SIMILAR SEGMENT DETECTION FOR MUSIC STRUCTURE ANALYSIS VIA VITERBI ALGORITHM

Yu Shiu[a], Hong Jeong[b] and C -C Jay Kuo[a]

[a]Integrated Media Systems Center and Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089-2564, USA
[b]Electronic and Electrical Engineering Department
Pohang University of Science and Technology, Pohang, South Korea
E-mails: yshiu@usc.edu, hjeong@postech.ac.kr and cckuo@sipi.usc.edu

ABSTRACT

The analysis of audio signals of popular and rock songs of the verse-chorus form to reconstruct its original musical structures is investigated in this work. We first compute the similarity degree between any two measures in a song based on selected features and represent these numbers in a measure-based similarity matrix. Then, we study the similarity across a sequence of consecutive measures, which is revealed by straight segments in parallel with the diagonal line of the similarity matrix. Generally, chorus parts have higher similarity values while verse parts have lower similarity values. As a result, the verse parts are difficult to detect in the presence of the chorus parts. To tackle this problem systematically, the Viterbi Algorithm is adopted to find optimal paths in the lower-triangular similarity matrix, which represent repetitive segments of both choruses and verses. Finally, several post-processing steps are developed to decode the music structure into the verse, the chorus and other non-repetitive parts. Experimental results obtained from several musical audio data are shown to demonstrate the performance of the proposed method.

1. INTRODUCTION

Automatic music structure analysis from audio signals is an interesting topic that receives a lot of attention these days. The technique can be used for music data analysis, indexing, retrieval and management. The music structure of many songs, including modern popular and rock songs, is of the verse-chorus form [1]. Under this form, chorus and verse parts are two different repetitive patterns. The chorus parts contain the same melody, chords and lyrics while the verse parts have the same melody and chord but different lyrics. Usually, the verse parts are for the story-telling purpose, and the chorus parts are for people to sing along. The analysis of audio signals of songs of the verse-chorus form to reconstruct its original musical structure is the main objective of this research.

Both verses and choruses appear as repetitive parts in a song. Accurate detection of both verse and chorus parts is a key component to the success of automatic music structure analysis. Foote [2] used the similarity matrix of a music piece along with its novelty measure for audio summarization and segmentation. His work aimed at detecting the boundaries of two segments that have distinct characteristics. Goto [3] examined the problem of chorus detection based on their high similarity. However, he did not consider the extraction of verse parts. In this work, we first compute the similarity degree between any two measures in a song based on selected features and represent these numbers in a measure-based similarity matrix. Then, we study the similarity across a sequence of consecutive measures, which is revealed by straight segments in parallel with the diagonal line of the similarity matrix. However, as compared to the chorus, verses tend to have weaker similarity among themselves due to different lyrics. This makes their robust detection more difficult.

Two techniques are proposed to enhance the detection performance of repetitive segments here. First, relative intensities of all pitch classes are examined. In particular, low frequency notes (lower than A3) and high frequency notes (higher than A6) are removed from the pitch class profile (PCP) feature calculation if they have dominating intensities since they are primarily contributed by musical instruments rather than human voices. Second, the Viterbi algorithm is used to find the optimal path in the lower-triangular part of a similarity matrix. Even though there may exist low similarity parts in a verse or chorus segment, the Viterbi algorithm can determine the global optimal segment while ignoring low similarity measures locally. Finally, we introduce post-processing steps to decompose the music structure into verses, choruses and non-repetitive parts such as intro, bridge and outro.

The rest of this paper is organized as follows. The framework of our music structure analysis methodology is explained in Sec. 2. Then, the problem of feature extraction and similarity computation for measures in audio signals is addressed in Sec. 3. Detection of multiple consecutive segments of high similarity using the Viterbi algorithm is discussed in Sec. 4. Experimental results are presented in Sec. 5. Concluding remarks are given in Sec. 6.

2. FRAMEWORK OF MUSIC STRUCTURE ANALYSIS

The concept of "similarity matrix" was introduced in [2][4] to measure the similarity degree between any two intervals of this song. Consider a song that is uniformly partitioned into L intervals. Simply speaking, similarity matrix S is a matrix of dimension $L \times L$, whose element s_{ij} represents the similarity degree between interval i and interval j, $1 \leq i, j \leq L$. In a measure-level similarity matrix, element s_{ij} represents the similarity between measures i and j. In other words, the "measure" is chosen as the basic interval unit. We assume the time signature of 4/4 for songs in our dataset, which is ubiquitous in popular and rock songs. The musical tempo information about the duration of a quarter note and a measure was examined in [4]. Since a measure of a song contains an audio signal over a period of time, we can divide it into finer sub-intervals and extract

some feature from each interval. Then, element s_{ij} can be calculated as

$$s_{ij} = \frac{1}{N}\sum_{n=1}^{N} x_{in}^T x_{jn}, \quad (1)$$

where $\{x_{i1}, \cdots, x_{iN}\}$ and $\{x_{j1}, \cdots, x_{jN}\}$ are sequences of feature vectors for measures i and j, respectively. In (1), x_{il}, $l = 1, 2, ..., N$, is a feature vector, and N is the number of the feature vector in a measure. Thus, the inner product of the feature vectors that are in the same position in a measure, say, the inner product of the feature vectors for the 3rd note in i-th measure and j-th measure is calculated in (1). Then, all inner products are averaged. The inner product is widely used in information retrieval systems as the similarity measure[5]. Without loss of generality, we can normalize the value of s_{ij} to lie between 0 and 1 as long as the feature vector is an unit vector. The higher the value is, the more similar the two measures of interest. The detailed procedure in similarity matrix computation will be presented in Sec. 3. Please note that (1) may not achieve the optimal similarity between two corresponding measures because their imperfect synchronization. In our [4], a dynamic time warping (DTW) technique was used to calculate the optimal similarity.

Given a measure-level similarity matrix, our next task is to study the global musical structure based on the pattern analysis of the similarity matrix. For example, we are interested in finding repetitive parts, since they represent verses or choruses. As shown in Fig. 1, an off-diagonal interval in the similarity matrix with consecutive high similarity values from (i_1, j_1) to (i_M, j_M) means that a strong similarity between two segments in the song composed by consecutive measures $j_1, \cdots, j_M$ and $i_1, \cdots, i_M$, respectively. Then, the two segments could be either a chorus, a verse or the combination of a chorus and a verse.

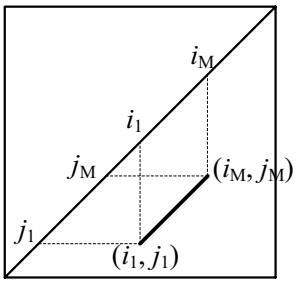

Fig. 1. Illustration of two similar segments in a song composed by two sequences of measures, $j_1 \cdots j_M$ and $i_1 \cdots i_M$, respectively.

Generally speaking, the chorus parts have strong similarity among themselves and show very apparent straight lines in the similarity matrix. The similarity values between two verse parts are not consistently high. They cannot constitute a solid straight but rather vague straight or broken line with many low similarity values in between. Therefore, the verse parts are difficult to detect and new techniques are needed to detect them. A systematic approach to the detection of segments of high similarity based on the Viterbi algorithm is described in Sec. 4.

3. PCP AND FILTERED PCP FOR SIMILARITY MATRIX COMPUTATION

In this section, we study features extracted from audio signals so as to compute the similarity degree between two measures as given in Eq. (1). The feature adopted in our work is the pitch class profile (PCP) [6], which is similar to the chorma vector in [3]. Each element in the vector represents the relative intensity of one of the 12 pitch classes, *i.e.,* A, $A\sharp$, B, C, $C\sharp$, D, $D\sharp$, E, $E\sharp$, F, G and $G\sharp$. It is calculated once for each basic time unit, which is selected to be the length of one half beat. For example, for a time signature of 4/4, the quarter note is one beat so that the duration of a one-eighth note is the basic time unit. The PCP vector are reported to be effective in musical key finding and identifying chord names in [6].

The calculation of PCP includes several steps. First, a Hamming window is applied to the music signal and its discrete-time Fourier transform (DFT) is calculated. Next, peaks that correspond to dominant harmonic components are picked from the magnitude spectrum, and their frequencies are mapped to one of the 12 pitch classes. Third, the energy of the peaks in the magnitude spectrum is added to the element of the PCP feature vector according to the pitch class number. That is, energies of all the peaks that have pitch class number i are added to the i-th element of a PCP vector. Each element of a PCP vector represents the relative intensity of each pitch class number. Finally, the PCP vectors are normalized to be with the unit length since we are only concerned with the energy distribution pattern of the PCP vector.

One main problem with the PCP feature vector extraction is that the accompanying background music from instruments such as basses and guitars may provide repetitive phrases all over the whole song, and their intensities are so strong that they dominate the PCP feature vectors. Then, the similarity matrix using PCP feature vectors may not reveal the repetitive patterns of choruses and verses properly. Instead, it shows the repetitive pattern of accompanying phrases in form of many short segments in the similarity matrix.

To suppress the effect of repetitive accompanying musical phrases, the intensity of each individual semitone between note A1 (55Hz) and A8 (7040Hz) is examined for the whole song. The note number can be computed as

$$\textit{Note Number for A Semitone} = \lfloor 12 \times \log_2(\frac{f}{440}) \rfloor + 69, \quad (2)$$

where 69 is the note number of A4. The middle range that lies between A3 and A6 (with A3 and A6 included) is the range where most vocal sounds are located, and it often corresponds to the frequency range of the main melody. Thus, if notes lower than A3 (220Hz) or higher than A6 (1760Hz) have strong intensity as compared to that of the middle range between A3 and A6, their frequency components are removed from the PCP calculation. One example is U2's *Vertigo*, where notes' intensities below A3 (note number 57) and above A6 (note number 93) are much higher than those between A3 and A6. The similarity in the middle range is masked by the low and high frequency components. After they are removed in the calculation of PCP, the similarity of the verse and chorus are shown more clearly in contrast to that of the others. The resulting PCP is called the *filtered PCP*.

4. SIMILARITY SEGMENT DETECTION VIA VITERBI ALGORITHM

4.1. Viterbi Algorithm

Once the similarity matrix is given, we would like to detect segments along its sub-diagonals that have high similarity values consecutively. Since the matrix is a symmetric one, we can focus on the lower-triangular part only. To overcome the problem of weaker similarity of verses, the Viterbi algorithm is used to detect these line segments reliably. The algorithm starts from the first measure of music

signals in the bottom-left corner of the similarity matrix. Originally, the x- and y-axes of the similarity matrix represent the measure index number of a given song. To perform the Viterbi algorithm, we interpret the x-axis as "time", the y-axis as the "state", and the element s_{ij}, as the probability at time i and state j. Thus, a higher similarity degree implies a larger probability. The Viterbi algorithm attempts to find a more similar segment, thus, a higher cumulative probability.

For each time index i and state index j, the Viterbi algorithm can update cumulative probabilities of different paths along time, and find the path with the highest probability. We use $Q(i-1,k)$ to denote the largest cumulative probability from some initial time i_0 to time $i-1$ and state k. Then, the largest cumulative probability $Q(i,j)$ from time i_0 to time i and state j can be written as

$$Q(i,j) = [\max_k P_T(j,k)Q(i-1,k)]P_S(i,j), \tag{3}$$

where $P_T(j,k)$ is the transitional probability from state k to state j and $P_S(i,j) = s(i,j)$ is the probability at time i and state number j. The best previous state for time i and state j is the one that maximizes $P_T(j,k)Q(i-1,k)$. Thus, it can be expressed as

$$R(i,j) = \arg\max_k P_T(j,k)Q(i-1,k), \tag{4}$$

Since only sub-diagonal lines are pertinent to the similarity of segments composed by consecutive measures, the state transition probability $P_T(j,k)$ is selected accordingly to reflect such a preference. That is, for state j, we choose

$$P_T(j,k) = \begin{cases} P_{T0}, & j = k+1, \\ \frac{1-P_{T0}}{L-1}, & otherwise, \end{cases} \tag{5}$$

where L is the number of measures in a song. Furthermore, we demand

$$P_{T0} > \frac{1-P_{T0}}{L-1}$$

to guarantee the preference along the sub-diagonal line. Practically, in the design of P_{T0}, we may examine the ratio of P_{T0} to $(1-P_{T0})/(L-1)$, which indicates the degree of preference to be placed along the 45-degree line. The larger the ratio is, the less probable that the optimal path will deviate from the 45-degree line.

Given appropriate initial conditions, the Viterbi algorithm recursively calculates $Q(i,j)$ and $R(i,j)$ first for $2 \le i \le L$, where L is the number of measures of the song, and then for $1 \le j < i$. At time $i = L$ (or the last measure), the maximum of $Q(L,j)$ for all states $1 \le j < L$ and the corresponding previous state $R(L,j)$ can be found via

$$Q^* = \max_{1 \le j \le L-1} Q(L,j), \tag{6}$$

$$R_L^* = \arg\max_{1 \le j \le L-1} Q(L,j) \tag{7}$$

Since the optimal path should lie in the lower triangular part of the similarity matrix, we demand $i > j$ for $Q(i,j)$. Backtracking is then applied to R_i^* in order to find the previous R_{i-1}^*. Then, the optimal state sequence can be found accordingly and expressed as

$$R_2^*, R_3^*, \cdots, R_L^*, \tag{8}$$

which is also called the optimal path in the similarity matrix. The initial condition for $Q(i,j)$ is $Q(2,1)$, which is defined to be

$$Q(2,1) = P_{Init} \cdot s_{21}, \tag{9}$$

where P_{Init} is an arbitrary positive constant.

4.2. Post-processing

The optimal path obtained by Viterbi algorithm as given in Eq. (8) have the following interesting properties.

1. It includes all high similarity sub-diagonal lines in the lower-triangular part of the similar matrix, which correspond to the chorus part.
2. It may transverse through lines of relatively weaker similarity, which correspond to the verse part.
3. If there are no strong similar segments, the optimal path may stay along the best path for the i-th measure that corresponds to the current time.

Given an optimal path, we develop several post-processing techniques for verse and chorus detection. First, the detected optimal path is segmented based on the change of similarity values, which is similar to the method given in [2]. This method detects dramatic changes of the average similarity which corresponds to used chords in the music. One simple way is to compute the averaged similarity values within two running windows, which are immediately before and after the current position, along the optimal path. If the difference of these two values is sufficiently large, we claim that there is a dramatic change. The similarity matrix for Nirvana's *Smell like teen spirits* is shown in Fig. 2, where the optimal path is indicated by a sequence of black dots. The optimal path could be segmented by the points of dramatic changes. Furthermore, if certain segment along the optimal path has a small average similarity degree, that segment is removed.

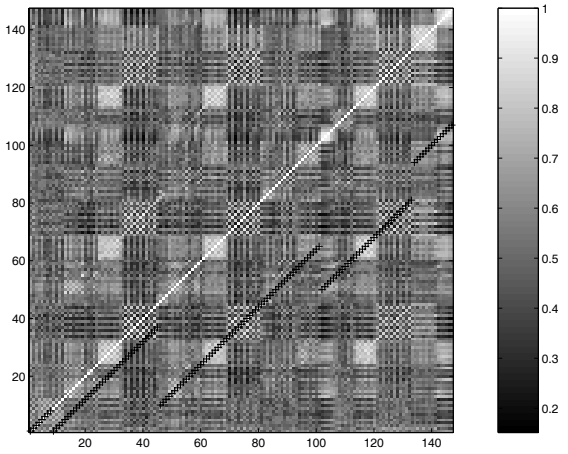

Fig. 2. The similarity matrix of Nirvana's *Smell like teen spirit*, where the detected path detected by the Viterbi algorithm is shown in black.

Second, if two similar parts of a song are overlapped, their boundaries have to be modified. For example, consider one detected segment starts from (i_1, j_1) to (i_M, j_M) as shown in Fig. 1. If $i_1 \le j_M$, the two corresponding similar parts (*i.e.,* $(i_1 \cdots i_M)$ and $(j_1 \cdots j_M)$ are overlapped in the interval of $(i_1 \cdots j_M)$. Then, we have trim their boundaries so that $j_M < i_1$. Suppose λ_1 and λ_2 are the numbers of measures to be trimmed for the head of $(i_1 \cdots i_M)$ and the tail of $(j_1 \cdots j_M)$, respectively. We demand

$$j_M - \lambda_2 < i_1 + \lambda_1. \tag{10}$$

In other words, the total number of measures to be trimmed $\lambda_1 + \lambda_2$ should be no less than $j_M - i_1 + 1$. To keep the detected segment as long as possible, we can set

$$\lambda_1 + \lambda_2 = j_M - i_1 + 1. \quad (11)$$

Since λ_1 and λ_2 are both positive integers, the best combination could be searched exhaustively using (12) so as to minimize the following accumulated similarity:

$$\arg\min_{\lambda_1,\lambda_2} \sum_{q=0}^{\lambda_1-1} S(i_1+q, j_1+q) + \sum_{p=0}^{\lambda_2-1} S(i_M-p, j_M-p) \quad (12)$$

To give an example, the detected segment in Fig. 2 from (46,10) to (101,65) corresponds to two overlapping parts in the song, i.e. one part from the 10-th measure to the 65-th measure and the other part from the 46-th measure to the 101-th measure. After the post-processing, we can trim the long segment into two short segments: (46,10)-(81,45) and (82,46)-(101,65).

Finally, to identify the verse parts, two conditions need to be met. First, their similarity is usually detected together with the chorus. The above trimmed segment from (46,10) to (81,45) in Fig. 2 actually indicates the similarity of two verse-chorus combined parts. Since the similarity of verses is lower than that of choruses in the combined verse-chorus part, we are able to use this property to separate the verse and the chorus structure of a song accordingly. Second, the final segmentation points for verse and chorus are one of the measures that dramatic change of average similarity described earlier in the section.

5. EXPERIMENTAL RESULTS

In the experiment, the proposed music structure analysis method was applied to a collection of 40 popular and rock songs in 80's and 90's. Examples include Nirvana's *Smell like teen spirit*, Oasis' *Don't look back in anger*, Police's *Every breath you take*, etc. These musical signals are of the following format: sampled at a rate of 22,050Hz, 16 bits per sample with mono channel. For each song, the musical tempo is assumed to be available, which may be obtained using techniques in [4] or from published musical sheets. The hamming window has the duration of a one-eighth note. All data in our collection have the $\binom{4}{4}$ time signature with the quarter-note as the beat. The window has a length of 250msec for 120 Beats Per Minute (BPM) tempo. The PCP feature vectors are then calculated for each windowed musical signals with no overlapping.

Performance was first evaluated based on the correctness of decomposing the music structure into verse and chorus parts. A song can be decomposed into a sequence of repetitive elements: verse(V) and chorus (C) and other non-repetitive elements such as intro (I), bridge (B) and outro (O). Please note that intro, bridge and outro are non-repetitive parts in the beginning, the middle and the end of tested songs, respectively. For example, the structure of Nirvana's *Smell like teen spirits* is IVCVCBVCO. The total correctness rate is $31/40 = 77.50\%$ for the dataset. Errors are mainly due to the complicated structure of some songs, including the difficulty in discriminating verses from choruses, multiple verse patterns, etc.

Next, we test the retrieval correctness of each segment's duration by using the F-measure [3], [5], which is defined as the harmonic mean of recall R and precision P as

$$F = \frac{2RP}{R+P}, \quad (13)$$

Table 1. The performance of chorus detection in terms of recall (R), precision (P) and F-measure (F).

	R	P	F
Original PCP	81.4%	79.9%	80.6%
Filtered PCP	89.3%	86.4%	87.8%

Table 2. The performance of verse detection in terms of recall (R), precision (P) and F-measure (F).

	R	P	F
Original PCP	61.7%	58.0%	59.8%
Filtered PCP	71.2%	66.5%	68.8%

where R is ratio of the number of measures that are correctly detected using our method over the number of correct measures in a given song and P is the ratio of the number of measures that are correctly detected using our method over the total number of detected measures. Among songs that could be correctly decomposed, the values of R, P and F for the chorus and the verse parts are shown in Tables 1 and 2, respectively. We see that the performance improves due to the use the filtered PCP feature over the original PCP feature in the similarity matrix computation.

6. CONCLUSION

A framework of automatic music structure analysis from audio signals was proposed in this work. The similarity matrix that records the similarity degree between measures was introduced. Then, the Viterbi algorithm was used to detect long similarity segments along the sub-diagonals of the matrix. Several post-processing techniques were proposed to fine-tune the decomposition procedure so that we are able to decompose a song into repetitive parts (*i.e.,* verses and choruses) and non-repetitive parts (*i.e.,* intro, bridge and outro). Performance of the proposed scheme was demonstrated.

7. REFERENCES

[1] S. Davis, *The Craft of Lyric Writing*, Writer's Digest Books, 1 edition, 1985.

[2] J. Foote, "Automatic Audio Segmentation using a Measure of Audio Novelty," *IEEE International Conference on Multimedia and Expo*, 2000.

[3] M. Goto, "A chorus-section detecting method for musical audio signals," *IEEE International Conference on Acoustic, Speech, and Signal Processing*, 2003.

[4] Y. Shiu H. Jeong and C.-C. J. Kuo, "Musical structure analysis using similarity and dynamic programming," *Proceedins of SPIE, Multimedia systems and applications VIII*, 2005.

[5] D. Jurafsky and J. H. Martin, *Speech and language processing*, Prentice Hall, 2000.

[6] Emilia Gomez, "Tonal description of polyphonic audio for music content processing," *INFORMS Journal on Computing*, 2005.