

Content-based Music Structure Analysis with Applications to Music Semantics Understanding

Namunu C Maddage^{1,2}, Changsheng Xu¹, Mohan S Kankanhalli², Xi Shao^{1,2}

¹Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613

{maddage, xucs, shaoxi}@i2r.a-star.edu.sg

²School of Computing
National University of Singapore
Singapore 117543
mohan@comp.nus.edu.sg

ABSTRACT

In this paper, we present a novel approach for music structure analysis. A new segmentation method, beat space segmentation, is proposed and used for music chord detection and vocal/instrumental boundary detection. The wrongly detected chords in the chord pattern sequence and the misclassified vocal/instrumental frames are corrected using heuristics derived from the domain knowledge of music composition. Melody-based similarity regions are detected by matching sub-chord patterns using dynamic programming. The vocal content of the melody-based similarity regions is further analyzed to detect the content-based similarity regions. Based on melody-based and content-based similarity regions, the music structure is identified. Experimental results are encouraging and indicate that the performance of the proposed approach is superior to that of the existing methods. We believe that music structure analysis can greatly help music semantics understanding which can aid music transcription, summarization, retrieval and streaming.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing – *abstract methods, indexing methods.*

General Terms

Algorithms, Performance, Experimentation

Keywords

Music structure, melody-based similarity region, content-based similarity region, chord, vocal, instrumental, verse, chorus

1. INTRODUCTION

The song structure generally comprises of Introduction (Intro), Verse, Chorus, Bridge, Instrumental and Ending (Outro). These sections are built upon the melody-based similarity regions and content-based similarity regions. Melody-based similarity regions are defined as having similar pitch contours constructed from the chord patterns. Content-based similarity regions are defined as the

regions which have both similar vocal content and melody. Corresponding to the music structure, the Chorus sections and Verse sections in a song are considered to be the content-based similarity regions and melody-based similarity regions respectively.

The previous work on music structure analysis focuses on feature-based similarity matching. Goto [10] and Bartsch [1] used pitch sensitive chroma-based features to detect repeated sections (i.e. chorus) in the music. Foote and Cooper [7] constructed a similarity matrix and Cooper [4] defined a global similarity function based on extracted mel-frequency cepstral coefficients (MFCC) to find the most salient sections in the music. Logan [14] used clustering and hidden Markov model (HMM) to detect the key phrases in the choruses. Lu [15] estimated the most repetitive segment of the music clip based on high level features (occurrence frequency, energy and positional weighting) calculated from MFCC and octave-based spectral contrast. Xu [24] used an adaptive clustering method based on the features (linear prediction coefficients (LPC) and MFCCs) to create music summary. Chai [3] characterized the music with pitch, spectral and chroma based features and then analyzed the recurrent structure to generate a music thumbnail.

Although some promising accuracies are claimed in the previous methods, their performances are limited due to the fact that music knowledge has not been effectively exploited. In addition, these approaches have not addressed a key issue: the estimation of the boundaries of repeated sections is difficult unless rhythm (time signature TS, tempo), vocal / instrumental boundaries and Key (room of the pitch contour) of the song are known.

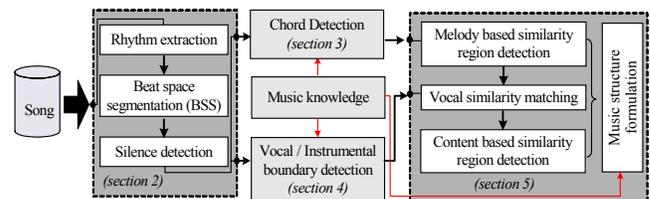


Figure 1: Music structure formulation

We believe that the combination of bottom-up and top-down approaches, which combines the complementary strength of low-level features and high-level music knowledge, can provide us a powerful tool to analyze the music structure, which is the foundation for many music applications (see section 7). Figure 1 illustrates the steps of our novel approach for music structure formulation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'04, October 10-16, 2004, New York, New York, USA.

Copyright 2004 ACM 1-58113-893-8/04/0010...\$5.00.

1. Firstly, the rhythm structure of the song is analyzed by detecting note onsets and the beats. The music is segmented into frames where the frame size is proportional to the inter-beat time length. We call this segmentation method as beat space segmentation (BSS).
2. Secondly, we employ a statistical learning method to identify the chord in the music and detect vocal/instrumental boundaries.
3. Finally, with the help of repeated chord pattern analysis and vocal content analysis, we define the structure of the song.

The rest of the paper is organized as follows. Beat space segmentation, chord detection, vocal/instrumental boundary detection, and music structure analysis are described in section 2, 3, 4, and 5 respectively. Experimental results are reported in section 6. Some useful applications are discussed in section 7. We conclude the paper in section 8.

2. BEAT SPACE SEGMENTATION

From the signal processing point of view, the song structure reveals that the temporal properties (pitch/melody) change in inter-beat time intervals. We assume the time signature (TS) to be 4/4, this being the most frequent meter of popular songs, and the tempo of the song to be constrained between 30-240 M.M (Mälzel’s Metronome: the number of quarter notes per minute) and almost constant [19]. Usually smaller length notes (eighth or sixteenth notes) are played in the bars to align the melody with the rhythm of the lyrics and fill the gap between lyrics. Thus segmenting the music into the smallest note length (i.e. eighth or sixteenth note length) frames instead of conventional fixed length segmentation in speech processing is important to detect the vocal/instrumental boundaries and the chord changes accurately. In section 2.1, we describe how to compute the smallest note length after detecting the onsets. This inter-beat time proportional segmentation is called beat space segmentation (BSS).

2.1 Rhythm extraction and Silence detection

Rhythm extraction is the first step of beat space segmentation. Our proposed rhythm extraction approach is shown in Figure 2. Since the music harmonic structures are in octaves [17] (Figure 5), we decompose the signal into 8 sub-bands whose frequency ranges are shown in Table 1. The sub-band signals are segmented into 60ms windows with 50% overlap and both the frequency and energy transients are analyzed using the similar method to that in [6].

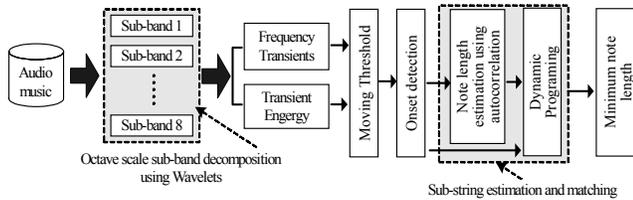


Figure 2: Rhythm tracking and extraction

We measure the frequency transients in terms of progressive distances between the spectrums 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	Higher Octaves
Freq-range (Hz)	0 ~ 64	64 ~ 128	128 ~ 256	256 ~ 512	512 ~ 1024	1024 ~ 2048	2048 ~ 4096	4096 ~ 8192	(8192 ~ 22050)

In order to detect dominant onsets in a song, we take the weighted summation of onsets, detected in each sub-band as described in Eq. (1). $On(t)$ is the sum of onsets detected in all eight sub-bands $Sb_i(t)$ at time ‘ t ’ in the music. In our experiments, the weight matrix $w = \{0.6, 0.9, 0.7, 0.9, 0.7, 0.5, 0.8, 0.6\}$ is empirically found to be the best set for calculating dominant onsets to extract the inter-beat time length and the length of the smallest note (eighth or sixteenth note) in a song.

$$On(t) = \sum_{i=1}^8 w(i).Sb_i(t) \quad (1)$$

Both the inter-beat length and the smallest note length are initially estimated by taking the autocorrelation over the detected onsets. Then we employ a dynamic programming [16] approach to check for patterns of equally spaced strong and weak beats among the detected dominant onsets $On(t)$, and compute both inter-beat length and the smallest note length.

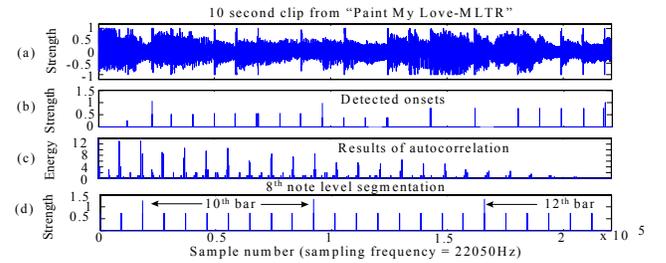


Figure 3: 10 seconds clip of the song

Figure 3(a) illustrates a 10-second song clip. The detected onsets are shown in Figure 3(b). The autocorrelation of the detected onsets is shown in Figure 3(c). Both the eighth note level segmentation and bar measure are shown in Figure 3(d). The eighth note length is 418.14ms

Silence is defined as a segment of imperceptible music, including unnoticeable noise and very short clicks. We use the short-time energy function to detect silent frames [24].

3. CHORD DETECTION

Chord detection is essential for identifying melody-based similarity regions which have similar chord patterns. Detecting the fundamental frequencies (F0s) of notes which comprise the chord is the key idea to identify the chord. We use a learning method similar to that in [20] for chord detection. Chord detection steps are shown in Figure 4. The Pitch Class Profile (PCP) features, which are highly sensitive to F0s of notes, are extracted from training samples to model the chord with HMM.

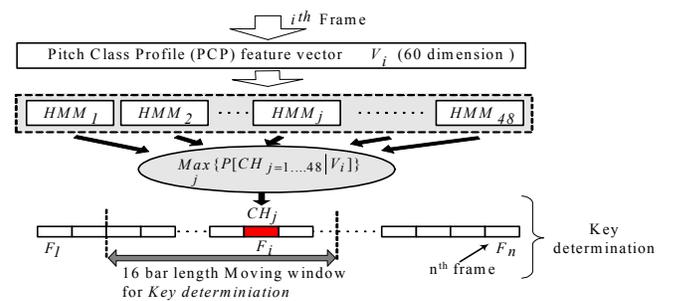


Figure 4: Chord detection and correction via Key determination

The polyphonic music contains the signals of different music notes played at lower and higher octaves. Some instruments like those of the string type have a strong 3rd harmonic component

[17] which nearly overlaps with the 8th semitone of next higher octave. This is problematic in lower octaves and it leads to wrong chord detection. For example, the 3rd harmonic of note C3 and F0 of note G4 nearly overlap (Table 1). To overcome such situations, in our implementation, music frames are first transformed into frequency domain using FFT with 2Hz frequency resolution (i.e. [sampling frequency-Fs / number of FFT points-N] \approx 2Hz). Then, the value of C in Eq. (2), which maps linear frequencies into the octave scale, is set to 1200, where the pitch of each semitone is represented with as high resolution as 100 cents[10]. We consider 128~8192Hz frequency range (sub-band 02 ~ 07 in Table 1) for constructing the PCP feature vectors to avoid adding percussion noise, i.e. base drums in lower frequencies below 128 Hz and both cymbal and snare drums in higher frequencies over 8192Hz, to PCP features. By setting F_{ref} to 128 Hz, the lower frequencies can be eliminated. The initial 1200-dimensional $PCP_{INT}(\cdot)$ vector is constructed based on Eq. (3), where $X(\cdot)$ is the normalized linear frequency profile, computed from the beat space segment using FFT.

$$p(k) = \left[C * \log_2 \left(\frac{Fs * k}{N * F_{ref}} \right) \right] \bmod C \quad (2)$$

$$PCP_{INT}(i) = \sum_{k:p(k)=i} |X(k)|^2 \quad i = 1, 2, \dots, 1200 \quad (3)$$

In order to obtain a good balance between computational complexity and efficiency, the original 1200 dimension of the PCP feature vector is reduced to 60. Thus each semitone is represented by summing 100 cents into 5 bins according to Eq. (4).

$$PCP(p) = \sum_{i=20(p-1)+1}^{20*p} PCP_{INT}(i) \quad p = 1, 2, 3, \dots, 60 \quad (4)$$

Our chord detection system consists of 48 continuous density HMMs to model 12 Major, 12 Minor, 12 Diminished and 12 Augmented chords. Each model has 5 states including entry and exit and 3 Gaussian Mixtures for each hidden state. The mixture weights, means and covariances of all GMs and initial and transition state probabilities are computed using the Baum-Welch algorithm [25]. Then the Viterbi algorithm [25] is applied to find the efficient path from starting to the end state in the models.

3.1 Error correction in the detected chords

The pitch difference between the notes of chord pairs (Major chord & Augmented chord and Minor chord & Diminished chord) are small. In our experiments, we sometimes find that the observed final state probabilities of HMMs corresponding to these chord pairs are high and close to each other. This may lead to wrong chord detection. Thus we apply a rule-based method (key determination) to correct the detected chords and then apply heuristic rules based on popular music composition to further correct the time alignment (chord transition) of the chords.

The key is defined by a set of chords. Song writers sometimes use relative Major and Minor key combinations in different sections, perhaps minor key for Middle eight and major key for the rest, which would break up the perceptual monotony effect of the song [21]. However, songs with multiple keys are rare. Therefore a 16-bar length window is run over the detected chords to determine the key of that section as shown in Figure 4. The key of that section is the one to which a majority of the chords belong. The 16-bar length window is sufficient to identify the key [21]. If Middle eight is present, we can estimate the region where it appears in the song by detecting the key change. Once the key is determined, the error chord is corrected as follow:

- First we normalize the observations of the 48 HMMs representing 48 chords according to the highest probability observed from the error chord.
- The error chord is replaced by the next highest observed chord which belongs to the same key and its observation is above a certain threshold (TH_{chord}).
- Replace the error chord with the previous chord, if there is no observation which is above the TH_{chord} and belongs to the chords of the same key.

$TH_{chord}=0.64$ is empirically found to be good for correcting chords. The music signal is assumed to be quasi-stationary between the inter-beat times, because the melody transition occurs on beat time. Thus we apply the following chord knowledge [11] to correct the chord transition within the window.

- Chords are more likely to change on beat times than on other positions.
- Chords are more likely to change on half note times than on other positions of beat times.
- Chords are more likely to change at the beginning of the measures (bars) than at other positions of half note times.

4. VOCAL BOUNDARY DETECTION

Even if the melodies in the choruses are similar, they may have different instrumental setup to break the perceptual monotony effect in the song. For example, the 1st chorus may contain snare drums with piano and the 2nd chorus may progress with bass and snare drums with rhythm guitar. Therefore after detecting melody-based similarity regions, it is important to analyze the vocal contents of these regions to decide which regions have similar vocal content. The melody-based similarity regions which have similar vocal content are called content-based similarity regions. Content-based similarity regions correspond to the choruses in the music structure. The earlier works on singing voice detection [2], [13] and instrument identification [8] have not fully utilized music knowledge as explained below.

- The dynamic behavior of the vocal and instrumental harmonic structures is in octaves.
- The frame length within which the signal is considered as quasi stationary is the note length [17].

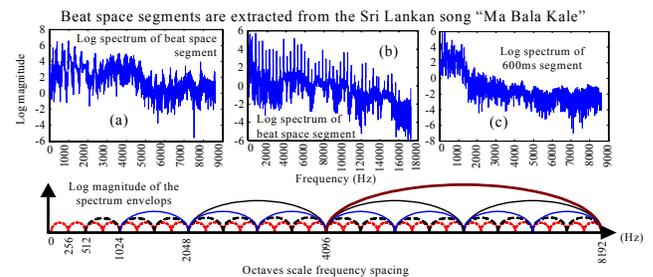


Figure 5: Top figures, (a) - Quarter note length (662ms) guitar mixed vocal music; (b) - Quarter note length (662ms) instrumental music (mouth organ); (c) - Fixed length (600ms) speech signal. Bottom figure - Ideal octave scale spectral envelope.

The music phrases are constructed by lyrics according to the time signature. Thus in our method we further analyze the BSS frames to detect the vocal and instrumental frames. Figure 5 (top) illustrates (a) the log spectrums of beat space segmented piano mixed vocals, (b) mouth organ instrumental music, (c) and log spectrum of fixed length speech. The analysis of harmonic

structures extracted from BSS frames indicates that the frequency components in the spectrums (a) and (b) are enveloped in octaves. The ideal octave scale spectral envelopes are shown in Figure 5 (bottom). Since the instrumental signals are wide band signals (up to 15 kHz), the octave spectral envelopes in instrumental signals are wider than those in vocal signals. However the similar spectral envelopes cannot be seen in the spectrum of speech signal.

Thus we use the “Octave Scale” instead of the Mel scale to calculate Cepstral coefficients [12] to represent the music content. These coefficients are called Octave Scale Cepstral coefficients (OSCC). In our approach, we divide the whole frequency band into 8 sub-bands (the first row in Table 1) corresponding to the Octaves in music. Since the useful range of fundamental frequencies of tones produced by music instruments is considerably less than the audible frequency range, we position triangular filters over the entire audible spectrum to accommodate the harmonics (overtones) of the high tones.

Table 2: Number of filters in sub-bands

Sub-band No	01	02	03	04	05	06	07	08
No of filters	6	8	12	12	8	8	6	4

Table 2 shows the number of triangular filters which are linearly spaced in each sub-band and empirically found to be good for identifying vocal and instrumental frames. It can be seen that the number of filters are maximum in the bands where the majority of the singing voices are present for better resolution of the signal in that range. Cepstral coefficients are then extracted from the Octave Scale using Eq. (5) & (6) to characterize music content, where N , N_{cb} , and n are the number of frequency sample points, critical band filters and Cepstral coefficients respectively [12].

$$Y(i) = \sum_{j=m_i}^{n_i} \log |S_i(j)| |H_i(j)| \quad (5)$$

$$C(n) = \frac{2}{N} \sum_{i=1}^{N_{cb}} Y(i) \cos\left(k_i \frac{2\pi}{N} n\right) \quad (6)$$

Figure 6 illustrates the deviation of the 3rd Cepstral coefficient derived from Mel and Octave scales for pure instrumental (PI) and instrumental mixed vocal (IMV) classes of a song. The frame size is quarter note length (662ms) without overlap. The number of triangular filters used in both scales is 64. It can be seen that the standard deviation is lower for the coefficients derived from the Octave scale, which makes it more robust for our application.

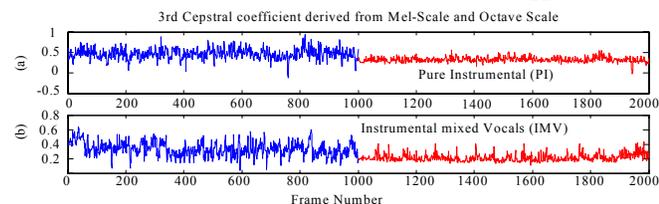


Figure 6: The 3rd Cepstral coefficient derived from Mel-scale (1~1000 frame) and Octave scale (1001~2000 frames).

Singular value decomposition (SVD) is applied to find the uncorrelated Cepstral coefficients for both Mel and Octave scales. We use the order range from 20-29 coefficients and from 10-16 coefficients respectively for both Mel scale and Octave scale. Then we train support vector machine [5] with radial based kernel function (RBF) to identify the PI and IMV frames.

4.1 Error correction of detected frames

The instrumental notes often connect with words at the beginning, middle or end of the music phrase in order to maintain the flow of words according to the melody contour. Figure 7 illustrates the

error correction for misclassified vocal/instrumental frames. Here we assume the frame size is the eighth note length.

- The Intro of a song is instrumental and the error frames can be corrected according to Figure 7(a) where the length of the Intro is X bars.
- The phrases in the popular music are typically 2 or 4 bars long [22] and the word/lyrics are more likely to start at the beginning of the bar than at the second half note in the bar. Thus in Figure 7(a), the number of instrumental frames at the beginning of the 1st phase of Verse 1 can be either zero or four ($Z = 0$ or 4)
- Figure 7 (b) illustrates the corrections of instrumental frames in the instrumental section (INST). The INST begins and ends at the beginning of the bar.

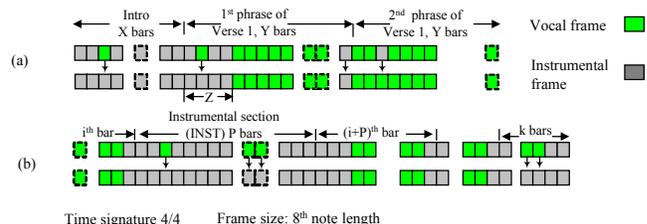


Figure 7: Correction of instrumental/vocal frames

5. MUSIC STRUCTURE ANALYSIS

In order to detect the music structure, we first detect melody-based and content-based similarity regions in the music and then apply the knowledge of music composition to detect the music structure.

5.1 Melody-based similarity region detection

The melody-based similarity regions have the same chord patterns. Since we cannot detect all the chords without error, the region detection algorithm should have tolerance to errors. For this purpose, we employ Dynamic Programming for approximate string matching [16] as our melody-based similarity region detection algorithm.

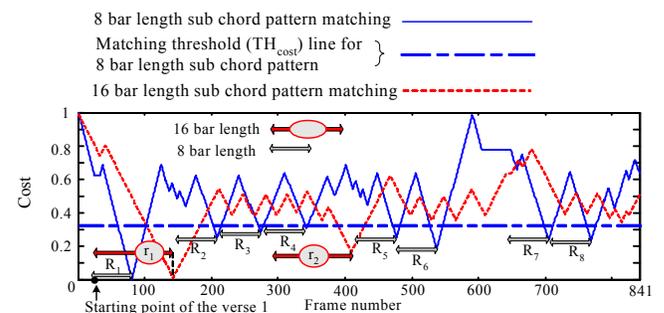


Figure 8: 8 and 16 bar length chord pattern matching results

Figure 8 illustrates the matching results of both 8 and 16 bar length chord pattern matching from the beginning of the Verse 1 in the song “Cloud No 9 – Bryan Adams”. Y-axis denotes the normalized cost of matching the pattern and X-axis represents the frame number. We set threshold TH_{cost} and analyze the matching cost below the threshold to find the pattern matching points in the song. The 8-bar length regions ($R_2 \sim R_8$) have the same chord pattern as the first 8-bar chord pattern (R_1 -Destination Region) in Verse 1. When we extend the Destination Region to 16 bars, only

r_2 region has the same pattern as r_1 where r_2 is the first 16 bars from the beginning of the Verse 2 in the song.

5.2 Content-based similarity region detection

Content-based similarity regions are the regions which have similar lyrics and more precisely they are the choruses regions in the song. The melody-based similarity regions R_i and R_j can further be analyzed to detect whether these two regions are content-based similarity regions, by following steps.

Step 1: The beat space segmented vocal frames of two regions are first sub-segmented into 30 ms with 50% overlapping sub-frames. Although two choruses have both similar vocal content (lyrics) and melody, the vocal content may be mixed with different set of instrumental setup. Therefore, to find the vocal similarity, it is important that the extracted features from the vocal content of the regions should be sensitive only to the lyrics but not to the instrumental line mixed with the lyrics. Figure 9 illustrates the variation of the 9th coefficient of OSCC, MFCC and LPC features for three words ‘clue number one’ which are mixed with notes of rhythm guitar. It can be seen that OSCC is more sensitive to the syllables in the lyrics than MFCC and LPC. Thus we extract 20 coefficients of OSCC feature per sub-frame to characterize the lyrics in the region R_i and R_j

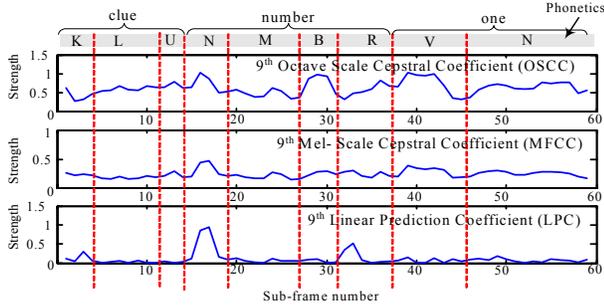


Figure 9: The response of the 9th OSCC, MFCC and LPC to the Syllables of the three words ‘clue number one’. The number of filters used in OSCC and MFCC are 64 each. The total number of coefficients calculated from each feature is 20.

Step 2: The distances between feature vectors of R_i and R_j are computed. The Eq. (7) explains how the k^{th} distance $dist(k)$ is computed between the k^{th} feature vectors V_i and V_j in the regions R_i and R_j respectively. The ‘ n ’ distances calculated from the region pair R_i and R_j are summed up and divided by ‘ n ’ to calculate the “ $dissimilarity(R_i, R_j)$ ”, which gives lower value for the content-based similarity region pairs as shown in Eq. (8).

$$dist_{n_i n_j}(k) = \frac{|V_i(k) - V_j(k)|}{|V_i(k)| * |V_j(k)|} \quad i \neq j \quad (7)$$

$$dissimilarity(R_i, R_j) = \frac{\sum_{k=1}^n dist_{n_i n_j}(k)}{n} \quad (8)$$

Step 3: To overcome the pattern matching errors due to detected error chords, we shift the regions back and forth in one bar step and the maximum size of the shift is 4 bars. Then repeat Step 1 & 2 to find the positions of the regions which give the minimum value for “ $dissimilarity(R_i, R_j)$ ” in Eq. (8).

Step 4: Compute “ $dissimilarity(R_i, R_j)$ ” in all region pairs and normalize them. By setting a threshold (TH_{smlr}) such that the region pairs below the TH_{smlr} are detected as content-based similarity regions implying that, they belong to chorus regions. Based on our experimental results $TH_{smlr} = 0.389$ gives good

performance. Figure 10 illustrates the calculated content-based similarity regions between melody-based similarity region pairs which are found in Figure 8 for the song “Cloud No 9 – Bryan Adams”. It is obvious that the dissimilarity is very high between R_1 which is the first 8-bar length of the Verse 1 and other regions. Therefore, if R_1 is the first 8-bar region of the Verse 1, the similarity between R_1 and other regions is not compared in our algorithm.

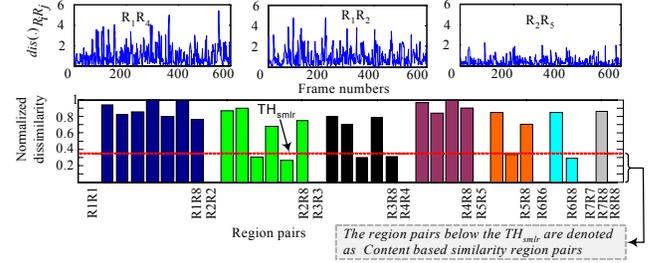


Figure 10: The normalized content-based similarity measure between regions ($R_1 \sim R_8$) computed from melody-based similarity regions of the song as shown in Figure 8 (Red dash line)

5.3 Structure formulation

The structure of the song is detected by applying heuristics which agree with most of the songs. Typical song structure follows the verse–chorus pattern repetition [22], as shown below.

- (a) Intro, Verse 1, Chorus1, Verse 2, Chorus2, Chorus3, Outro
- (b) Intro, Verse 1, Verse 2, Chorus1, Verse 3, Chorus2, *Middle eight or Bridge*, Chorus3, Chorus4, Outro

Following constraints are considered for music structure analysis:

- The minimal number of choruses and verses that appears in a song is 3 and 2 respectively. The maximal number of verses that appears in a song is 3.
- Verse and chorus are 8 or 16 bars long.
- All the verses in the song share the similar melody and all the choruses also share the similar melody. Generally the verse and chorus in the song does not share the same melody. However, in some songs the melody of chorus may be partially or fully similar to the melody of the verse.
- In a song, the lyrics of all verses are quite different, but the lyrics of all the choruses are similar.
- The length of the Bridge is less than 8 bars.
- The length of Middle Eight is 8 or 16 bars

5.3.1 Intro detection

Since the Verse 1 starts at the beginning of either the bar or the second half note in the bar, we extract the instrumental section till the 1st vocal frame of the Verse 1 and detect that section as Intro. If silent frames are present at the beginning of the song, they are not considered as part of the intro because they do not carry a melody.

5.3.2 Verses and Chorus detection

The end of the Intro is the beginning of Verse 1. Thus we can detect Verse 1 if we know whether it is of length 8 or 16 bars and then detect all the melody-based similarity regions. Since the minimum length of the verse is 8 bars, we find the melody-based similarity regions (MSR) based on the first 8-bar chord pattern of the Verse 1 according to the method in section 5.1. We assume the 8-bar MSRs are $R_1, R_2, R_3 \dots R_n$ in a song where n is the

number of MSRs. The Cases 1 & 2, describe how to detect the boundaries of both the verses and the choruses when the number of MSRs is ' ≤ 3 ' and '>3'.

Case 1: $n \leq 3$

The melodies of the verse and chorus are different in this case.

Verse boundary detection: To decide whether the length of the verse is 8 or 16 bars, we further detect the MSRs based on the first 16-bar chord pattern extracted from the starting of the Verse 1. If the detected number of 16-bar MSRs is same as the earlier detected 8-bar MSRs (i.e. n), then the verse is of 16 bars length. Otherwise it is 8-bars long.

Chorus boundary detection: Once the verse boundaries are detected, we check the gap between the last two verses. If the gap is more than 16 bars, the length of the chorus is 16 bars otherwise 8 bars. Since the chorus length is computed, we find the chorus regions in the song according to section 5.1. The verse chorus repetition patterns [(a) or (b)] imply that the Chorus 1 appears between the last two verses and bridge may appear between the 2nd last verse and the Chorus 1. Thus we assume that the Chorus 1 ends at the beginning of the last verse and then MSRs are found based on the chord pattern of the approximated Chorus 1. In order to find the exact boundaries of the choruses we use content-based similarity measure (see section 5.2) between the detected chorus regions.

- We compute the dissimilarity of Chorus 1 and other estimated chorus regions based on step 1, 2, and 3 in section 5.2. We sum all the dissimilarities as Sum_dissm (0) where 0 is the zero shift.
- Then we shift the chorus backward by one bar and re-compute Sum_dissm (-1B), where -1B is 1-bar backward shift.
- Repeat shifting and computing Sum_dissm () till Chorus 1 comes to the end of the 2nd last verse.
- The position of Chorus 1 which gives the minimum value for Sum_dissm () defines the exact chorus boundaries.

Case 2: $n > 3$,

The melodies of the chorus and verse are partially or fully similar in this case. It can be seen from Figure 8 that there are 8 MSRs detected with 8-bar length verse chord pattern.

- First we compare content-based similarities among all the regions except R_1 based on step 1, 2, 3 and 4 in section 5.2. The region pairs of dissimilarities (Eq. (8)) that are lower than TH_{smlr} are the 8-bar length chorus sections.
- If the gap between R_1 and R_2 is more than 8 bars, the verse is 16 bars and based on the 16-bar Verse 1 chord pattern we find other verse regions.
- If a found verse region overlaps with a earlier detected 8-bar chorus region, the verse region is not considered as verse.
- Once the verse regions are found we can detect the chorus boundaries in a way similar to that of Case 1.

5.3.3 Instrumental sections (INST) detection

The Instrumental section may have a melody similar to the chorus or verse. Therefore, the melody-based similarity regions which have only instrumental music are detected as INSTs. However some INSTs have a different melody. In that case, we run a window of 4 bars to find regions which have INSTs (see point 3 in section 4.1).

5.3.4 Bridge and Middle eighth detection

The length of the Bridge is less than 8 bars long. The Middle eighth is 8 or 16 bars long and it appears in pattern (b). Once the boundaries of verses, choruses and INSTs are defined, the appearance of Bridges can be found by checking the gaps between these regions. If the song follows the pattern (b), we check the gap between Chorus 2 and Chorus 3 to see whether they are 8 or 16 bars long and contain vocal frames. If the gaps are less than 8 bars and contain vocal frames, they are detected as the bridge. Otherwise they are detected as Middle eighth.

5.3.5 Outro detection

From the song patterns [(a) & (b)], it can be seen that before the outro there is a chorus. Thus we detect Outro based on the length between both the end of the final chorus and the song.

6. EXPERIMENTAL RESULTS

Our experiments are conducted using 40 popular English songs (10- MLTR, 10 – Bryan Adams, 6 – Beatles, 8 –Westlife, and 6 – Backstreet Boys). The original keys and chord timing of the song are obtained from a commercially available music sheet. All the songs are first sampled at 44.1 kHz with 16 bits per sample and stereo format. Then we manually annotate the songs to identify the timing of vocal/instrumental boundaries, chord transitions and song structure. The following subsections explain both the performance and the evaluation results of rhythm extraction, chord detection, vocal/instrumental boundary detection and music structure detection.

6.1 Rhythm extraction and silence detection

To compute the average length of the smallest note which is seen in the song, we test the first 30, 60 and 120 seconds of the song. Our system manages to detect the smallest note length of 38 songs correctly implying a 95% accuracy with 30ms error margin. The 30ms error margin is set because in the rhythm tracking system the windows are of 60ms each and they are 50% overlapped with each other. Then we set the frame size equal to the smallest note length and segment the music. The frames which have normalized short time energies below a threshold (TH_s) are detected as *silence* frames. TH_s set to 0.18 in our experiments.

6.2 Chord detection

We use HTK tool box [25] to model 48 chord types with HMM modeling. The feature extraction and model configuration of HMMs are explained in section 3. 40 songs are used by cross validation, where 30/10 songs are used as training/testing in each turn. In addition to the song training chords, over 6 minutes of each chord sample spanning from C3 to B6 has been used for HMM training. Chord data are generated from original instruments (Piano, bass guitar, rhythm guitar etc) and synthetic instruments (Roland RS- 70 synthesizer, cakewalk software). The reported average frame-based accuracy of chord detection is 79.48%. We manage to determine the correct key of all the songs. Therefore the 85.67% of frame-based accuracy is achieved after error correction with key information.

6.3 Vocal/instrumental boundary detection

The SVM Torch II [5] is used to classify frames into vocal or instrumental class and similar classifier training and testing procedures described in section 6.2 are applied to evaluate the accuracy. In Table 3, the average frame-based classification accuracy of OSCCs is compared with the accuracy of MFCCs. It

is empirically found that both the number of filters and coefficients of the features give the best performance in classifying instrumental frames (PI) and vocal frames (PV-Pure vocals, IMV).

Table 3: Correct classification for vocal and instrumental classes

Feature	No of filters	No of coefficients	PI (%)	IMV+PV (%)
OSCC	64	12	84.82	80.97
MFCC	36	24	78.43	75.32

We compare the performance of SVM with GMM. Since GMM is considered as a one state HMM, we use the HTK tool box [25] to setup GMM classifiers for both vocal and instrumental class. It is experimentally found that 62 and 48 Gaussian mixtures, which respectively model vocal and instrumental classes, give the best classification performances. Figure 11 compares the frame-based classification accuracies of SVM and GMM classifiers before and after the rule based error corrections. It can be seen that SVM performs better than GMM. The classification accuracy can be significantly improved by 2.5-5.0% after applying rule based error correction scheme to both vocal and instrumental classes.

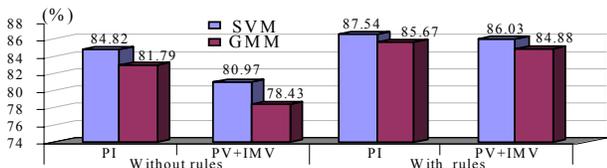


Figure 11: Comparison between SVM and GMM without rules and with rules.

6.4 Intro/verse/chorus/bridge/Outro detection

We evaluate the results of detected music structure in two aspects.

- How accurately are all the parts in the music identified? For example, if 2/3 of the choruses are identified in the song, the accuracy of identifying the choruses is 66.66%.
- How accurately are the sections detected? In Eq. (9), the accuracy of detecting the section is explained. For example, if the accuracies of detecting 3 chorus sections in the song are 80.0%, 89.0% and 0.0%, then the average accuracy of detecting chorus section in the song is $(80+89+0)/3 = 56.33\%$.

$$\text{Detection accuracy of a section (\%)} = \frac{\text{length of correctly detected section}}{\text{correct length}} * 100 \quad (9)$$

In Table 4, the accuracy of both identification and detection of the structural parts in the song “Cloud No 9 – Bryan Adams” is reported. Since the song has 3 choruses and they are identified, 100% accuracy is achieved in identification of chorus sections in the song. However the average correct length detection accuracy of the chorus is 99.74%.

Table 4: Evaluation of identified and detected parts in a song

Parts in the song	I	V	C	INST	B	O
Number of parts	1	2	3	1	1	1
Number of parts identified	1	2	3	1	1	1
Individual accuracy of parts identification (in %)	100	100	100	100	100	100
Average detection accuracy (in %)	100	100	99.74	99.26	98.88	100

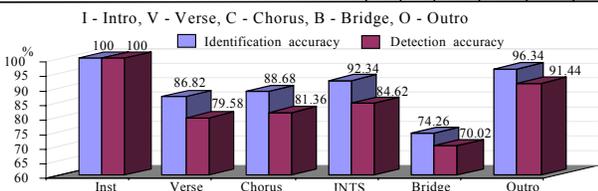


Figure 12: The average detection accuracies of different sections

Figure 12 illustrates our experimental results for average detection accuracy of different sections. It can be seen that Intro (I) and the Outro (O) have been detected with very high accuracy. But for Bridge (B) section the detection accuracy is the lowest.

Using our test data set, we compare our method with previous method described in [10]. For both chorus identification and detection, 69.57% and 72.34% are the respective accuracies reported by the previous method whereas we achieved over 80% accuracy for both identification and detection of the chorus sections. This comparison reveals that our method is more accurate than the previous method.

7. APPLICATIONS

Music structure analysis is essential for music semantics understanding and is useful in various applications, such as music transcription, music summarization, music information retrieval and music streaming.

Music transcription: Rhythm extraction and vocal/instrumental boundary detection are the preliminary measures for both lyrics identification and music transcription. Since music phrases are constructed with rhythmically spoken lyrics [18], rhythm analysis and BSS can be used to identify the word boundaries in the polyphonic music signal (see Figure 9). The signal separation techniques can further be applied to reduce the signal complexity within the word boundary to detect the voiced/unvoiced regions. These steps simplify the *lyrics identification* process. The content based signal analysis helps to identify the possible instrumental signal mixture within the BSS. The chord detection extracts the pitch/melody contour in the music. These are the essential information for *music transcription*.

Music summarization: The existing summary making techniques [1], [3], [15], [24] face the difficulty in both avoiding content repetition in the summary and correctly detecting the content-based similarity regions (i.e. chorus sections) which they assume to be the most suitable section as music summary. Figure 13 illustrates the process for generating music summary based on the structural analysis. The summary is created with the chorus, which is melodically stronger than the verse [22] and the music phrases are included anterior or posterior to selected chorus to get the desired length of the final summary. The rhythm information is useful for aligning musical phrases such that the generated summary has smooth melody.

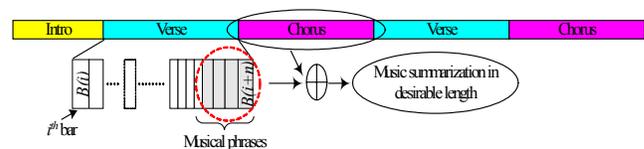


Figure 13: Music summarization using music structure analysis

Music information retrieval (MIR): In most of MIR by humming systems, a F0 tracking algorithms are used to parse a sung query for melody content [9]. However these algorithms are not efficient due to complexity of the polyphonic nature of the signals. To make the MIR in real sound recording more practical, it is required the extract information from different sections such as instrumental setup, rhythm, melody contours, key changes and multi-source vocal information in the song. In addition, the low-level vector representation of non-repeated music scenes/events is useful for achieving songs in music databases for information retrieval because it reduces both the memory storage and retrieval

time. The structural analysis identifies both content-based and melody-based similarity regions and when they are represented with vector format, the accurate music data search engines can be developed based on quarry by humming.

Error concealment in Music streaming: The most recently proposed content-based unequal error protection technique [23] effectively repairs the lost packets which have percussion signals. However this method is inefficient in repairing lost packets which contain signals other than percussion sounds. Therefore, the structural analysis such as the instrumental/vocal boundary detection simplifies the signal content analysis at the sender side and the pitch information (melody contour) is helpful for better signal restoration at the receiver side. The detection of content-based similarity regions (CBR) can avoid re-transmitting packets from the similar region. Thus the bandwidth consumption is reduced. In addition CBR can be construed to be another type of music signal compression scheme which can increase the compression ratio up to 10:1 whereas it is about 5:1 in conventional audio compression technique such as MP3.

8. CONCLUSION

In this paper, we propose a novel content-based music structure analysis approach, which combines high-level music knowledge with low-level audio processing techniques, to facilitate music semantic understanding. Experimental results of beat space segmentation, chord detection, vocal/instrumental boundary detection, and music structure identification & detection are promising and illustrate that the proposed approach performs more accurately and robustly than existing methods. The proposed music structure analysis approach can be used to improve the performance in music transcription, summarization, retrieval and streaming. The future work will focus on improving the accuracy and robustness of the algorithms used for beat space segmentation, chord detection, vocal/instrumental boundary detection, and music structure identification & detection. We also hope to develop complete applications based on this work.

9. REFERENCES

- [1] Bartsch, M. A., and Wakefield, G. H. To Catch a Chorus: Using Chroma-based Representations for Audio Thumbnailing. In *Proc. WASPA*. 2001.
- [2] Berenzweig, A. L., and Ellis, D.P.W. Location singing voice segments within music signals. In *Proc. IEEE WASPAA*. 2001.
- [3] Chai, W., and Vercoe, B. Music Thumbnailing via Structural Analysis. In *Proc. ACM Multimedia*. 2003, 223-226.
- [4] Cooper, M., and Foote, J. Automatic Music Summarization via Similarity Analysis. In *Proc. ISMIR*. 2002.
- [5] Collobert, R., and Bengio, S. SVM-Torch: Support Vector Machines for Large-Scale Regression Problems. *Journal of Machine Learning Research*. 2001, Vol 1, 143-160.
- [6] Duxburg, C., Sandler, M., and Davies, M. A Hybrid Approach to Musical Note Onset Detection. In *Proc. International Conference on DAFx*. 2002.
- [7] Foote, J., Cooper, M., and Girgensohn, A. Creating Music Video using Automatic Media Analysis. In *Proc. ACM Multimedia*. 2002.
- [8] Fujinaga, I. Machine Recognition of Timbre Using Steady-state Tone of Acoustic Musical Instruments. In *Proc. ICMC*. 1998, 207-210.
- [9] Ghias, A., Logan, J., Chamberlin, D., and Smith, B. C. Query By Humming: Musical Information Retrieval in an Audio Database. In *Proc. ACM Multimedia*. 1995, 231-236.
- [10] Goto, M. A Chorus-Section Detecting Method for Musical Audio Signals. In *Proc. IEEE ICASSP*. 2003.
- [11] Goto, M. An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds. *Journal of new Music Research*. June. 2001, Vol.30, 159-171.
- [12] Deller, J. R., Hansen, J.H.L., and Proakis, H. J. G. *Discrete-Time Processing of Speech Signals*, IEEE Press, 2000.
- [13] Kim, Y.K., and Brian, Y. Singer Identification in Popular Music Recordings Using Voice Coding Features. In *Proc. ISMIR* 2002.
- [14] Logan, B., and Chu, S. Music Summarization Using Key Phrases. In *Proc. IEEE ICASSP*. 2000.
- [15] Lu, L., and Zhang, H. Automated Extraction of Music Snippets. In *Proc. ACM Multimedia*. 2003, 140-147.
- [16] Navarro, G. A guided tour to approximate string matching, *ACM Computing Surveys*, March 2001, Vol.33, No 1, 31-88.
- [17] Rossing, T.D., Moore, F. R., and Wheeler, P. A. *Science of Sound*. Addison Wesley, 3rd edition 2001.
- [18] *Rudiments and Theory of Music*. The associated board of the royal schools of music, 14 Bedford Square, London, WC1B 3JG, 1949.
- [19] Scheirer, E. D. Tempo and Beat Analysis of Acoustic Musical Signals. *Journal of the Acoustical Society of America*. January 1998, Vol 103, No 1, 588 - 601.
- [20] Sheh, A., and Ellis, D.P.W. Chord Segmentation and Recognition using EM-Trained Hidden Markov Models. In *Proc. ISMIR* 2003.
- [21] Shenoy, A., Mohapatra, R., and Wang, Y. Key Detection of Acoustic Musical Signals, In *Proc, ICME* 2004.
- [22] Ten Minute Master No 18: Song Structure. *MUSIC TECH magazine*. www.musictechmag.co.uk (Oct. 2003), 62 – 63.
- [23] Wang, Y. *et al.* Content -Based UEP: A New Scheme for Packet Loss Recovery in Music Streaming. In *Proc. ACM Multimedia*. 2003. 412 – 421.
- [24] Xu, C.S., Maddage, N.C., and Shao, X. Automatic Music Classification and Summarization. In *IEEE Transaction on Speech and Audio Processing* (accepted).
- [25] Young, S. *et al.* *The HTK Book*. Dept of Engineering, University of Cambridge, Version 3.2, 2002.