Music Structure based Vector Space Retrieval

Namunu C. Maddage, Haizhou Li
Institute for Infocomm Research (I^2R)
21, Heng Mui Keng Terrace, Singapore 119613
{maddage, hli}@i2r.a-star.edu.sg

Mohan S. Kankanhalli
School of Computing
National University of Singapore 117543
mohan@comp.nus.edu.sg

ABSTRACT

This paper proposes a novel framework for music content indexing and retrieval. The music structure information, i.e., timing, harmony and music region content, is represented by the layers of the music structure pyramid. We begin by extracting this layered structure information. We analyze the rhythm of the music and then segment the signal proportional to the inter-beat intervals. Thus, the timing information is incorporated in the segmentation process, which we call *Beat Space Segmentation*. To describe *Harmony Events*, we propose a two-layer hierarchical approach to model the music chords. We also model the progression of instrumental and vocal content as *Acoustic Events*. After information extraction, we propose a vector space modeling approach which uses these events as the indexing terms. In *query-by-example* music retrieval, a query is represented by a vector of the statistics of the *n*-gram events. We then propose two effective retrieval models, a hard-indexing scheme and a soft-indexing scheme. Experiments show that the vector space modeling is effective in representing the layered music information, achieving 82.5% top-5 retrieval accuracy using 15-sec music clips as the queries. The soft-indexing outperforms hard-indexing in general.

Categories and Subject Descriptors

H.3.1. [**Information Storage and Retrieval**]: Content Analysis and Indexing - *Indexing methods,* H.3.3 Information Search and Retrieval - *Retrieval models*

General Terms

Algorithms, Performance, Experimentation

Keywords

Music structure, beat space segmentation, harmony event, acoustic event, vector space modeling, *n*-gram,

1. INTRODUCTION

Over the past decades, increasingly powerful technology has made it easier to compress, distribute and store digital media content. There is an increasing demand in tools for automatic indexing and retrieval of music recordings. The task of music retrieval is to rank a collection of music clips according to each one's relevance to a query. In this paper, we are particularly interested in music information retrieval (MIR) for popular songs that are recorded in the raw audio format. In general we aim at providing musicians and scholars tools that search and study different musical pieces of similar music structures (rhythmic structure, melody/harmony structure, music descriptions, etc); help entertainment service providers index and retrieve the songs of similar tones and semantics in response to the user queries in the form of music clips, which is also referred to as *query-by-example*.

The challenges of a MIR system include effective indexing of music information that supports run-time quick search, accurate query representation as the music descriptor, and robust retrieval modeling that ranks the music documents by relevance score. Many MIR systems have been reported in the survey articles [18][24]. MIR research community initially focused on developing text based systems where both database and the query are in the MIDI format and the information is retrieved by matching the melody of query with the database[5][6][10][12][17][19][25]. Since the melody information of both query and song database are text based (MIDI), the research has been devoted to database organization of the music information (monophonic or/and polyphonic nature) and to text-based retrieval models. The retrieval models in those systems includes dynamic programming (DP)[15][22][25], *n*-gram-based matching [5][6][25] and vector space model[17].

Recently, with the advances in information technologies, the community has started looking into developing MIR systems for music in raw audio format. Successful examples towards this research objective includes the *query-by-humming* systems [10][22], which allows a user to input the query by humming a melody line via the microphone. To do so, research efforts have been made to extract the pitch contours from the hummed audio, and to build a retrieval model that measures the relevance between the pitch contour of the query and the melody contours of the intended music signals. Autocorrelation [10], harmonic analysis [22] and statistical modeling via audio feature extraction [21] are some of the techniques that have been employed for extracting pitch contour from hummed queries. In [4][9][10], fixed length audio segmentation, spectral and pitch contour sensitive features are discussed to measure similarity between music clips.

However, the melody-based retrieval model is insufficient for MIR because it is highly possible that different songs share an identical melody contour. The challenge for MIR of music in raw audio format is to represent the music content including harmony/melody, vocal and song structure information holistically.

SIGIR'06, August 6–11, 2006, Seattle, Washington, USA.

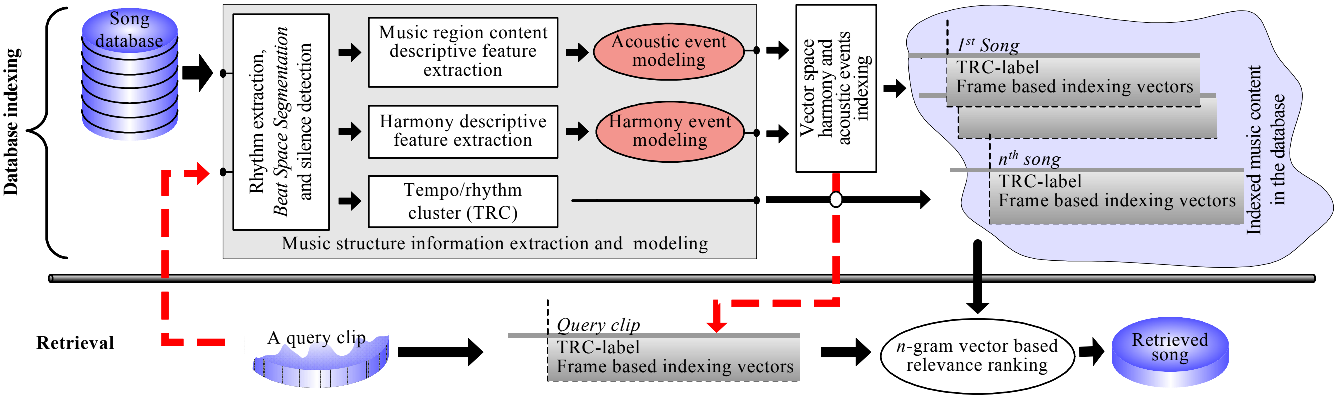
Database indexing
Song database
Rhythm extraction, Beat Space Segmentation and silence detection
Music region content descriptive feature extraction
Acoustic event modeling
Harmony descriptive feature extraction
Harmony event modeling
Tempo/rhythm cluster (TRC)
Music structure information extraction and modeling
Vector space harmony and acoustic events indexing
1st Song
TRC-label
Frame based indexing vectors
nth song
TRC-label
Frame based indexing vectors
Indexed music content in the database
Retrieval
A query clip
Query clip
TRC-label
Frame based indexing vectors
n-gram vector based relevance ranking
Retrieved song

Figure 1: Music structure information extraction, vector space content indexing and retrieval

In this paper, we propose novel indexing and retrieval framework which describes a music signal with a multi-layer representation and it is continuation of our earlier research [15]. We incorporate timing information of the song with the music segmentation process. Then we detect the progression of both music chord and the contents in the music regions to describe the harmony events and the acoustic events respectively. Inspired by the success of vector space modeling in text-based information retrieval, we index and retrieve the songs using vectors of *n*-gram statistics of those events. The proposed framework is illustrated in Figure 1.

This paper is organized as follows. Conceptual music structure pyramid to visualize the information in the music structure is discussed in section 2. Section 3 details the extraction and statistical modeling of layered music information. In Section 4, we propose a vector space modeling framework for MIR with two retrieval models, the hard-indexing and the soft-indexing models. In Section 5, we describe the experiment results. Finally, we conclude in Section 6.

2. MUSIC STRUCTURE

As shown in Figure 2, we represent music information conceptually by a multi-layer pyramid structure.

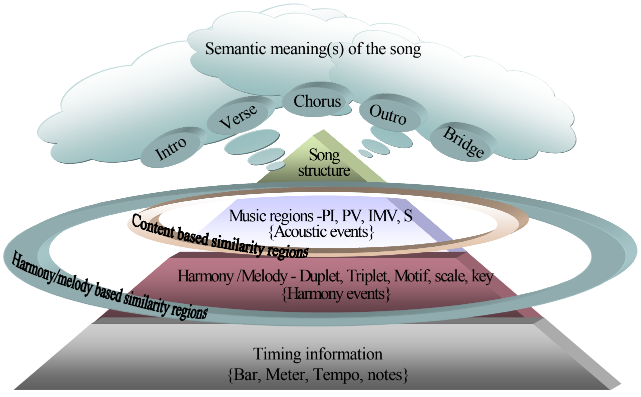
Semantic meaning(s) of the song
Intro
Verse
Chorus
Outro
Bridge
Song structure
Music regions -PI, PV, IMV, S
{Acoustic events}
Content based similarity regions
Harmony /Melody - Duplet, Triplet, Motif, scale, key
{Harmony events}
Harmony/melody based similarity regions
Timing information
{Bar, Meter, Tempo, notes}

Figure 2: Layer wise information representation of music

The 1st layer is the foundation of the pyramid which dictates the timing of a music signal. As time elapses mixing multiple notes together in the polyphonic music, a harmony line is created which is the 2nd layer of music information. Pure instrumental (PI), pure vocal (PV), instrumental mixed vocal (IMV) and silence (S) are the regions that can be seen in a song. PV regions are rare in popular music. Silence regions (S) are the regions which have imperceptible music including unnoticeable noise and very short clicks. The content of the music regions are represented in the 3rd layer. The 4th layer and above depicts the semantics of the song structure, which describes the events or the messages to the audience. Out of all the layers, the most difficult task is to understand the information in the top layer, the semantics of a song from the song structure point of view. In the case of *query-by-example* MIR, we often have a partial clip instead of a full-length song as a query. Therefore, we believe that the lower layer music information is more informative than the top layer as far as MIR is concerned. As such, the top layer information is less critical.

It is noted that popular songs are similar in many ways, for example, *similar beat cycle* – common beat patterns, *similar harmony/melody* - common chord patterns, *similar vocal* – similar lyrics and similar *semantic content* – music pieces or excerpts that creates similar auditory scenes or sensation. In this paper, we will study the retrieval model that evaluates the song similarities in the aspects of beat pattern, melody pattern and vocal pattern.

3. MUSIC INFORMATION MODELING

The fundamental step for audio content analysis is the signal segmentation where the signal within a frame can be considered as quasi-stationary. With quasi-stationary music frames, we can extract features to describe the content and model the features with statistical techniques. The quality of signal segmentation has an impact on system level performance of music information extraction, modeling and retrieval. Like in speech processing, earlier music content analysis [1][3][8] approaches have used fixed length signal segmentation.

A music note can be considered as the smallest measuring unit of the music flow. Usually smaller notes (1/8, 1/16 or 1/32 notes) are played in the bars to align the melody with the rhythm of the lyrics and to fill in the gap between lyrics. Therefore the information within the duration of a music note can be considered quasi-stationary. In this paper we segment the music into frames of the smallest note length instead of fixed length frames. Since the inter-beat interval of a song is equal to the integer multiples of the smallest note, this music framing strategy is called *Beat Space Segmentation (BSS)*. We will discuss the music segmentation in

Section 3.1 that captures timing information (1st layer in Figure 2) of the music structure. In Section 3.2 and 3.3 we will further discuss extraction of harmony and music region content descriptive features that model music information in the 2nd and the 3rd layer.

3.1 Music Segmentation and Silence Detection

We illustrate the proposed onset detection and smallest note length calculation in Figure 3. As highlighted in [15], the spectral characteristics of the music signals are enveloped proportional to octaves. So, we first decompose the music signal into 8 sub-bands whose frequency ranges are shown in Table 1.

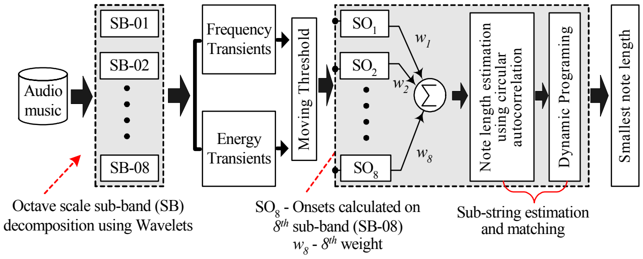
Audio music
SB-01
SB-02
SB-08
Frequency Transients
Energy Transients
Moving Threshold
SO_1
SO_2
SO_8
w_1
w_2
w_8
$\sum$
Note length estimation using circular autocorrelation
Dynamic Programing
Smallest note length
Octave scale sub-band (SB) decomposition using Wavelets
SO_8 - Onsets calculated on 8^{th} sub-band (SB-08)
w_8 - 8^{th} weight
Sub-string estimation and matching

Figure 3: Onset detection and smallest note length calculation

Then the sub-band signals are segmented into 60ms frames with 50% overlap. Both the frequency and energy transients are analyzed using a method similar to that in [7]. We measure the frequency transients in terms of progressive distances in sub-band 01 to 04 because fundamental frequencies (F0s) and harmonics of music notes in popular music are strong in these sub-bands. The energy transients are computed from sub-band 05 to 08.

Table 1: The frequency ranges of the octaves and the sub-bands

Sub-band No	01		02	03	04	05	06	07	08
Octave scale	~B1	C2~B2	C3~B3	C4~B4	C5~B5	C6~B6	C7~B7	C8~B8	C9~B9
Freq-range(Hz)	~64	64~128	128~256	256~512	512~1024	1024~2048	2048~4096	4096~8192	8192~16384

Eq.(1) describes the computation of final onset at time 't', *On(t)* which is the weighted sum of sub-band onsets $SO_r(t)$.

$$On(t) = \sum_{r=1}^{8} w(r).SO_r(t) \qquad (1)$$

The weight matrix w = {0.6, 0.9, 0.7, 0.9, 0.7, 0.5, 0.8, 0.6} has been empirically found to be the best set for calculating dominant onsets in music signals. We run circular autocorrelation over the detected onsets to estimate the inter-beat proportional note length. By varying this estimated note length, we check for patterns of equally spaced intervals between dominant onsets *On(.)* using a dynamic programming approach. The most frequent smallest interval, which is also an integer fraction of other longer intervals, is taken as the smallest note length. Figure 4(a) illustrates the process for a 10-second song clip. The detected onsets are shown in Figure 4(b). The autocorrelation of the detected onsets is shown in Figure 4(c). Inter-beat proportional smallest note level (183.11ms) measure is shown in Figure 4(d). We assume that the tempo of the song is constant. Therefore the starting point of the song is used as the reference point for BSS. Similar steps are followed for computation of the smallest note length in the query song clip. However the first dominant onset is used as the reference point to segment the clip back and forth accordingly. The reference onset is marked in dashed line in Figure 4(b). The smallest note length and its multiples form the tempo/rhythm cluster (TRC). By comparing the TRC of query clip with TRC of the songs in the database, we can narrow down the search space.

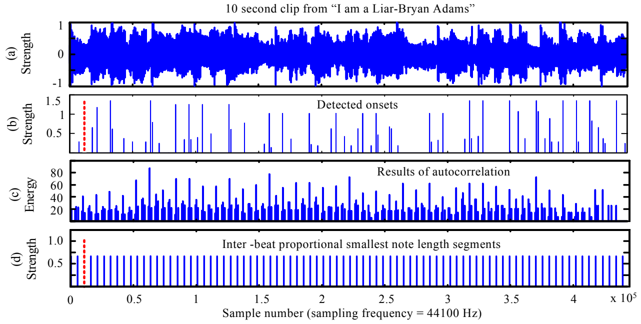

Figure 4: 10 seconds clip of the song

Silence is defined as a segment of imperceptible music, including unnoticeable noise and very short clicks. We use short-time energy function to detect the silent frames.

3.2 Chord Modeling

The progression of music chords describes the harmony event of music. A chord is constructed by playing set of notes (>2) simultaneously. Typically there are 4 chord types (*Major, Minor, Diminished* and *Augmented*) and 12 chords per chord type that can be found in the western music. For efficient chord detection, the tonal characteristics (F0s, harmonics and sub-harmonics) of the music notes which comprise a chord should be well characterized by the feature. Goldstein (1973) [11] and Terhardt (1974) [23] proposed two psycho-acoustical approaches: harmonic representation and sub-harmonic representation, for complex tones respectively. It is noted that harmonics and sub-harmonics of a music note are closely related to the F0 of another note. For example, 3rd and 6th harmonics of note C4 are close to F0 of G5 and G6. Similarly 5th and 7th sub-harmonics of note E7 are closed to F0 of C5 and F#4 respectively.

In our chord detection system, we place 12 filters centered on F0s of 12 notes in each octave covering 8 octaves (C2B2 ~C8B8) to capture the strengths of F0s, sub-harmonics and harmonics. The filter positions are calculated using Eq.(2) which first maps the linear frequency scale (f_{linear}) into octave scale (f_{octave}) where *Fs*, *N*, F_{req} are sampling frequency, number of FFT points and reference mapping point respectively. We set frequency resolution (Fs/N) equal to 1Hz, F_{req}=64Hz (F0 of the note C2) and *C*=12 (12 pitches). The filter (rectangular filter in dashed line) position near note G in both octave and linear frequency axis is depicted in Figure 6.

$$f_{Octave} = \left[C * \log_2 \left(\frac{Fs * f_{linear}}{N * F_{ref}} \right) \bmod C \right] \qquad (2)$$

The reasons for using filters to extract tonal characteristics of notes are explained below.

1. Due to physical configuration of the instruments, the F0s of the notes may vary from the standard values (A4=440Hz is used as the concert pitch).

2. Though the physical octave ratio is 2:1, cognitive experiments have highlighted that this ratio is close at lower frequencies, but increases with the higher frequencies. It exceeds by 3% at about 2 kHz [26]. Therefore, we position the filters to detect the strengths of the harmonics of the shifted notes.

In our experiments, it is found that the tonal characteristics in an individual octave can even effectively represent the music chord. To model these tonal characteristics in the octaves, we propose a 2-layer hierarchical model for music chord (see Figure 5). The models in the 1st layer are trained using pitch class profile (PCP) feature vectors (12-dimensional) which are extracted from individual octaves. Due to poor chord detection accuracy in the C9B9 octave, only C2B2~C8B8 octaves are considered. The construction of PCP vector for n^{th} signal frame and for each octave is explained in Eq.(3). F0 strengths of the α^{th} note and related harmonic and sub-harmonic strengths of other notes are summed up to form the α^{th} coefficient of the PCP vector. In Eq.(3), *S(.)* is the frequency domain magnitude (in dB) signal spectrum. $W_{(OC,\ \alpha)}$ is the filter whose position and the pass-band frequency range varies with both octave index (OC) and α^{th} note in the octave (OC). If the octave index is 1, then the respective octave is C2B2.

$$PCP_{OC}^{n}(\alpha)=\left[S(.)W_{(OC,\alpha)}\right]^{2}\quad OC=1....7,\quad \alpha=1.....12. \qquad (3)$$

The 2nd layer model is trained with the outputs of the 1st layer models which are organized into a feature vector. In our implementation we use 4 Gaussian mixtures for each model in layer 1 and 2. Therefore input vectors to the layer 2 model are probabilistic vectors. This 2-layer modeling can be visualized as first transforming feature space represented tonal characteristics of the music chord into probabilistic space at the layer 1 and then modeling them at layer 2. We use this 2 layer representation to model 48 music chords in our chord detection system.

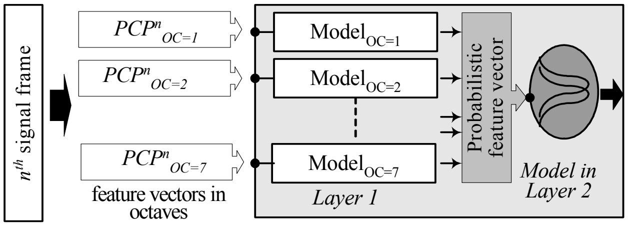

Figure 5: Two layers hierarchical representation of a music chord

3.3 Music Region Content Modeling

As discussed in Section 2, PV, PI, IMV and S are the regions types in a song (3rd layer). However PV regions are comparatively rare in popular music. Therefore both PV and IMV regions are considered as vocal (V) region. In this way, we can just focus on contents of 3 regions (PI, V and S). *Silence* detection has been discussed in Section 3.1.

Sung vocal line carries more descriptive information about the song than other regions. In the PI regions, the extracted feature must be able to capture the information generated by lead instruments (typically the tunes/melody). To this end, we examine Octave scale cepstral coefficient (OSCC) feature and Mel-frequency cepstral coefficient (MFCC) feature for their capabilities to characterize music region content information. MFCC have been highly effective characterizing subjective pitch and the frequency spectrum of speech signals [4]. OSCCs are computed by using a filter bank in frequency domain. Filter positions in the linear frequency scale (f_{linear}) are computed by transforming linearly positioned filters in the octave scale (f_{octave}) to f_{linear} using Eq.(2). We set C=12, F_{ref}=64 Hz in the Eq.(2) so that 12 overlapping rectangular filters are positioned in each octave from C2B2 to C9B9 octave (64 ~ 16384) Hz. The Hamming shape of filter/window has sharp attenuation and it suppresses valuable information in the higher frequencies nearly by 3 fold as compared to the rectangular shape filter [4]. Therefore, a rectangular filter is better than Hamming filter for music signal analysis because they are wide band signals compared to speech signals. Figure 6 depicts octave to linear filter position transformation. The output *Y(b)* of the b^{th} filter is computed according to Eq.(4) where *S(.)* is the frequency spectrum in decibel (dB), $H_b(.)$ is the b^{th} filter, and m_b and n_b are boundaries of b^{th} filter.

$$Y(b)=\sum_{a=m_b}^{n_b} S(a)H_b(a) \qquad (4)$$

Eq.(5) describes the computation of β^{th} cepstral coefficient where k_b, N_f and Fn are center frequency of the b^{th} filter, number of frequency sampling points and number of filters respectively (Fn=12 in our case).

$$C(\beta)=\frac{2}{\beta}\sum_{b=1}^{Fn} Y(b)\cos(k_b\frac{2\pi}{N_f}\beta) \qquad (5)$$

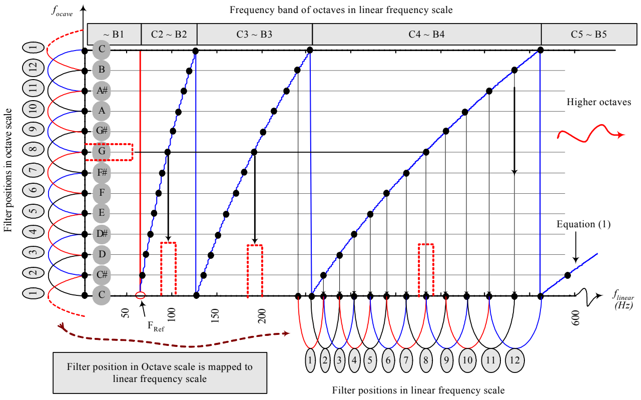
f_{ocave}
Frequency band of octaves in linear frequency scale
~ B1
C2 ~ B2
C3 ~ B3
C4 ~ B4
C5 ~ B5
Filter positions in octave scale
1
2
3
4
5
6
7
8
9
10
11
12
1
C
C#
D
D#
E
F
F#
G
G#
A
A#
B
C
Higher octaves
Equation (1)
50
100
150
200
600
F_{Ref}
f_{linear} (Hz)
Filter position in Octave scale is mapped to linear frequency scale
Filter positions in linear frequency scale

Figure 6: Transformation of octave scale filter positions to linear frequency scale

Singular values (SVs) indicate the variance of the corresponding structure. Comparatively high singular values describe the number of dimension in which the structure can be represented orthogonally. Smaller singular values indicate the correlated information in the structure and considered to be noise. We perform singular value decomposition (SVD) over feature matrices extracted from PI and V regions.

Figure 7 shows the normalized singular value variation of 20 OSCCs and 20 MFCCs extracted from both PI and V regions of a Sri Lankan Song “Ma Bala Kale (මා බාල කාලේ)”. We use 96 filters for calculating MFCCs and OSCCs. It can be seen that singular values of OSCCs are higher than of MFCCs for both PV and PI frame. The average of 20 singular values per OSCCs for PV and PI frames are 0.1294 and 0.1325. However, for MFCC, they are as lower as 0.1181 and 0.1093 respectively. As shown in Figure 7, the singular values are in descending order with respect to the ascending coefficient numbers. The average of the last 10 singular values of OSCCs is nearly 10% higher than those of MFCCs, which means the last 10 OSCCs are less correlated than the last 10 coefficients of MFCCs. Thus we can conclude that the OSCCs are less correlated than MFCCs in representing content of music regions.

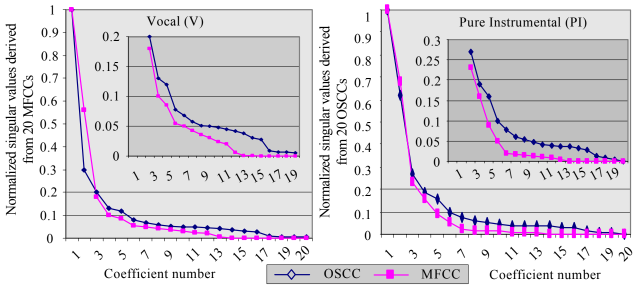

Figure 7: Singular values from OSCCs and MFCCs for PV and PI frames. The frame size is a quarter note length (662ms)

4. MUSIC INDEXING AND RETRIEVAL

Unlike text document that uses words or phrases as indexing terms, a music signal is a continuous digital signal without obvious anchors for indexing. The challenges of indexing music signal are two fold. First, what would be good indexing anchors; second, what would be good representation of music contents for indexing and retrieval. In Section 3, we have discussed the statistical modeling of music information in a multi-layer paradigm as illustrated in Figure 2. Layer-wise information representation allows us to describe a music signal quantitatively in a descriptive data structure. Next, we will propose two indexing terms i.e. harmony event and acoustic event to describe the information in the 2nd and 3rd layers of the music structure pyramid.

4.1 Harmony Event and Acoustic Event

Progression of music chords describes the *Harmony Event*. In Section 3.2, we explained sub-band PCP feature extraction and a 2-layer hierarchical chord modeling. Note that it is relatively easy to detect beat spacing in a music signal. A beat space is a natural choice as a music frame, and thus the indexing resolution of a music signal. Suppose that we have trained 48 frame-based chord models, as shown in Figure 5 (4 chord types *Major, Minor, Diminish and Augmented* in combination with 12 chords each type). Each chord model describes a frame-based harmony event which can serve as the indexing term. One can think of music as a chord sequence, with each chord spanning over multiple frames. A chord model space $\Lambda = \{C_i\}$ can be trained on a collection of chord-labeled data. We use the HTK 3.3 toolbox for training such a 2-layer chord model space. At run-time, a music frame O_n is recognized and converted to a harmony event $\hat{I}_h$, and a music signal is therefore tokenized into a chord sequence.

$$\hat{I}_h = \arg\max_{c} p(o_n \mid c_i) \quad i = 1,\ldots,48 \qquad (6)$$

Pure instrumental (PI) and the vocal (V) regions contain the descriptive information about the music content of a song. A song can be thought of as a sequence of interwoven PI and V events, that we call *Acoustic Events*. We extract 20 OSCC features from each music frame. We train two Gaussian Mixture models (GMMs) of 64 mixtures. Each GMM is trained on a collection of features from one of the two events. We define the frame-based acoustic events as another type of indexing term in parallel with harmony events. Suppose we denote r_1 for PI and r_2 for V event. They are trained from a labeled database. At run-time, a music frame O_n is recognized and converted to a V or PI event $\hat{I}_r$. Eq.(6) and (7) can be seen as the chord and acoustic event decoders.

$$\hat{I}_r = \arg\max_{i=1,2} p(o_n \mid r_i) \qquad (7)$$

We index the contents in silence regions (S) with zero observation. Inspired by the idea in text categorization where we use lexical words as indexing terms to form a document vector for a text document, we attempt to use the events as indexing terms to design a vector for a music segment. Next let us formulate the indexing and retrieval problem cast in the vector space model framework. We will study two retrieval models, hard-indexing and soft-indexing *n*-gram MIR models.

4.2 *n*-gram Vector

The harmony and acoustic decoders serve as the tokenizers for music signal. The tokenization process results in two synchronized streams of events, a chord and an acoustic sequence, for each music signal. An event is represented by a tokenization symbol. They are represented in a text-like format. It is noted that *n*-gram statistics has been used in many natural language processing tasks to capture short-term substring constraints such as letter *n*-gram in language identification [2] and spoken language identification [14]. If we think of the chord and acoustic tokens, as the letters of music, then a music signal is a document of chord/acoustic transcripts. Similar to the letter *n*-gram in text, we can use the token *n*-gram of music as the indexing term, which aims at capturing the short-term syntax of musical signal. The statistics of token themselves represent the token unigram.

Vector space modeling (VSM) has become a standard tool in text-based IR systems since its introduction several decades ago [20]. It uses a vector to represent a text document. One of the advantages of the method is that it makes partial matching possible. We can derive the distance between documents easily as long as the vector attributes are well defined characteristics of the documents. Each coordinate in the vector reflects the presence of the corresponding attribute, which is typically a term. A chord/acoustic token in a music signal is just like a term in a document. Inspired by the idea of VSM in text-based IR, we propose using a vector to represent a music segment. If a music segment is thought of as an article of chord/acoustic tokens, then the statistics of the presence of the tokens or token n-grams describe the content of the music.

Suppose that we have a token sequence, $t_1t_2t_3t_4$. We derive the unigram statistics from the token sequence itself. We derive the bigram statistics from $t_1(t_2)$ $t_2(t_3)$ $t_3(t_4)$ $t_4(\#)$ where the acoustic vocabulary is expanded over the token's right context. Similarly, we derive the trigram statistics from the $t_1(\#,t_2)$ $t_2(t_1,t_3)$ $t_3(t_2,t_4)$ $t_4(t_3,\#)$ to account for left and right contexts. The # sign is a place holder for free context. In the interest of manageability, we only use up to bigrams. In this way, for an acoustic vocabulary of $|c|$=48 token entries in the chord stream, we have 48 unigram frequency items f_n^i in the chord vector $\overrightarrow{f_n} = \{f_n^1,\ldots,f_n^i,\ldots,f_n^{48}\}$ as in Figure 8. f_n^i is equal to 1, if t_n=c_i otherwise it is *0*.

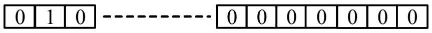

Figure 8: A count vector representation of a music frame

Similarly we have 2 unigram frequency items in the acoustic vector for the acoustic stream. For simplicity, we only formulate $\overrightarrow{f_n}$ next. To capture the short-term dynamics, we can easily derive the bigram representation for two consecutive frames. As such, we build a chord bigram vector of [48x48=2304] dimensions,

$\overrightarrow{f_n'} = \{f_n^{1,1},...,f_n^{i,j},...,f_n^{48,48}\}$ where if both f_n^i=1 and f_{n+1}^j=1, then $f_n^{i,j}$ =1; otherwise $f_n^{i,j}$ =0. Similarly an acoustic bigram vector of [2x2=4] dimensions can be formed. For a music segment of N frames, we construct a chord unigram vector $\overrightarrow{f_N} = \{f_N^1,...,f_N^i,...,f_N^{48}\}$ by aggregating the frame vectors with the i^{th} element as

$$f_N^i = \sum_{n=1}^N f_n^i \qquad (8)$$

We can construct its chord bigram vector of [48x48=2304] dimensions $\overrightarrow{f_n'} = \{f_N^{1,1},...,f_N^{i,j},...,f_N^{48,48}\}$ in a similar way with the $(i,j)^{th}$ element as

$$f_N^{i,j} = \sum_{n=1}^N f_n^{i,j} \qquad (9)$$

The acoustic vector can be formulated in a similar way with a 2-dimensional vector for unigram and [2x2=4] dimensional vector for bigram. Figure 9 shows schematically how an *n*-gram vector is constructed using N frames of unigram vector and how the relevance score is evaluated between a query and a music segment.

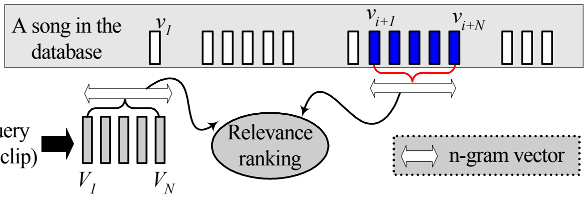

Figure 9: The music database is indexed by *n*-gram vector. Each harmony/acoustic event is associated with an indexing vector. The similarity between a query and a music segment in the database is measured for relevance ranking.

Although we use 2-dimensional coordinate for the bigram count, the vector can be treated as a 1-dimensional array. The process of deriving unigram and bigram vectors for a music segment involves minimum computation. In practice, we can compute those vectors at run-time directly from the chord/acoustic transcripts resulting from the tokenization. Note that the tokenization process compares a music frame against all the chord/acoustic events at a higher computational cost. It can be done off-line.

Following the text-based IR process, the MIR process computes the similarity between a query music segment and all the candidate music segments. For simplicity, let $\overrightarrow{f_N^i}(q)$ denote the chord unigram vector (48 dimensions) and $\overrightarrow{f_N^{i,j}}(q)$ denote the chord bigram vector (2304 dimensions) for a query of N frames. Similarly, a chord unigram vector $\vec{f}_N^i(d)$ and a chord bigram vector $\overrightarrow{f_N^{i,j}}(d)$ can be obtained from any segment of N frames in the music database. The similarity between two *n*-gram vectors can be defined as

$$s(\vec{f}_N^i(q),\vec{f}_N^i(d)) = \frac{\vec{f}_N^i(q)\cdot\vec{f}_N^i(d)}{|\vec{f}_N^i(q)|\times|\vec{f}_N^i(d)|} \qquad (10)$$

$$s(\vec{f}_N^{i,j}(q),\vec{f}_N^{i,j}(d)) = \frac{\vec{f}_N^{i,j}(q)\cdot\vec{f}_N^{i,j}(d)}{|\vec{f}_N^{i,j}(q)|\times|\vec{f}_N^{i,j}(d)|} \qquad (11)$$

With Eq.(10) and Eq.(11), we can rank the music segments by their relevance. The relevance is can be defined by the fusion of unigram and bigram similarity scores.

4.3 Expected Frequency of *n*-gram

Although it would be convenient to derive the term count from token sequences derived from a music query, we find that the tokenization is affected by many factors. For example, the tokenization does not always produce identical token sequence for two similar music segments. The difference could be due to the variation in beat detection, variation of music productions between the query and the intended music. The inconsistency between the tokenization of the query and the intended music produce an undesired mismatch as far as MIR is concerned. Assuming that the numbers of beats in query and music are detected correctly, the inconsistency is characterized by substitutions of tokens between the desired label and the tokenization results. If a token is substituted, then it presents a mismatch between the query and the intended music segment. To address this problem, we propose using the tokenizers as *probabilistic machines* that generate a *posteriori* probability for each of the chord and acoustic events. If we think of the *n*-gram counting as integer counting, then the *posteriori* probability can be seen as *soft-hits* of the events. For brevity, we only formulate the *soft-hits* for chord vector. According to Bayes' rule, we have

$$p(c_i \mid o_n) = \frac{f(o_n \mid c_i)p(c_i)}{\sum_i f(o_n \mid c_i)p(c_i)} \qquad (12)$$

where $p(c_i)$ be the prior probability of the event c_i. Assuming no prior knowledge about the events, $p(c_i)$ can be dropped from Eq.(12), which is then simplified as

$$p(c_i \mid o_n) = \frac{f(o_n \mid c_i)}{\sum_i f(o_n \mid c_i)} \qquad (13)$$

Let $P(c_i|o_n)$ be denoted as p_n^i. It can be interpreted as the expected frequency of event c_i at n^{th} frame, with the following properties, (a) $0\leq p_n^i \leq 1$, (b) $\sum_{i=1}^{48} p_n^i = 1$. A frame is represented by a vector of continuous values as illustrated in Figure 8, which can be thought of a soft-indexing approach as opposed to the hard-indexing approach for music frame using n-gram counting. The soft-indexing reflects how a frame is represented by the whole model space while the hard-indexing estimates the n-gram count based on the top-best tokenization results. We have good reason to expect soft-indexing to provide higher resolution vector representation for a music frame.

0.04	0.75	0.01	--------	0.07	0.01	0.02	0.01	0.03	0.06	0.00

Figure 10: An expected frequency vector for a music frame

Assuming the music frames are independent of each other, the joint *posteriori* probability of two events i and j between two frames, n^{th} and $(n+1)^{th}$ can be estimated as

$$p_n^{i,j} = p_n^i \times p_{n+1}^j \qquad (14)$$

where $p_n^{i,j}$ has properties similar to that of p_n^i, (a) $0 \leq p_n^{i,j} \leq 1$, (b) $\sum_{i=1}^{48}\sum_{j=1}^{48} p_n^{i,j} = 1$. For a query of N frames, the expected frequency of unigram and bigram can be estimated as

$$E\{f_N^i\} = \sum_{n=1}^N p_n^i \qquad (15)$$

$$E\{f_N^{i,j}\} = \sum_{n=1}^N p_n^{i,j} \qquad (16)$$

Thus the soft-indexing vector for query and matching music segment are $E\{\vec{f}_N^i(q)\}$ and $E\{\vec{f}_N^i(d)\}$ respectively. Replacing

$\overrightarrow{f_N^i}(q)$ with $E\{\overrightarrow{f_N^i}(q)\}$, $\overrightarrow{f_N^i}(d)$ with $E\{\overrightarrow{f_N^i}(d)\}$ in Eq.(12) and Eq.(13), the similar relevance scores can be used for soft-indexing ranking.

5. EXPERIMENTS

We first study the chord and acoustic modeling performance. Then we carry out MIR experiments. We established a 300 song database DB1 (44.1 kHz sampling rate, 16 bits per sample, mono channel) extracted from music CDs for MIR experiments. Songs in DB1 are sung by 20 artists as listed in Table 2, each on average contributing 15 songs. The tempos of the songs are in the rage of 60~180 beats per minute

Table 2: The artists in the song database

Female Artists		Male Artists	
01. Agnetha Faitskog	06. Kathryn Williams	11. Ben Jelen	16. Michael Bolton
02. Celine Dion	07. Madonna	12. Bryan adams	17. Michael Jackson
03. Cranberries	08. Mandy Moore	13. Cliff Richard	18. MLTR
04. Dido	09. Mariah Carey	14. Elton John	19. Richard Marx
05. Faith Hill	10. Shania Twain	15. Justin Timberlake	20. West life

5.1 Harmony Event Modeling

Harmony events are described by the progression of music chords. Each of the 48 chord models is a 2-layer representation of Gaussian mixtures (see Figure 5) and is trained with annotated samples in a chord database (CDB). The CDB includes recorded chord samples from original instruments (string type, bow type, blowing type, etc) as well as synthetic instruments (software generated). In addition, the CDB also includes chord samples extracted from 40 English songs (a subset of DB1), with the aid of music sheets and listening tests. Therefore we have around 10 minutes of each chord sample spanning from C2 to B8. 70% of the samples of each chord are used for training and the rest 30% for testing in cross validation setup. Experimental results are shown in Figure 11.

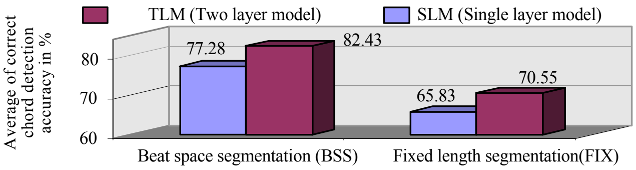

Figure 11: Average correct chord detection accuracy

The results of the proposed 2-layer model (TLM) are compared with single layer model (SLM). Single layer chord model is constructed using 128 Gaussian mixtures. General PCP features vectors *(GPCP)* are used for training and testing the SLMs. Eq.(17) explains the computation of α^{th} coefficient of the *GPCP* feature vectors.

$$GPCP^n(\alpha) = \sum_{OC=1}^{7} PCP^n_{OC}(\alpha) \qquad \textbf{(17)}$$

It is noted that the proposed TLM with feature extracted from BSS outperforms the SLM approach by 5% in absolute accuracy.

5.2 Acoustic Event Modeling

We compare performance of OSCCs and MFCCs for modeling regions PI and V. SVD analysis depicted in Figure 7 highlights that OSCCs characterize music content more uncorrelatedly than MFCCs. In this experiment, we selected 100 English songs (10 songs per artist and 5 artists per gender) from DB1. We annotate the V and PI regions. Each V and PI class information is then modeled with 64 GMs. 100 songs are used by cross validation where 60/40 songs are used as training/testing in each turn. Table 3 shows correct region detection accuracies for optimized number of both the filters and coefficients of MFCC and OSCC features. We report the correct detection accuracy for PI-region and V-region, when the frame size is equal to both beat space and fixed length (30ms). Both OSCC and MFCC perform better when the frame size is beat space. As OSCC outperforms MFCC in general, we use it for modeling acoustic events.

Table 3: Correct Average classification of PI and V regions

Feature	No. of filters	No. of coefficients	PI(%) -BSS	V(%)-BSS	Avg(PI+V) %-FIX
OSCC	96	20	83.78	81.32	74.65
MFCC	36	24	77.91	76.54	71.11

5.3 Music Information Retrieval

In DB1, we select 4 clips of 30-second music as queries from each artist in the database, totaling 80 clips. Out of 4 clips, two clips belong to V region and other two mainly belongs to PI region. For a given query, the relevance score between a song and the query is defined as the sum of the similarity score between the top K most similar indexing vectors and the query vector. Typically, we set K to be 30.

After computing the smallest note length in the query, we check the tempo/rhythm clusters of the songs in the data base. For song relevance ranking, we only consider the songs whose smallest note lengths are in the same range (with ±30ms tolerance) as the smallest note length of the query or integer multiples of them. Then the surviving songs in the DB1 are ranked according to their respective relevance scores. Figure 12 shows the average accuracy of the correct song retrieval when the query length is varied from 2-sec to 30-sec. Both chord events and acoustic events are considered for constructing *n*-gram vectors.

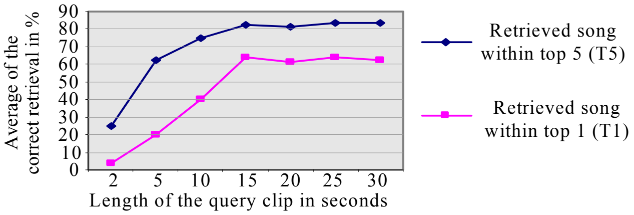

Figure 12: Average retrieval accuracy of songs

The average accuracy of correct song retrieval in top choice is around 60% for the query length varies fro 15 ~30-sec. For the similar query lengths, the retrieval accuracy for top-5 candidates is improved by 20%. In Table 4 we study the chord event effect and the combined effect of chord and acoustic events on the retrieval accuracy.

Table 4: Effects of chord and acoustic event information in MIR

Avg accuracy in % (15sec Query length)	T5		T1	
	Harmony event - I_h	I_h+ acoustic event	Harmony event - I_h	I_h+ acoustic event
Soft indexing	61.25	82.5	38.75	63.75
Hard indexing	48.75	73.75	32.5	55.0

It can be found that soft-indexing outperforms hard-indexing (see Eq.(8), Eq.(9)). In general, combining acoustic events and chord events yields a better performance. This can be understood by the fact that similar chord patterns are likely to occur in different songs. The acoustic content helps differentiate one from the other.

6. DISCUSSION AND CONCLUSION

We have proposed a novel framework for MIR. We visualize music information (timing, harmony and music region contents) in the form of a music structure pyramid. We incorporate timing information in the beat space segmentation of music signal. A two-layer hierarchical chord model has been proposed to describe the harmony events. Content progression of instrumental and vocal regions has also been modeled to describe acoustic events. After modeling layered music information in the vector space, we explored two retrieval models, hard-indexing and soft-indexing.

Our experiments show that octave scale music information modeling followed by the inter-beat interval proportion segmentation is more efficient than with the fixed length music segmentation. We found the soft-indexing retrieval model is more effective than the hard-indexing one. The fusion of chord model and acoustic model statistics improves retrieval accuracy effectively. The overall experimental results convince our focus on layer-wise music processing and vector space retrieval model are promising research directions. We find that music information in different layers complements each other to achieve an improved MIR performance. The robustness in this retrieval modeling framework depends on how well the information is extracted. We will continue to focus on the extraction of uncorrelated music information.

Even though music retrieval is the targeted application in this paper, the proposed vector space music modeling framework is useful for developing many other applications such as music summarization, streaming, music structure analysis, and creating multimedia documentary using music semantics. In the future, we will work on extending it to other relevant applications.

7. REFERENCES

[1] Berenzweig, A., Logan, B., Ellis, D.P.W., and Whitman, B. A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures. In *Computer Music Journal,* Summer, 2004, 63-74.

[2] Cavnar, W.B., and Trenkle, J.M. N-Gram-Based Text Categorization. In *Proc*. of 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994.

[3] Chai, W., Vercoe, B. Structure Analysis of Music Signals for Indexing and Thumbnailing. In *Proc. of the ACM/IEEE JCDL,* May 2003.

[4] Deller, J. R., Hansen, J.H.L., and Proakis, H. J. G. *Discrete-Time Processing of Speech Signals,* IEEE Press, 2000.

[5] Doraisamy, S., and Rüger, S. Robust Polyphonic Music Retrieval with N-Grams. In *Journal of Intelligent Information Systems*. Vol 21, No. 1. pp 53-70, 2003.

[6] Downie, J.S., and Nelson, M. Evaluating a Simple Approach to Music Information Retrieval Method. In *Proc. ACM SIGIR,* July 2000.

[7] Duxburg. C, Sandler. M., and Davies. M. A Hybrid Approach to Musical Note Onset Detection. In *Proc. Int. Conf. DAFx.* Hamburg, Germany, Sept, 2002.

[8] Foote, J. Visualizing Music and Audio Using Self-Similarity. In *Proc. ACM MM*, Oct, 1999.

[9] Fujishima, T. Real Time Chord Recognition of Musical Sound: A System Using Lisp Music. In *Proc. ICMC,* Oct. 1999.

[10] Ghias, A., Logan, J., Chamberlin, D., and Smith, B. C. Query by Humming: Musical Information Retrieval in an Audio Database. In *Proc. of ACM MM,* Nov, 1995.

[11] Goldstein, J. L. An Optimum Processor Theory for the Central Formation of the Pitch of Complex Tones. In *JASA,* Vol. 54, 1973.

[12] Kageyama, T., Mochizuki, K., and Takashima, Y. Melody Retrieval with Humming. In *Proc. ICMC,* Sept, *1993.*

[13] Lemström, K., and Laine, P. Music Information Retrieval using Musical Parameters. In *Proc. of the ICMC,* Oct, 1998.

[14] Ma, B., and Li, H., A Phonotactic-Semantic Paradigm for Automatic Spoken Document Classification. In *Proc. of ACM SIGIR*, Aug, 2005.

[15] Maddage C. N., Xu, C., Kankanhalli, M.S., and Shao, X, Content-based Music Structure Analysis with the Applications to Music Semantic Understanding, In *ACM Multimedia Conference*, Oct. 2004.

[16] McNab, R.J., Smith, L.A., Witten, I.H., Henderson, C.L., and Cunningham, S.J. Towards the Digital Music Library: Tune Retrieval from Acoustic Input. In *Proc. ACM Digital Libraries,* March, 1996.

[17] Melucci, M., and Orio, N. Music Information Retrieval using Melodic Surface. In *Proc. ACM Digital Libraries,* Aug, 1999

[18] Pickens, J. *A Survey of Feature Selection Techniques for Music Information Retrieval.* Technical report, Center of Intelligent Information Retrieval, Dept. of Computer Science, University of Massachusetts, 2001.

[19] Pickens, J. and Iliopoulos, C. Markov Random Fields and Maximum Entropy Modeling for Music Information Retrieval. In *Proc. of ISMIR,* Sept, 2005.

[20] Salton, G. *The SMART retrieval system.* Prentice-Hall, Englewood Cliffs, NJ, 1971.

[21] Shih, H.-H., Narayanan, S. S., and Kuo, C.-C. J. An HMM-Based Approach to Humming Transcription. In *Proc. of ICME,* Aug, 2002.

[22] Song, J., Bae, S. Y., and Yoon, K. Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System. In *Proc. of ISMIR,* Oct, 2002.

[23] Terhardt, E. Pitch, Consonance and Harmony. In *JASA,* Vol. 55, No. 5, 1974.

[24] Typke, R., Wiering, F., and Veltkamp, R. A Survey of Music Information Retrieval Systems. In *Proc. of the ISMIR*, Sept. 2005.

[25] Uitdenbogerd, A. L., and Zobel, J. An architecture for effective music information retrieval. In *Journal of the American Society for Information Science and Technology*, Vol. 55, No. 12, pp. 1053-1057, 2004.

[26] Ward, W. Subjective Music Pitch. In *JASA,* Vol. 26, 195