in polyphonic music and other multi-dimensional datasets. In *the 5th World Multi-Conference on Systemics, Cybernetics and Informatics*, pages 61–66, Orlando, 2001.

[27] G. Monti and M. Sandler. Pitch locking monophonic music analysis. In *Proceedings of the 112th Convention of the Audio Engineering Society (AES)*, Munich, Germany, May 10-13 2002.

[28] J. Pickens. A comparison of language modeling and probabilistic text information retrieval approaches to monophonic music retrieval. In *Proceedings of the 1st International Symposium for Music Information Retrieval (ISMIR)*, October 2000. See http://ciir.cs.umass.edu/music2000.

[29] J. Pickens and T. Crawford. Harmonic models for polyphonic music retrieval. In *Proceedings of the ACM Conference in Information Knowledge and Management (CIKM)*, McLean, Virginia, November 2002.

[30] J. M. Ponte. *A Language Modeling Approach to Information Retrieval*. PhD thesis, University of Massachusetts Amherst, 1998.

[31] H. Purwins, B. Blankertz, and K. Obermayer. A new method for tracking modulations in tonal music in audio data format. citeseer.nj.nec.com/purwins00new.html.

[32] W. Rand and W. Birmingham. Statistical analysis in music information retrieval. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR)*, pages 25–26, Indiana University, Bloomington, Indiana, October 2001.

[33] I. Shmulevich, O. Yli-Harja, E. Coyle, D. Povel, and K. Lemström. Perceptual issues in music pattern recognition - complexity of rhythm and key find. *Computers and the Humanities*, 35(1):23–35, 2001. Appeared also in the Proceedings of the AISB'99 Symposium on Musical Creativity.

[34] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. In J. S. Downie and D. Bainbridge, editors, *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR)*, pages 205–210, Indiana University, Bloomington, Indiana, October 2001.

[35] A. Uitdenbogerd and J. Zobel. Melodic matching techniques for large music databases. In *Proceedings of ACM International Multimedia Conference (ACMMM)*, Orlando Florida, USA, Oct. 1999. ACM, ACM Press.