UNIVERSITY OF CALIFORNIA

Los Angeles

Data-Driven Structural Sequence Representations of Songs and Applications

A dissertation submitted in partial satisfaction of requirements for the degree Doctor of Philosophy in Electrical Engineering

by

Chih-Li Wang

2013

ABSTRACT OF THE DISSERTATION

Data-Driven Structural Sequence Representations of

Songs and Applications

by

Chih-Li Wang

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2013

Professor Vwani Roychowdhury, Chair

Content-based music analysis has attracted considerable attention due to the rapidly growing digital music market. A number of specific functionalities, such as the exact look-up of melodies from an existing database or classification of music into well-known genres, can now be executed on a large scale, and are even available as consumer services from several well-known social media and mobile phone companies. In spite of these advances, robust representations of music that allow efficient execution of tasks, seemingly simple to many humans, such as identifying a cover song, that is, a new recording of an old song, or breaking up a song into its constituent structural parts, are yet to be invented. Motivated by this challenge, we introduce a method for determining approximate structural sequence representations purely from the chromagram of songs without adopting any prior knowledge from musicology. Each song is represented by a sequence of states of an underlying Hidden Markov Model, where each state may represent a property of a song, such as the harmony, chord, or melody. Then, by adapting different versions of the sequence alignment algorithms, the method is applied to the problems of: (i) Exploring and identifying repeating parts in a song; (ii) identifying cover songs; and (iii) extracting similar sections from two different songs. The proposed method has a number of advantages, including elimination of the unreliable beat estimation step and the capability to match parts of songs. The invariance of key transpositions among cover songs is achieved by cyclically rotating the chromatic domain of a chromagram. Our data-driven method is shown to be robust against the reordering, insertion, and deletion of sections of songs, and its performance is superior to that of other known methods for the cover song identification task.

The dissertation of Chih-Li Wang is approved.

Lieven Vandenberghe

Ying Nian Wu

Kung Yao

Vwani Roychowdhury, Committee Chair

University of California, Los Angeles

2013

Table of Contents

Biography

Chih-Li Wang received his B.S. degree in Electrical Engineering from the National Tsing Hua University, Taiwan, and M.S. degree in Electrical Engineering from the National Taiwan University, Taiwan. His research focuses on signal processing for music analysis and music information retrieval.

Publications:

C.L. Wang, Q. Zhong, S.Y. Wang, and V. Roychowdhury, "Cover Song Identification by Sequence Alignment Algorithms," presented at the International Conference on Signal and Information Processing, 2010, pp. 187-191.

C.L. Wang, Q. Zhong, S.Y. Wang, and V. Roychowdhury, "Data-Driven Chord-Sequence Representations of Songs and Applications" presented at the International Association of Science and Technology for Development (IASTED) on Signal and Image Processing, 2011.

1 Introduction

The proliferation of the Internet has facilitated the acquisition of digital music. However, digital recordings, especially older recordings, are not always tagged with appropriate metadata. Therefore, content-based automatic music analysis, such as organizing songs into genres, music summarization, locating a specific song, or finding a cover version among thousands of songs, has attracted considerable attention in recent years.

Various groups of researchers have focused on content-based automatic music analysis. Normally, automatic music analysis comprises two steps. The first is numerical representation of a musical property, such as timbre, pitch, and tempo, called a "feature." Features can be based on either the time domain signals or the frequency domain spectrum. For example, features such as the root mean square (RMS) and the zero-crossing rate (ZCR) are based on the time domain signals, while information such as the mel-frequency cepstrum coefficients (MFCC) [1], chromagram [2], and key strength [3] rely on the frequency domain spectrum. Some of these features, such as MFCC, the chromagram, and the fluctuation pattern [4], are designed to simulate human perception. Algorithms are applied to an extracted feature or set of features to achieve the goals of the musical analysis. However, feature selection is crucial. If a suitable feature is not selected, the analysis results will not be efficient even if the most sophisticated algorithm is used. Some of the important features and their properties will be discussed in sections 2 and 3.

More recently, Venkatachalam et al. [5] and Wells et al. [6], [7] achieved remarkable success in identifying exact song recordings from different sources. Using their method, only a short section (around 15 s) of a query song is needed to achieve identification. In brief, the short section is divided into frames with some overlap. Signals above 11,025 Hz are filtered to avoid high frequency noise. For each frame, the volume of signals in the time domain is normalized by histogram equalization. Then, the discrete cosine transform (DCT) is used to partition the signals into 15 frequency bands. Finally, the mean and variance over the frames of each band are calculated to represent the fingerprint of a piece. The search by range reduction (SRR) technique is used as the search engine. SRR starts with the space containing all the fingerprints in the database at stage 0; at stage J, the search space is reduced to the distance between the first J components of the fingerprints in the database and the first J components of the query's fingerprint smaller than reasonable constants. This process continues until the search space is small enough. The authors claimed that the false positive or false negative rate is less than 3.4%, regardless of whether the scale is large or small.

Widely used in speech recognitions, MFCC is also used in automatic music analysis [4], including the areas of music similarity measurement [8][9][10], classification of genres [11], [12], and identification of artists [13], [14]. Williams and Ellis [15] claimed that MFCC can remove or hide the phonetically irrelevant aspects of the signal and may not be suitable for speech/ nonspeech discrimination. They used a neural network to estimate the posterior probability (PPF) of a feature, and calculated statistics such as the mean per-frame entropy and the average probability "dynamism." They assumed that the audio signal is pre-segmented, and thus they were actually able to discriminate between the voice and music instead of segregating them. The lowest error rate of the method is 1.4%. Berenzweig, Logan, Ellis, and Whitman [8] tried to segment the music and the speech parts of FM signals using a hidden Markov model (HMM). Many features are examined in their studies, such as cepstral coefficients, log-PPFs, entropy, and so on. The lowest error rate that they achieved was 19%. The objective of both methods [8], [15] was to discriminate the speech and music parts of FM signals, and therefore, a phonetically relevant feature, such as the MFCC, was preferred.

Zhang [16] studied the patterns of a feature associated with a musical property and classifying music pieces by tree structure. The tree structure and the features used at each stage are shown in Figure 1. A threshold is set for each branch. For example, Zhang claimed that the average ZCR curves of purely instrumental music should be smoother (without peaks) than the curves of vocal music, and the spectrogram of purely instrumental music does not contain many ripples. Hence, in this method, some statistics, such as mean and variance, are calculated; a threshold can be set for each branch. However, the performance of the system was not evaluated. Furthermore, the pattern of a spectrum in a real case may not be as clear as that shown in [16]. Finally, Zhang's paper gives us a good understanding about using the spectrogram for different types of music, but the method may not be practical.

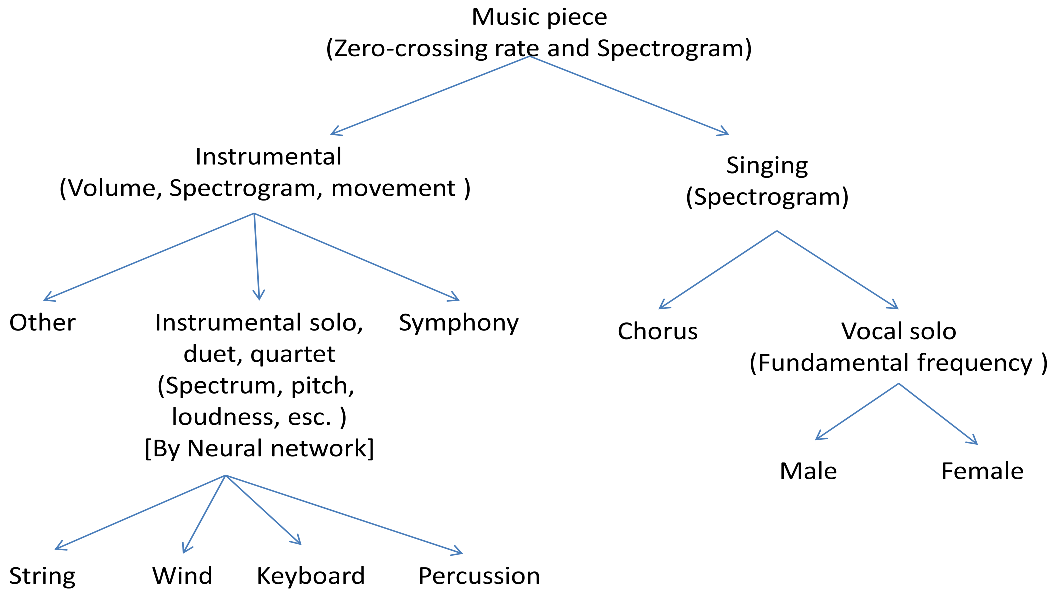


Figure 1 Tree structure of Zhang's method

Kosina[17] classified 63 metal, 65 dance, and 61 classical pieces of music by utilizing MFCC, ZCR, energy, and beat in the K-nearest neighbor (KNN) algorithm. The reported accuracy rate was 86.24% for $k = 4$ and 84.12% for $k = 3$. Although the results were satisfactory, metal, dance, and classical music are highly dissimilar genres; the performance of the system is unknown if the selected genres are similar. Furthermore, Malheiro [18] applied a three-layer feed-forward neural network (FFNN) [19] to classify classical music (flute, piano, violin, choral, and opera) using the perception of amplitude, ZCR, spectral centroid, and uniformity of the spectrum. The accuracy of the classification of the flute, piano, and violin music with 20 neurons was about 85%, of choral and opera music with 25 neurons about 90%, and of all five genres with 20 neurons around 64%. In addition, Li and Guo [20] applied a support vector machine to classify music. The features they used were (1) Perceptual feature sets: total spectrum power, subband powers, brightness, bandwidth, and pitch frequency, and (2) the cepstral feature vector, MFCC. The mean and variance of these features over the frames of a song were computed to represent the song. They reported that using the perceptual feature sets and MFCC with eight MFCC coefficients applying a support vector machine (SVM) yields the lowest error rate, which is less than 11%. They also reported that SVM is the best classification method among SVM, nearest neighbor (NN), 5-NN, and nearest center (NC).

Most of the methods above constitute supervised learning. For unsupervised learning, the *k*-means algorithm is simple and has been used in many research studies. However, choosing an unsuitable *k* may result in bad clustering. The fuzzy C-mean algorithm is similar to k-means; however, it gives the degree of belonging to a cluster instead of making a hard decision. However, setting a suitable *C* is also crucial to obtaining good results. Pampalk, Rauber, and Merkl [4] proposed a method to cluster songs by applying the fluctuation pattern (the rhythm pattern) in the self-organizing map (SOM). SOM is similar to the fuzzy c-means algorithm in that it does not make a hard decision, and hence, the relative similarity among songs can be read on the map. In section 3.3.4.3, instead of using the fluctuation pattern, different features or sets of features are examined to find the best feature or feature set to classify the genres.

For automatic music summarization, or segmentation, Foote [21] proposed the checkerboard method based on a frame-by-frame self-similarity matrix that calculates the pairwise distance of the MFCCs between the frames of a song. The simplest checkerboard is a 2×2 unit kernel, which is $\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$; a larger kernel can be obtained by using the Kronecker product of the 2×2

unit kernel with a matrix of one. The novelty scores are calculated by correlating a predefined checkerboard kernel with the similarity matrix sliding along the main diagonal. Thus, a novelty score measures the self- and cross-similarity of the past and future regions at a given time. A cutting point is at a moment with high self-similarity and low cross-similarity corresponding to a point with a local maximum novelty score. Foote and Cooper [22] tried to segment "Wild Honey" by U2 using MFCC and to cluster these segments by singular value decomposition (SVD). They showed a good agreement between manual and automatic segmentation of the song. However, in this method, it is not easy to determine what checkerboard size is suitable, since a larger checkerboard will result in many missing cutting points and a smaller checkerboard will result in many unnecessary cutting points. Another researcher, Logan[23], employed a fixed-state unsupervised HMM in a song's MFCCs in order to explore the most memorable excerpt by finding the sections corresponding to the most frequently occurring state. A similar method was proposed by Aucouturier and Sandler [24]. The authors concluded that a better result is obtained by using MFCCs together with several features, such as linear prediction and discrete cepstrum. However, *a priori* knowledge of the number of HMM states is crucial in these methods. We propose a method that adopts the chromagram in section 4.1.5. Several other research groups [25]–[28] used the chromagram (pitch class profile) [29] to extract chords or keys, a method that demonstrated great promise for automatic music analysis. A chromagram captures the melody similarity and is less sensitive to timbre. Hence, our method focuses more on the similarity of the melody than do the other MFCC-based methods described above.

Despite significant progress in the field of song identification by Venkatachalam et al. and Wells et al. [5]–[7], [30], a similar but not identical topic, cover song identification, still presents a challenge. A cover version is a new recording of an old song. Typically, the melody of the original song is retained but other critical properties, such as the performing instruments, rhythm, and structure, may be altered. Hence, unlike song identification, which requires an exactly matching algorithm, cover song identification requires a robust matching algorithm. A method based on the correlation of the two songs' beat-synchronous chromagram (BSC) [31] achieved the best performance in the cover song recognition task at the 2006 MIREX conference. To improve the system's robustness against the variation of a cover version, we propose a method based on the hidden Markov model (HMM) and the local sequence alignment algorithms presented in section 4. Instead of calculating the cross-correlation of the BSCs [31] directly, we align the state sequences derived from the HMMs of chromagrams, which further improves the robustness of the models. Moreover, a local alignment algorithm may perform better than the cross-correlation of the entire songs, especially when the structure of the cover version is different from the query song owing to the reordering, insertion, and deletion of some parts. Due to allowing gaps in the sequence alignment method to alleviate the tempo variation, tempo or beat estimation is not required, and therefore, a constant hop chromagram (CHS) is utilized in our method; thus, the complexity of the system is reduced. Furthermore, because a local alignment algorithm is used, matching that relies on the Smith–Waterman alignment may be more appropriate than the Needleman–Wunsch–Sellers algorithm (global alignment) [32] or dynamic time wrapping [33], [34], especially when the input query consists of only parts of the song.

In addition to cover song identification, "Songs that sound like other songs" [35] is also a fascinating area. A short section of a song might sound like that of another song, or a new song might be composed by connecting sections of several other songs with suitable key transpositions. Though the local sequence alignment of fixed-state HMMs is still used to explore similar sections of two songs, a superior method, a state-splitting state (SSS) HMM, is proposed for the task.

The HMM and sequence alignment algorithm are also used in Bello's method [32]. However, our method, a fixed-state or SSS HMM, is entirely data-driven. More specifically, our method adopts k-means clustering to extract the initial conditions of an HMM, whereas the initial conditions of Bello's method are given by prior knowledge of the musical chords [36], which may not be adequate for a wide variety of genres. Furthermore, in contrast to a predefined alignment score, the score in our method is based on the distance between the empirical mean of different states. Finally, knowledge of the beginning and ending points is necessary in the Needleman–Wunsch–Sellers algorithm used in Bello's method [32], which limits its usefulness for some applications

2 Features

Audio signals are composed of many properties, such as amplitude, tempo, pitch, volume, and so on. The first step of automatic music analysis is to decide which property is most important for the analysis. A musical property can be transferred into a numerical representation called a feature. The features used in audio signal analysis are divided into categories: physical and perceptual. A physical feature is based on a statistical analysis of the signal's properties, such as the ZCR, whereas a perceptual feature, such as MFCC, is designed to simulate human perception. Although humans can hear the detail of audio signals, it is sometimes not crucial for a human to identify sounds. Stereo signals were therefore transferred into mono by averaging the two channels throughout this study. Some of the features are introduced in this section.

2.1 Root-mean-square Energy (RMS)

The volume of audio signals can be represented by the root mean square (RMS) energy. The RMS is computed by taking the root average of the square of the amplitude. Calculating the RMS energy of a human's voice is simple, and it is sometimes a good indication of a human’s mood. For example, the RMS energy value of one's voice is larger when one is happy and lower when one is unhappy:

$$x_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}$$

2.2 Zero–crossing Rate

The ZCR is the number of zero crossings or sign changes in the time domain per interval. The ZCR relates to the fundamental frequency, since a sinusoidal signal will cross zero twice. However, in general, the tone of most audio signals is not pure. ZCR cannot completely represent all the information about the fundamental frequency since the partials at higher frequencies also cross zero in the time domain [37]. Due to the pronunciation of consonants, the ZCR curves show a number of significant peaks for a singing voice [16]. The ZCR curves are calculated as

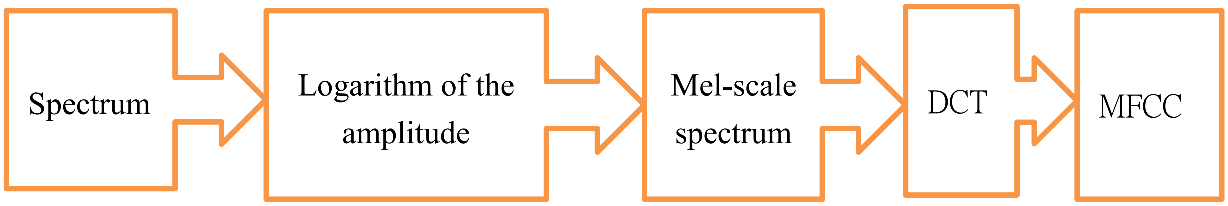
$$Z_n = \sum_{m} |sgn[x(m)] - sgn[x(m-1)]| w(n-m),$$

$$sgn[x(m)] = \begin{cases} 1 & x(n) \geq 0 \\ -1 & x(n) \leq 0 \end{cases}$$

$$w(n) = \begin{cases} 1/2 & 0 \leq n \leq N-1 \\ 0 & otherwise. \end{cases}$$

We will show that ZCR is effective for identification of instruments in section 3.3.4.3.2.

2.3 Mel-frequency Cepstral Coefficient (MFCC)



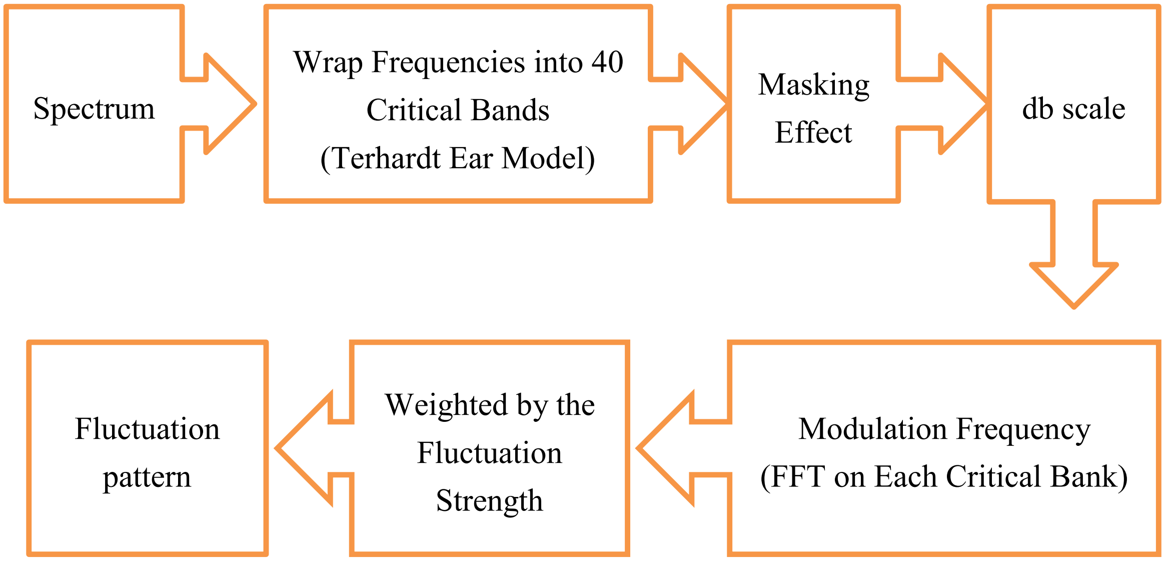
MFCCs are often used in speech or speaker recognition. They are also suitable for music information retrieval [1].

MFCCs are derived as follows [17]:

1. Take the short time Fourier transform of the signal and obtain the amplitude spectrum;
2. Take the logarithm of the amplitude;
3. Convert to the mel spectrum;
4. Take the discrete cosine transform (DCT).

The human auditory system's response is linear in frequencies lower than 1 KHz and logarithmic in frequencies higher than 1 KHz. Hence, the mel spectrum is applied in step 3. In a nutshell, the mel spectrum is computed by multiplying the spectrum by each of the triangular mel weighting filters and integrating the result. DCT in step 4 approximates the Karhunen Loève transform (KL transform) when the samples' correlation is high. Hence, DCT decorrelates the mel-frequency spectrum. Therefore, most of the information about the mel-frequency spectrum tends to be concentrated in a few low-frequency components of DCT.

2.4 Fluctuation Pattern


Spectrum
Wrap Frequencies into 40 Critical Bands (Terhardt Ear Model)
Masking Effect
db scale
Modulation Frequency (FFT on Each Critical Bank)
Weighted by the Fluctuation Strength
Fluctuation pattern

The rhythm pattern can be measured by the fluctuation pattern. Briefly, the fluctuation pattern is based on a spectrogram, but the frequency bands are modified according to a human's perception, and then the amplitude is transferred into db scale. Finally, the FFT (fast Fourier transform) is applied again on the time domain of the human perception-based spectrogram. The detailed process is as follows.

1. Divide the whole sequence of a song into sub-sequences of at least 2 s. Normally, a fluctuation pattern is measured every 6 s.
2. The spectrum of a sub-sequence is computed using 23-ms Hanning windows with a 50% overlap.
3. Terhardt outer-ear modeling is calculated, and the frequency is wrapped into 40 critical-bands according to the Bark scale [38], which considers the fact that the resolution of a

human's hearing in a frequency is different in different frequency ranges, as shown in Figure 2 .

4. A masking effect is the occlusion of a quiet sound by a louder sound when both sounds occur simultaneously at similar frequencies. The pre-defined masking matrix, which takes the influence of the j^{th} critical band on the i^{th} critical band into consideration, is applied. Hence, the spectrum is spread across critical bands.
5. The amplitudes are transformed into the dB scale.
6. The FFT is computed on each critical bank for each 6-s sequence to obtain the modulation frequency
7. A human's hearing sensation is most sensitive to a modulation frequency (rhythm frequency) of around 4 Hz, and sensitivity gradually decreases until a modulation frequency of 15 Hz. Hence, the amplitude-modulation is weighted by the fluctuation strength and retains only 0 Hz to 10 Hz, as shown in Figure 3 .

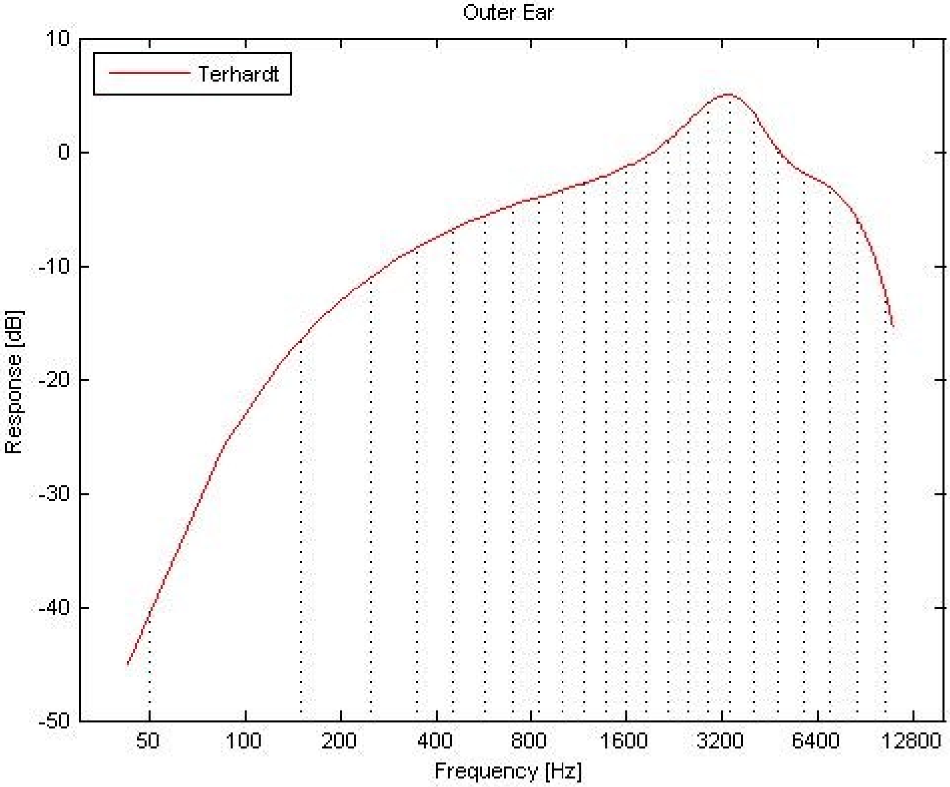


Figure 2 Terhardt outer-ear model (from MA toolbox)

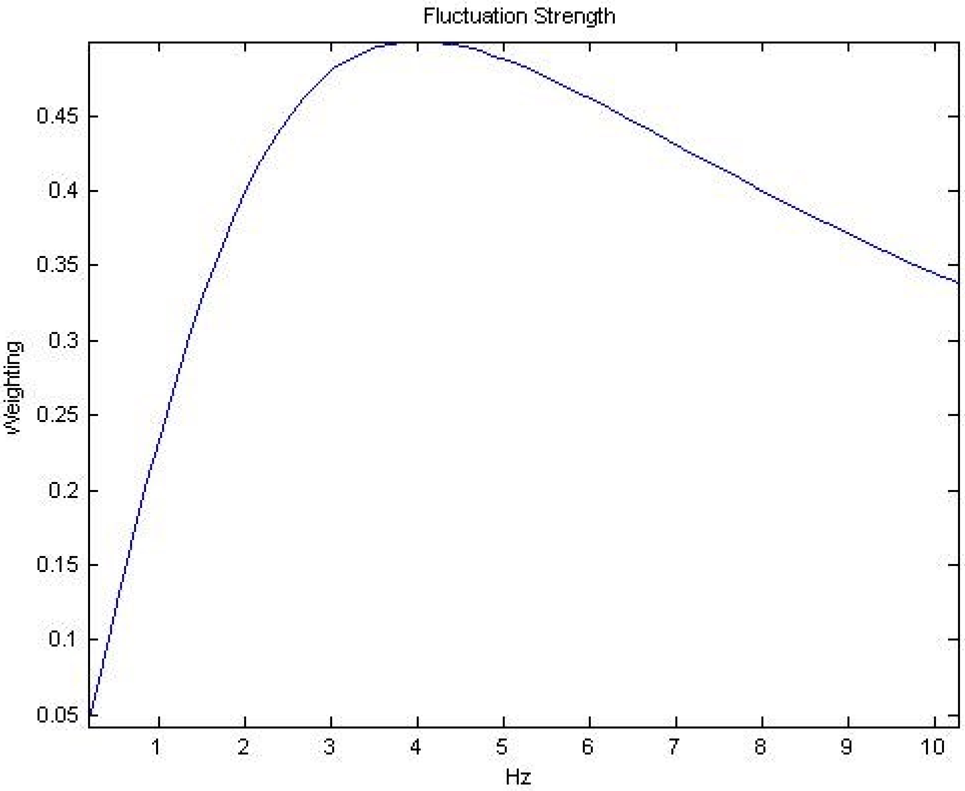


Figure 3 The fluctuation strength vs. the modulation frequency

For each sub-sequence (6 s), a three-dimensional figure, as shown in Figure 4, is obtained. The *y* axis is the critical bands; higher channels or critical bands mean higher frequencies, but the frequencies of the critical bands are not linear. The *x* axis is the frequency of the rhythm from 0-10 Hz, and the *z* axis, the color, is the strength of the fluctuation.

The strength of the critical bands (*y*-axis) can be summed for each rhythm frequency (or modulation frequency) (*x*-axis) to obtain the summary fluctuation pattern for each sub-sequence (6 s). Hence, the dimension of the fluctuation pattern is reduced from a 3-D matrix to 2-D vectors, called the summary fluctuation pattern, which loses the information about the source of the critical band of the rhythm. Figure 5 is the summary fluctuation pattern vs. time of a demo song. The *x*-axis is the time position. The *y*-axis is the rhythm frequency from 0 to 10 Hz. The color, or *z*-axis, is the rhythm strength.

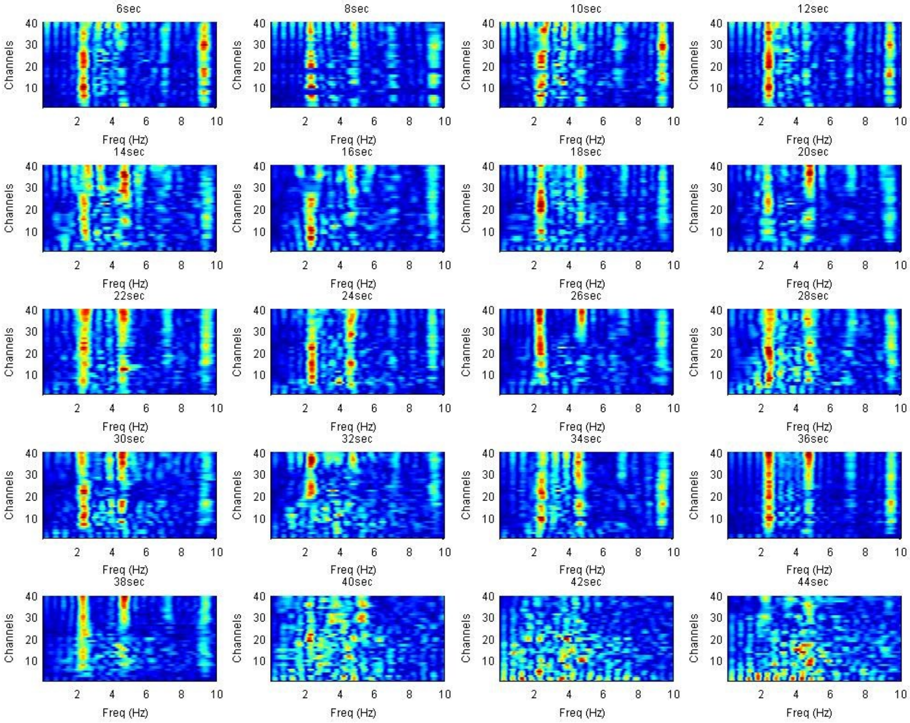


Figure 4: The first 44-s fluctuation pattern of the demo song

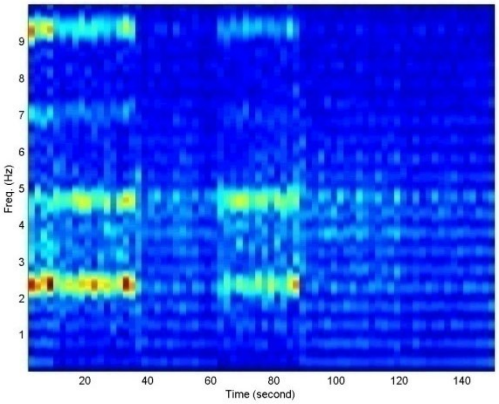


Figure 5: The summary fluctuation pattern of the demo song

2.5 Unwrapped/Wrapped Chromagram

2.5.1 Chromagram Estimation

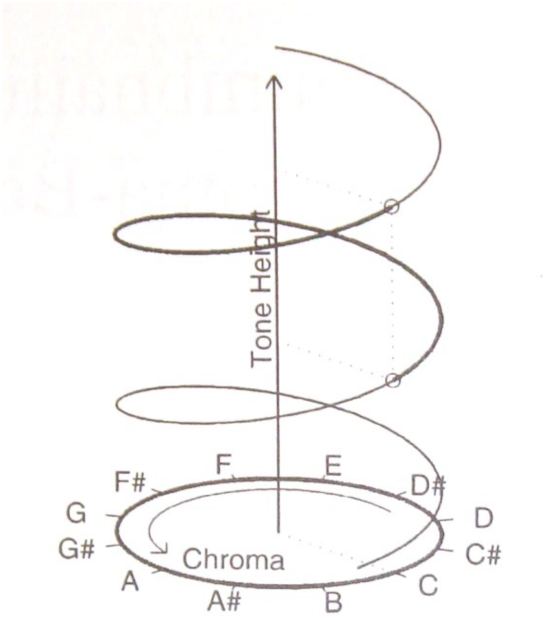


Figure 6 Shepard's helix of pitch perception

The human auditory systems' perception of pitch is a function of tone height and chroma, and can be represented as a helix, as shown in Figure 6 [39]. The tone height characterizes the vertical dimension and the chroma characterizes the angular dimension. A semitone is defined as

$\frac{f_1}{f_2} = 2^{\frac{1}{12}}$. There are 12 semitones (chroma), that is C, C#, D, D#, E, F, F#, G, G#, A, A#, within a chromatic octave. The tone height indicates to which octave a note belongs. Hence, the perception of two C notes with different tone heights will be different. An unwrapped chromagram distributes the energy along a different pitch and does not wrap the chroma with a different tone height, as shown in Figure 7 (a). On the other hand, a wrapped chromagram is composed of a 12-dimensional vector, which represents the relative intensity in each of 12 semitones in a chromatic scale, as shown in Figure 7(b). Similar to the short-time Fourier transform (STFT) and MFCC, a short-time chromagram representation is available, as shown in Figure 8.

The basic approach for obtaining a chromagram is as follows [40]:

1. Calculate the constant Q spectrum of the input signal.

- $f_k = \left(2^{\frac{1}{12}}\right)^k f_{min}$ is the kth spectral component;
- Q is the "quality factor" defined as $Q = \frac{f_k}{\delta f_k}$;
- Window length for the kth bins

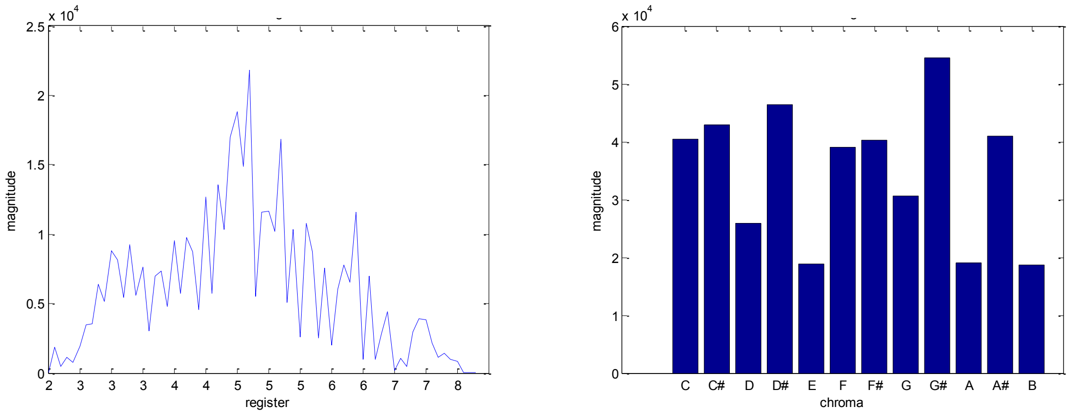
$N[k] = \frac{f_s}{\delta f_k} = \frac{f_s}{f_k} Q$ where f_s is the sampling rate.

- $X_{CQ}[k] = \frac{1}{N[K]} \sum_{n=0}^{N[k]-1} w[k,n] x[n] e^{-\frac{j2\pi Qn}{N[k]}}$ where $w[k,n]$ is the window function

2. Obtain chromagram vector :

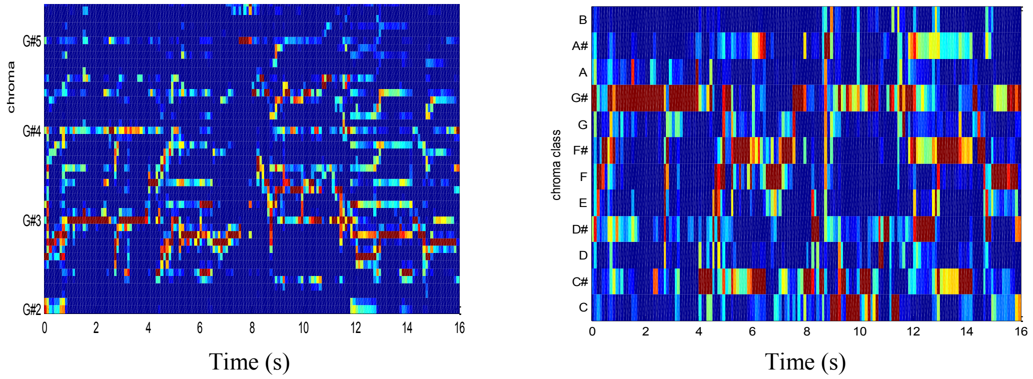
$CH(b) = \sum_{m=0}^{M-1} \left| X_{CQ}(b + 12m) \right|$

where b is the chromagram bin index (1,2…12) and M is the number of octaves spanned in the constant Q spectrum



(a) **(b)**

Figure 7 Chromagrams: (a) Unwrapped chromagram; (b) Wrapped chromagram



(a) **(b)**

Figure 8 Short-time chromagram: (a) Unwrapped chromagram; (b) Wrapped chromagram

2.5.2 Key Transposition

Human perception of pitch is less sensitive to the absolute pitch but more sensitive to the relative pitch. As a result, melodies in different keys can be perceived as being the same.

Figure 9 shows the wrapped/unwrapped chromagram of a song. The pitch of the song is raised 4 and 12 semitones without altering the tempo, using free software “Audacity.” The chromagrams when the pitch is raised 4 and 12 semitones are shown in Figure 10 and Figure 11, respectively. An octave is constructed of 12 semitones. Hence, raising the pitch of a song 12 semitones changes the tone height but does not alter the position of the chroma, while raising the pitch 4 semitones changes the position of the chroma. To make the comparisons shown in Figure 9, Figure 10 and Figure 11 easier to read, please pay attention to the root notes, defined as the notes with the maximum strength. In comparison to Figure 9, in Figure 10, the notes move up four positions. However, in Figure 11, the notes of the unwrapped chromagram still move up 12 positions, as compared to Figure 9, while those of the wrapped chromagram remain the same. Therefore, it can be concluded that the wrapped chromagram ignores the octave information, while the unwrapped chromagram retains it. Hence, by cyclically rotating the chromatic domain of the wrapped chromagram, a key invariance in the wrapped chromagram among songs sung in different keys is achieved.

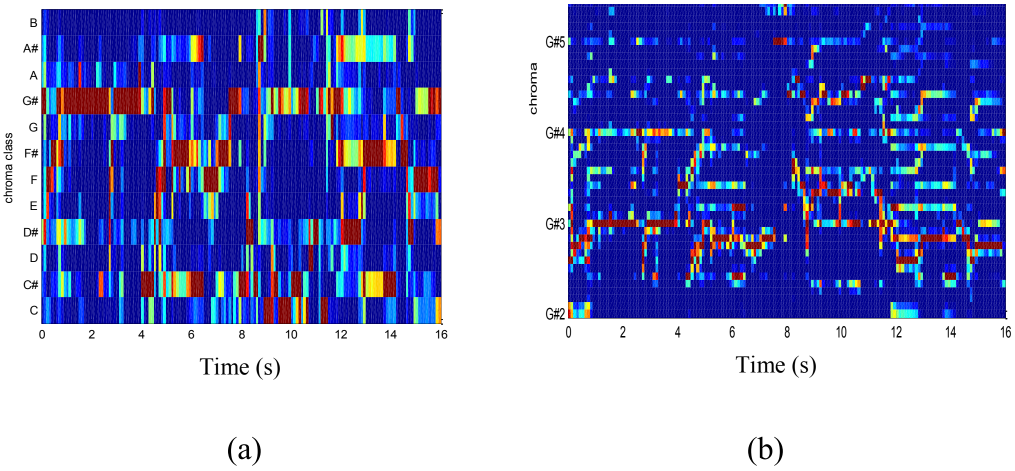
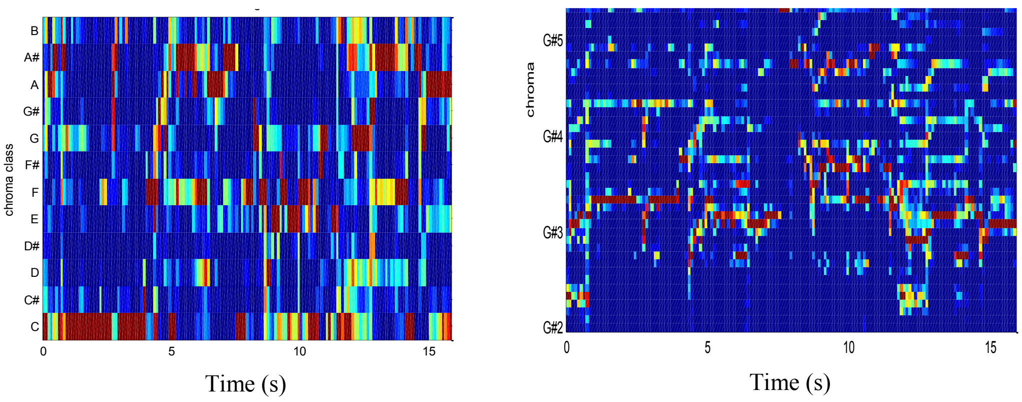
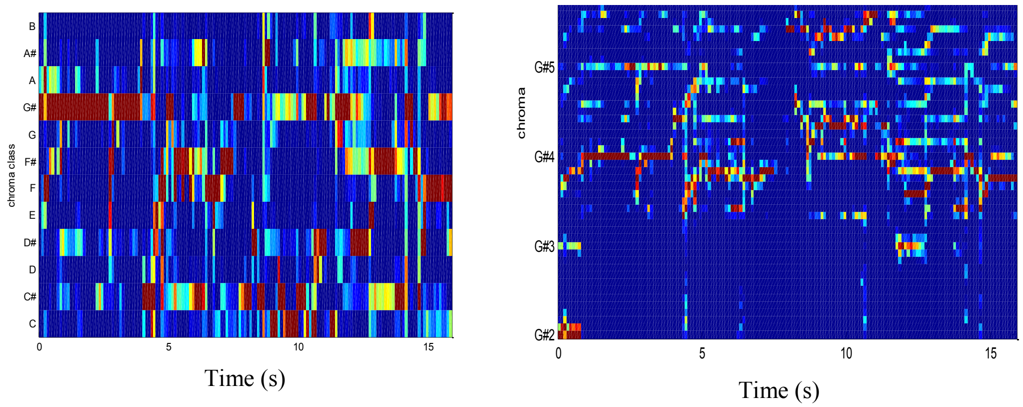


Figure 9 Chromagram of the song: (a) Wrapped chromagram; (b) Unwrapped chromagram



(a) (b)

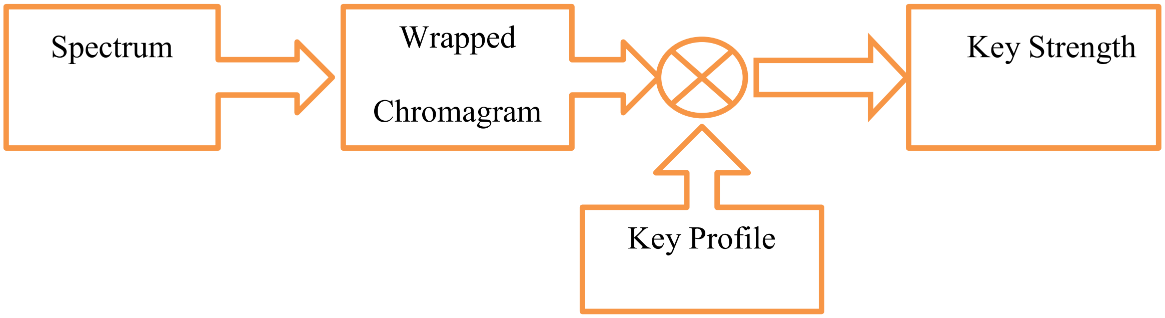
Figure 10 Chromagram of the song raising the pitch four semitones: (a) Wrapped chromagram; (b) Unwrapped chromagram



(a) (b)

Figure 11 Chromagram of the song by raising the pitch four semitones: (a) Wrapped chromagram; (b) Unwrapped chromagram

2.6 Key Strength



The key, which is a system of relationships between a series of pitches having a tonic, is an important feature in Western music. There are two key modes: major and minor. Hence, there are 24 keys. A key profile is the chroma distribution of a key and has been established through experiments [2]. The key strength is calculated by the cross-correlation of the chromagram with the key profile, as shown in Figure 12. The dominant key of a piece is detected reliably if the maximum value of the key strength is larger than 0.5, while if the maximum value is less than 0.5, the key of the piece is not well recognized by the method.

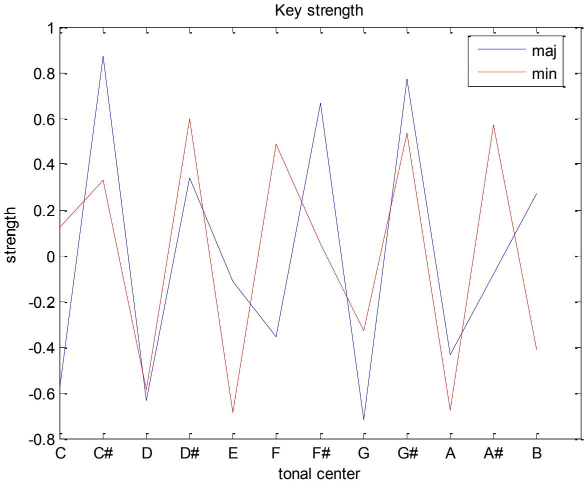


Figure 12: Key strength

3 Music Feature Interpretation

3.1 Self-similarity Matrix

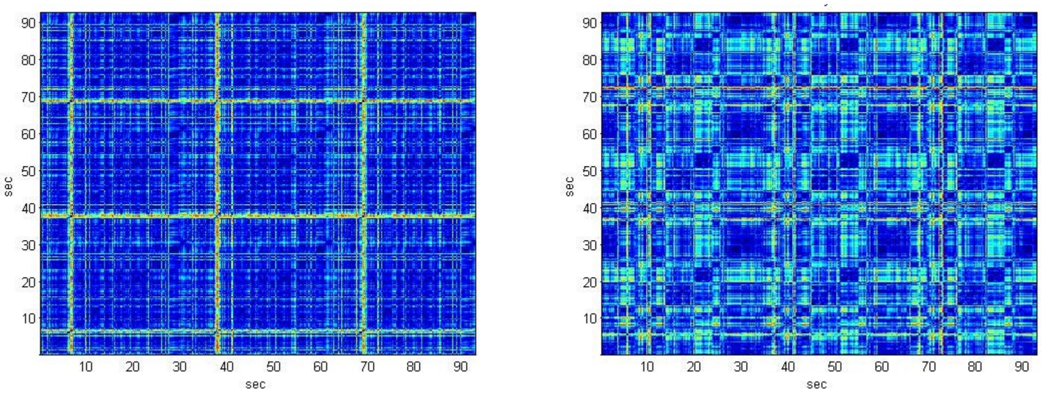
A self-similarity or similarity matrix is a non-parametric method to represent the structure of a song. A feature or a combination of features can be used to construct the similarity matrix. Basically, any pixel in the similarity matrix is the pairwise designed distance between two frames. Depending on the application, the distance can be the cosine, n-norm, or any properly designed distance.

Suppose a song is segmented into *N* frames and the feature (v) is a B-dimensional vector. The similarity matrix (S) will be a $N \times N$ matrix. A pixel or element at position i, j on the similarity matrix $S(i, j)$ is the distance $d(v_i, v_j)$ of v_i and v_j, where v_i: R^B is the feature vector of the i^{th}frame and d: $R^B \times R^B \rightarrow R$ is the distance. Since $d(v_i, v_j) = d(v_j, v_i)$, the similarity matrix is symmetric. Finally, if two sections within a song are similar in terms of a feature, a sub diagonal corresponding to the shorter distance between frames will be shown in the similarity matrix.

3.2 Interpretation

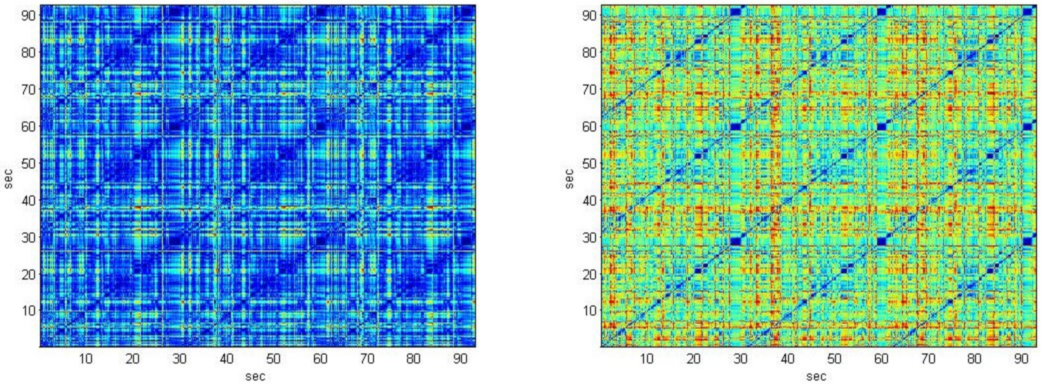
In order to identify which property is captured by a feature, we chose a Chinese song sung by a famous group, Lollipop. We cut the first 30 s of the song and replicated the sub-section twice at 30 s and 60 s; hence, the total length of the demo song was about 90 s. The song does not have a strong bass. A self-similarity matrix was adopted to check a feature's sensitivity to a property. If the feature is sensitive to the variation of a property, the distance between two similar sub-sections is much shorter than between dissimilar sub-sections. Hence, a sub-diagonal structure is obvious in the similarity matrix. In the demo track, since a 30 s sub-section is replicated at 30 s and 60 s, two subdiagonals beginning from 30 s and 60 s should be obvious in the similarity matrix, as shown in Figure 13, if the feature can detect a property's variation well. The similarity matrix of the MFCC, key strength, wrapped and unwrapped chromagram, and RMS are shown in Figure 13.

In Figure 13, the subdiagonals begin at 30 s and 60 s due to the replication. In addition, MFCC shows a clear edge at about 7 s, 37 s, and 67 s. After listening to the song carefully, we decided that this might be due to a sound generated by a crystal whose timbre is different from that of the rest of the song. Therefore, it may suggest that MFCC is a feature that is sensitive to the timbre variation.



(a)

(b)



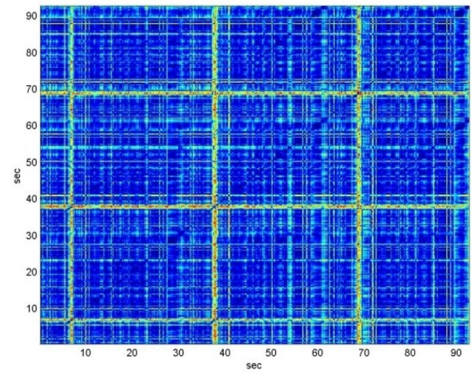
(c) (d)

Figure 13 The similarity matrix of (a) MFCC, (b) key strength, (c) wrapped chromagram, and (d) unwrapped chromagram

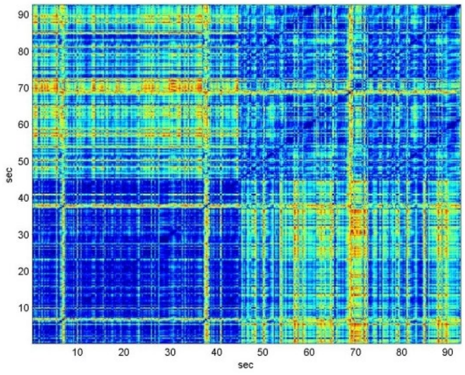
3.2.1 Key Variation

The pitch of the sub-section after 45 s of the demo track was raised by 4 or 12 semitones of the key using “Audacity” software in order to check whether a feature is sensitive to the pitch change. Again, 12 semitones make an octave. Hence, raising the pitch of a section 12 semitones changes the tone height but does not change the position of the chroma, while raising it by 4 semitones does change the position of the chroma. Therefore, the sensitivity of the variation of the octave and chroma can be examined.

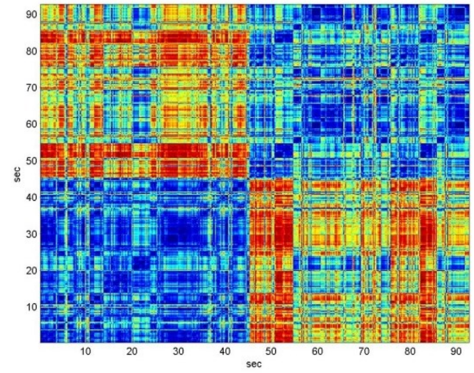
The similarity matrices of MFCC, key strength, and wrapped/unwrapped chromagram are given in Figure 14. Due to clear edges at 45 s and the missing subdiagonals in the similarity matrix, all the four features can detect the variation of the chroma, but the MFCC is the least sensitive. As for octave variation, namely raising the pitch 12 semitones, the MFCC and unwrapped chromagram can detect it, but the key strength and wrapped chromagram cannot. Consequently, the wrapped chromagram and key strength ignore the octave information, but the unwrapped chromagram retains it.



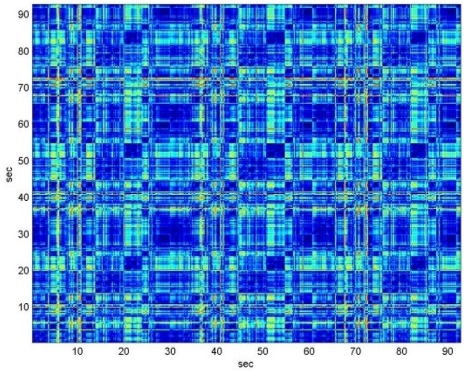
MFCC raising the pitch 4 semitones



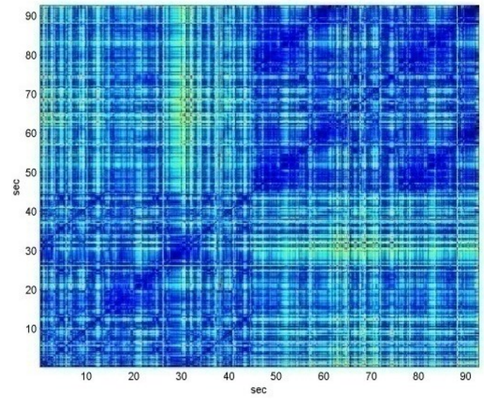
MFCC raising the pitch 12 semitones



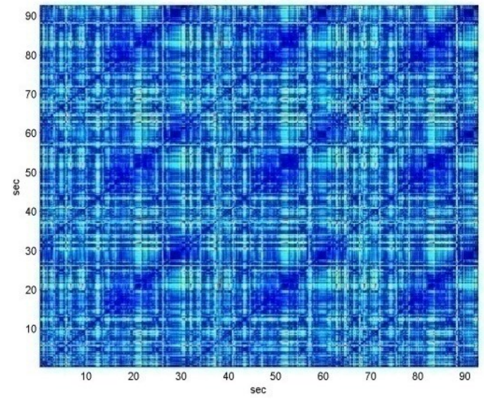
Key strength raising the pitch 4 semitones



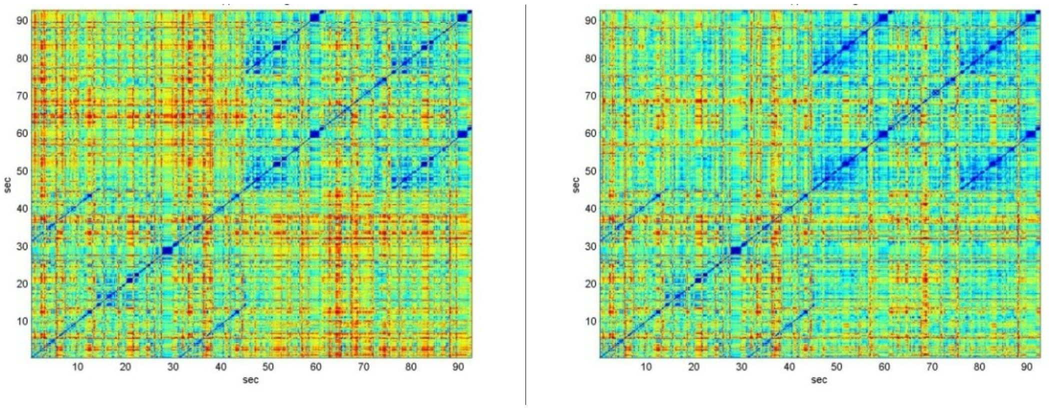
Key strength raising the pitch 12 semitones



Wrapped chromagram raising the pitch 4 semitones



Wrapped chromagram raising the pitch 12 semitones



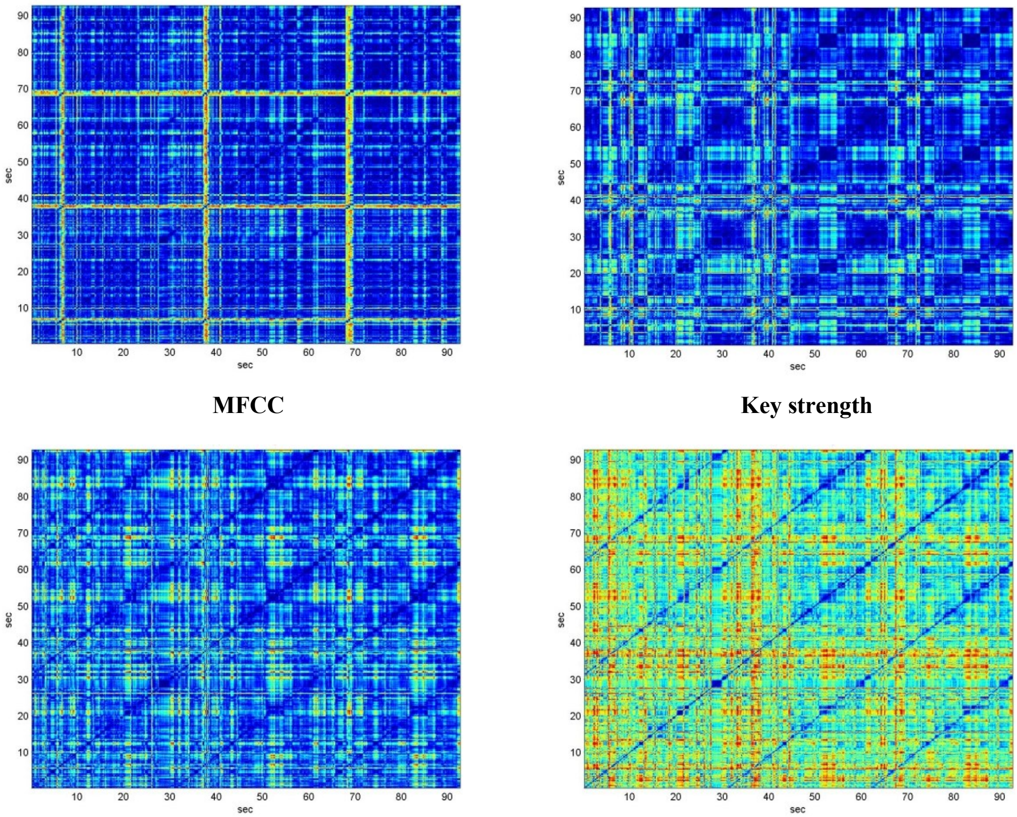
Unwrapped chromagram raising the pitch 4 semitones **Unwrapped chromagram raising the pitch 12 semitones**

Figure 14: The similarity matrix when the pitch is raised 4 semitones and 12 semitones.

3.2.2 Remove/Reduce Vocal Signals

Normally, vocal signals tend to be balanced in the left and right channels of a stereo sound, but the background accompaniment is not. Hence, to remove the vocal signals, a track should be recoded to stereo. The simplest method is to subtract the left channel from the right or vice versa. However, in general, the assumption about the balance of the vocal signals between the two channels and the imbalance of the background accompaniment is not true. Hence, a more sophisticated method to remove vocal signals is to adopt a band stop filter to filter or suppress the signals of the human voice.

In this study, the subtraction method was sufficient for our goal, and hence, we did not apply any filter. The vocal signals of the demo track after 45 s were removed by the subtraction method. Figure 15 shows the similarity matrices. All the features retain the sub-diagonals' structure at 30 and 60 s. Therefore, we may conclude that all the features are insensitive to the removal of the vocal signals.



MFCC

Key strength

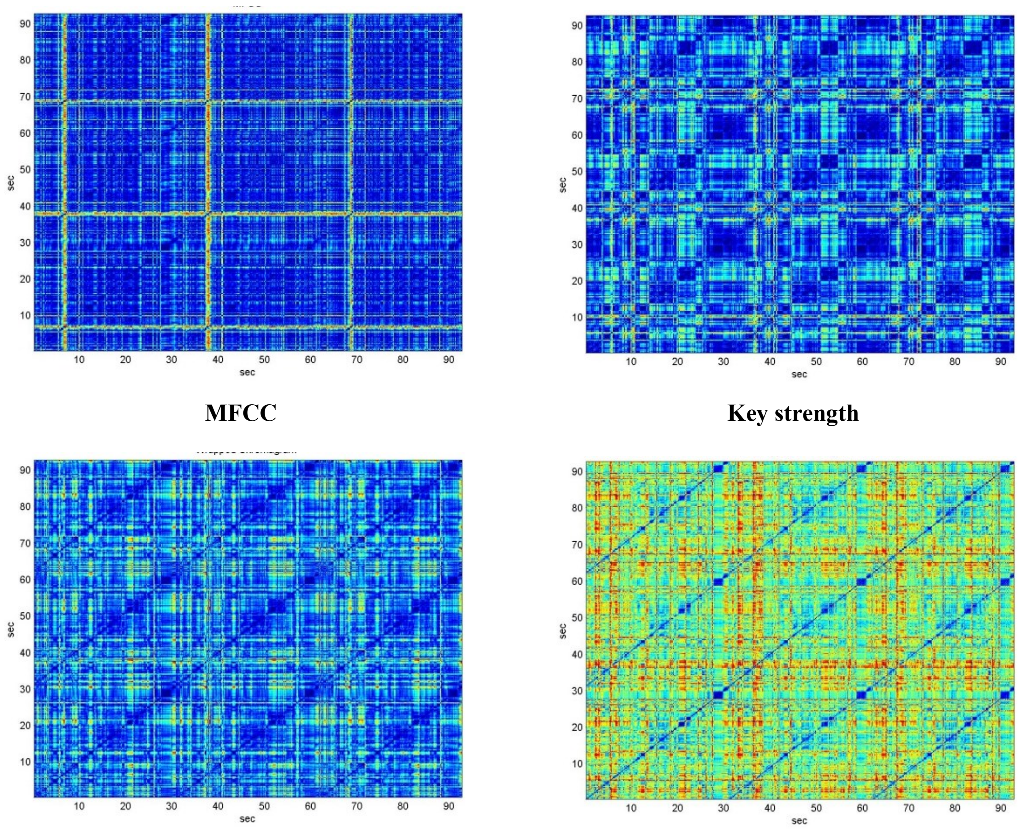
Wrapped chromagram

Unwrapped chromagram

Figure 15: The similarity matrix of reducing vocal signals after 45 s of the demo track.

3.2.3 Wahwah Effect

We also ran an interesting effect, the Wahwah effect provided by Audacity, after 45 s of the demo track, in order to check if a feature that detects the change in a sound can be as sensitive as a human's perception. In the manual of Audacity, it is stated that "[Wahwah] is using bandpass filter to create its sound. A low frequency oscillator is used to control the movement of the filter throughout the frequency spectrum. Adjust the phase of the left and right channels when given a stereo selection, so that the effect seems to travel across the speakers." Although the two channels of the signals are transferred into a single channel by taking the average, the sound with the Wahwah effect is still completely different from that without it. The Wahwah effect was applied on the demo track after 45 s and the results of the similarity matrixes are shown in Figure 16. The results are similar to those when the vocal signals were reduced, showing that these features are not sensitive to the Wahwah effect.



MFCC **Key strength**

Wrapped chromagram **Unwrapped chromagram**

Figure 16 The similarity matrixes of the Wahwah effect after 45 s

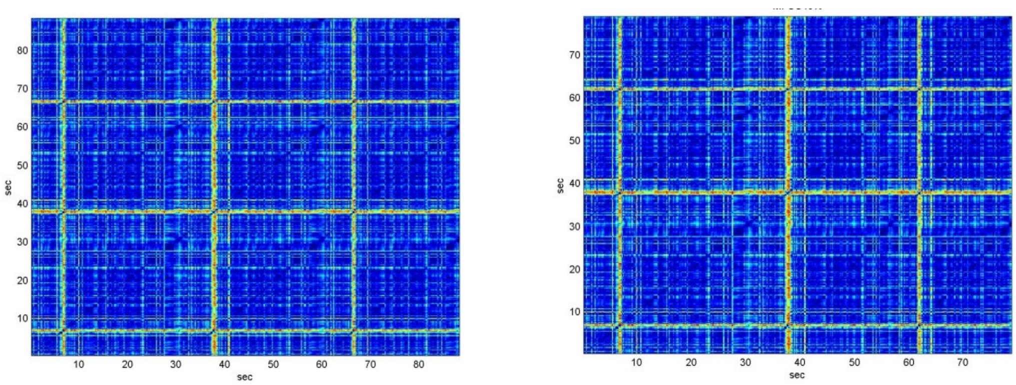
3.2.4 Tempo Variation

The tempo of a recoded song can be changed without changing the pitch, using Audacity. The algorithm for increasing or decreasing the tempo of a recoded song without changing the pitch can be found in [41]. It is called the Overlap-add (OLA) algorithm. Briefly, a song is segmented into overlapping frames, for example, 30 ms with 50% overlap. To increase the tempo, some frames are removed or repositioned/realigned while keeping the STFT of the desired signals as close as possible to the original or the envelope of the squeezed song as similar as possible to the envelope of the original by searching the nearby frames that are “maximally” similar to the removed one.

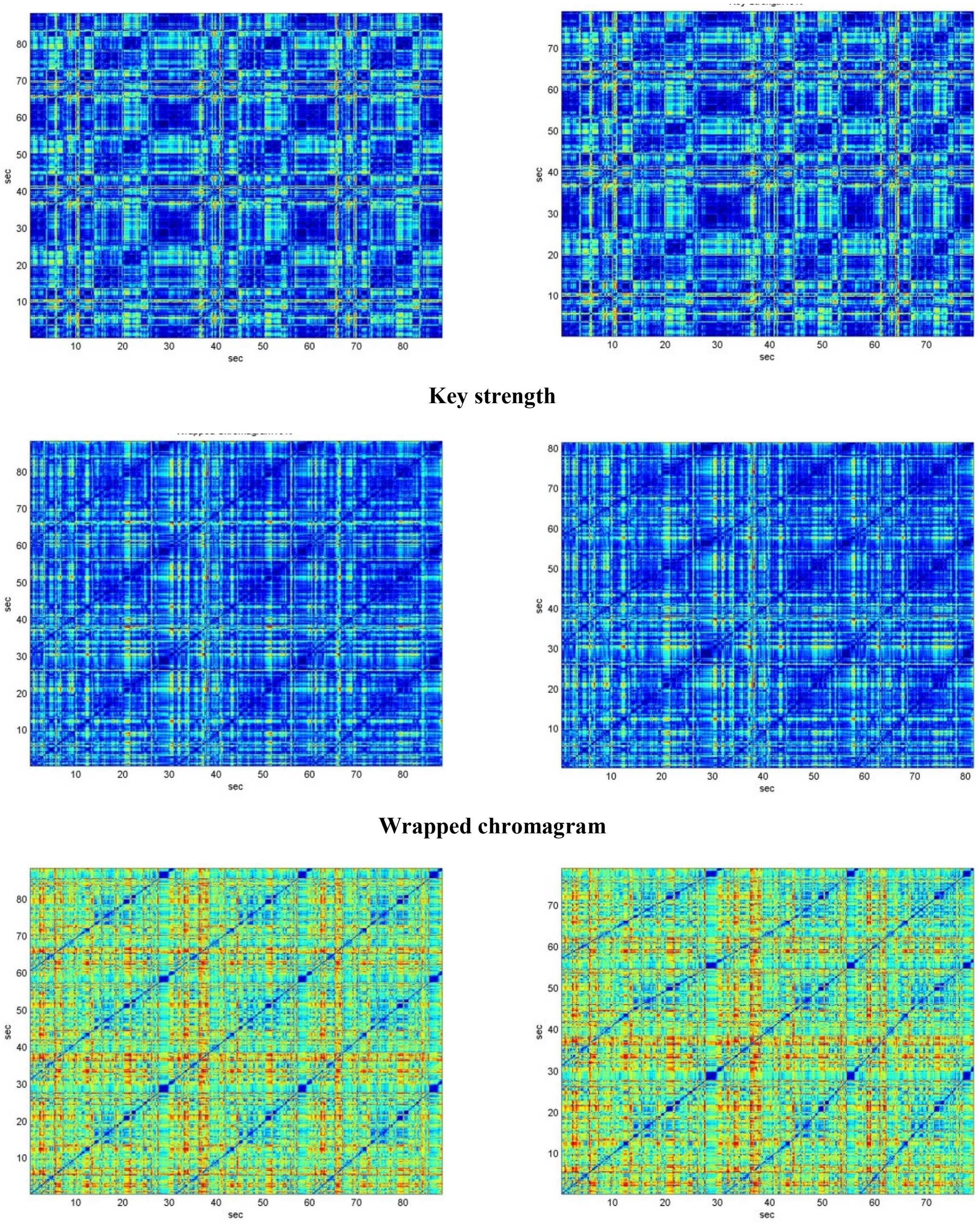
$$y(n) = \frac{\sum_k v(n - L_k + \Delta_k) x(n + \tau^{-1}(L_k) - L_k + \Delta_k)}{\sum_k v(n - L_k + \Delta_k)}$$

where $v(n)$is the window function, L_k is the window position, and Δ_k is the tolerance shift factor. The numerator shows that the input signals are segmented into frames at $\tau^{-1}(L_k) + \Delta_k$ and repositioned L_k, and then added together to form the output signal. Instead of segmenting at exactly $\tau^{-1}(L_k)$, shifting within Δ_k is acceptable in order to obtain the maximal similarity.

We increased the tempo 10% and 40% to examine the appearance of the similarity matrixes. Figure 17 shows that as the tempo increases, the slope of the subdiagonal due to the repeating sections will also change. Hence, the information about the tempo variation, without altering the timbre of a song, is kept in the time-domain scale regardless of the feature.



MFCC



Key strength

Wrapped chromagram

Unwrapped chromagram

Figure 17 Similarity matrix of increased tempo. Left: increase of 10% after 45 s; right: increase of 40% after 45 s

To examine whether the fluctuation strength can really capture the rhythm, we used the same song sung by Lollipop as before, but kept only the first 30 s without replication.

Figure 18 shows the figures of the fluctuation pattern as the tempo increases. As we expected, the dominant tempo moves from the 4 Hz to 5.7 Hz due to the faster rhythm.

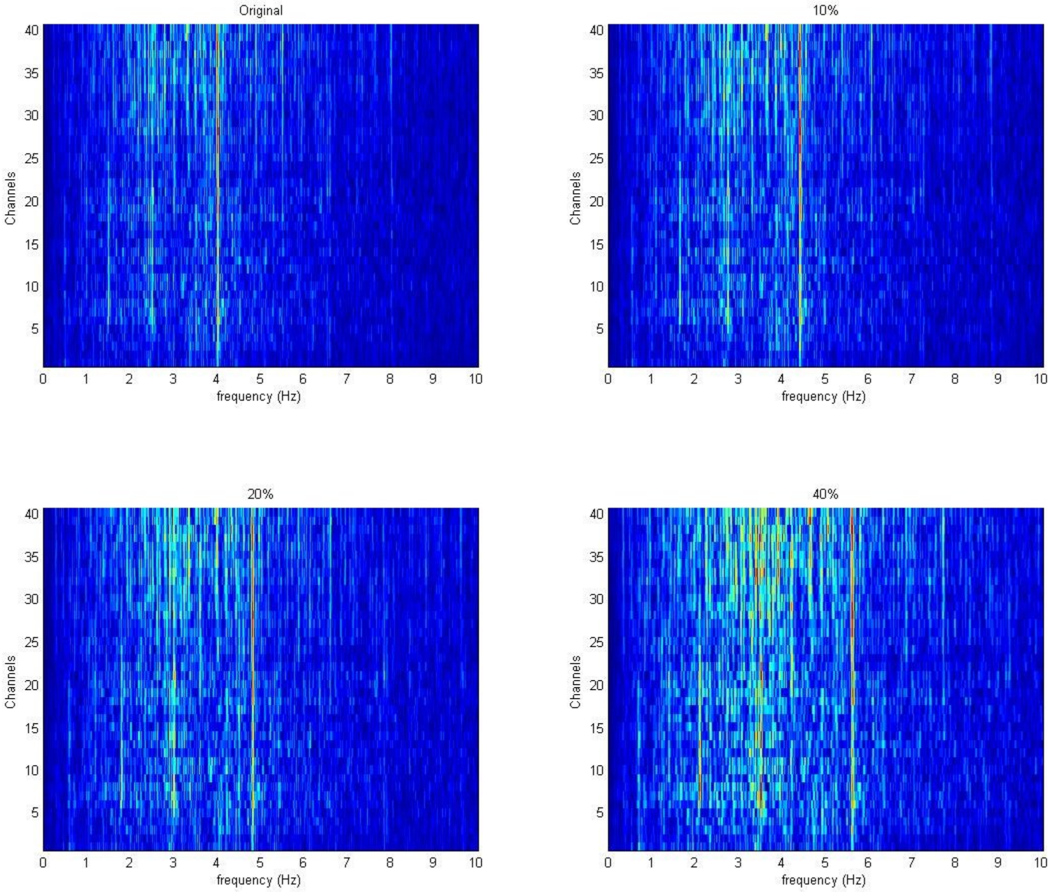


Figure 18 Fluctuation pattern: The tempo of the demo song is increased by up to 40% of the original tempo.

A summary of the results in this section is shown in Table 1.

Table 1 Summary of the features' response to the tempo variation

Feature Effect	MFCC	Key strength	Wrapped chromagram	Unwrapped chromagram
Raise 4 Semitones	Not Sensitive	Sensitive	Sensitive	Sensitive
Raise 12 Semitones	Sensitive	Not Sensitive	Not Sensitive	Sensitive

Reduce Vocal	Not Sensitive	Not Sensitive	Not Sensitive	Not Sensitive
Wahwah	Not Sensitive	Not Sensitive	Not Sensitive	Not Sensitive
Increase Tempo	Slope Change	Slope Change	Slope Change	Slope Change

3.3 Music Feature Analysis by Clustering

3.3.1 K-means

K-means is an algorithm for clustering *M* samples in to *K* groups where *K* <*M*. It minimizes the intra-cluster variance, that is, $\sum_{i=1}^{K}\sum_{x\in S_i}(x_i-\mu_i)^2$, where there are *K* groups S_i , $i = 1 \ldots K$, and μ_i is the centroid of the samples x_i belonging to S_i.

K-means clustering is sensitive to the number of clusters (*K*). We will show in section 3.3.4.1 that k-means might not be a good method for clustering songs due to the difficulty of selecting the best *K* if the number of genres or clusters cannot be decided in advance.

3.3.2 Fuzzy C-mean

Fuzzy c-means is similar to k-means, but, instead of giving a hard decision of data belonging to a cluster, it gives a soft decision that is the degree of data belonging to each cluster. For each sample, x_i, there is a coefficient that gives the degree of the sample in the j^{th} cluster, say u_{ij}.

The goal of fuzzy c-means is to minimize the objective function

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{i,j}^{m}\left\|x_i - c_j\right\|^2,$$

where *m* is a real number greater than 1, *N i*s the number of data, and *C* is the number of clusters.

The center of a cluster is the means of all points, weighted by its degree of belonging to the cluster, which is $C_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_x u_{ij}^m}$

The degree of data x_i in a cluster j is defined as $u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|}\right)^{\frac{2}{m-1}}}$.

The algorithm is designed to update C_j and u_{ij} iteratively, such that $max_{ij}\{u_{ij}^{k+1} - u_{ij}^k\} < \varepsilon$.

When *m* is equal to 2, u_{ij} is the probability of sample x_i in cluster j; otherwise, u_{ij} is proportional to the probability of sample x_i in cluster *j*. When *m* is close to 1, the cluster center closest to the point is given much more weight than the others, and the algorithm is similar to k-means.

3.3.3 Self-organizing Map (SOM)

An SOM, which projects the high dimensional data onto a lower dimensional map, contains neurons represented by the weight vectors. The dimension of weight vector $\boldsymbol{m} = [m_1 \ldots m_d]$ is the same as the dimension of the feature vector $\boldsymbol{x} = [x_1 \ldots x_d]$. The best-matching unit (BMU) is defined as the neuron closest to the feature vector, that is, $\|x - m_c\| = \min_i \|x - m_i\|$. SOM is similar to the k-means algorithm. However, not only is the best-matching weight vector updated but also the neighboring weight vectors are moved toward the feature vectors such that the neighboring neurons have similar weight vectors.

The sequential training algorithm for the weight vector m_i of unit i is

$$m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]$$

where α is the time varying learning rate, and h_{ci} is the time-varying neighborhood kernel. The neighborhood kernel is a non-increasing function of time. c and i represent the winner unit c and neighboring unit i. There are several different choices of neighborhood functions, for example, bubble, Gaussian, etc. However, in general, Gaussian is used [42]. The neighborhood function will shrink with time [19].

The algorithm is

1. Randomize the neurons or the weight vectors of the map;
2. Select an input feature vector;
3. Calculate the distance between the input vector and the weight vector for each neuron in the map. Then, find the BMU, that is, find the neuron among all neurons in the map having the shortest distance;
4. Update the neighborhood of the BMU by moving them closer to the input vector $m_i(t+1) = m_i(t) + \alpha(t)h_{ci}(t)[x(t) - m_i(t)]$;
5. Repeat from step 2 until *t* reach the preset criteria.

Normally, a clustering result of SOM is represented by a unified distance matrix (U-matrix). The U-matrix visualizes the distances between neighboring map units. A good cluster should have a valley structure, that is, a lower neighborhood-distance area surrounded by high neighborhood-distance edges. However, when the data gathered at a region of the map can still form a cluster, even the high neighborhood-distance edge does not exist. In general, an area with larger values of the U-matrix indicates that the variation within a cluster is larger and vice versa.

The component plane shows the value (not distance) of one component (x_i) from a feature vector $x = [x_1 \dots x_d]$ in each map unit. The component plane can imply the discriminability of a component related to the others in the feature vectors for an application.

3.3.4 Results

In the following subsections, 12 albums and more than 100 songs were prepared and used. The information of these songs, including the singer, player, and genre, is provided in

in the Appendix. Every song was segmented into 0.5-s frames, and the mean and the variance of the features was calculated to represent a song.

3.3.4.1 K-means

Xu, Maddage, and Tian [43] claimed that the variance of MFCC 3 can distinguish between vocal and instrumental music. Since the k-means algorithm is sensitive to the choice of k, the ANOVA/ Tukey method was applied before k-means in an attempt to examine how these albums are distributed given the features' statistics results of the songs.

Figure 19 shows the multiple comparison of the variance of MFCC 3 at the 10% level. MFCC 3 can roughly separate these albums into four groups. The first group includes Classical, Choirboys, Piano, and Violin. Except for Choirboys, the first group contains only instrumental music, but String is misclassified into the second group. The majority of the second group is music with vocals, including Sarah, Morrissey, Mariah, String, and Usher. The remaining albums are in the third and fourth groups.

Hence, we assume that to cluster the vocal and instrumental music the best choice of the value of k is 4 for the variance of MFCC 3. Figure 20 shows the misclassification of the vocal and instrumental music vs. the different choice of *k* using the variance of MFCC3 on k-means. When *k* is larger than 2, a cluster is defined as an instrumental music cluster when most of the songs in a group are instrumental music, and the vocal music in this group is set misclassified, and vice versa. Although *k* should be set equal to 2 normally, to separate vocal from instrumental music, and this definition of misclassification might be problematic, the results still show the discriminability of the variance of MFCC 3 and suggest that it is not easy to determine the number of clusters, *k*, without any prior knowledge.

According to the above, setting *k* = 4 is better than setting arbitrary *k*. In addition, the outcome of k-means is sometimes sensitive to the initial choice of the centroid. Therefore, 150 replications are required here. The number of misclassified songs is 25 for *k* = 4 and the centroids of the k-means are at 96.89, 319.23, 759.75, and 1395.12. Finally, the majority of misclassifications are for Choirboys.

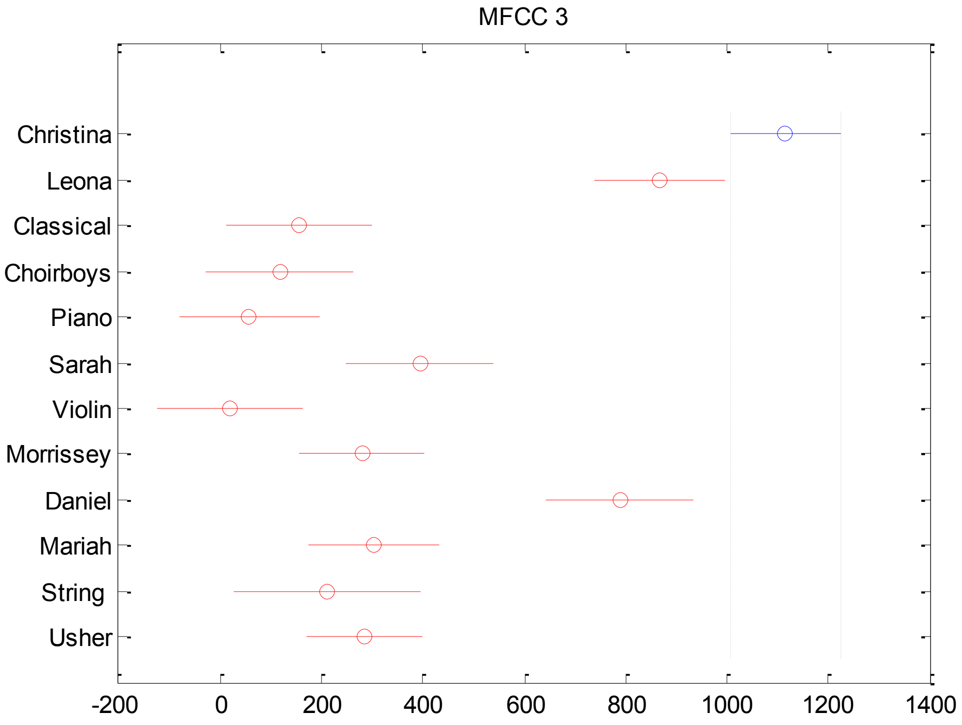


Figure 19 Multiple comparison of the variance of MFCC

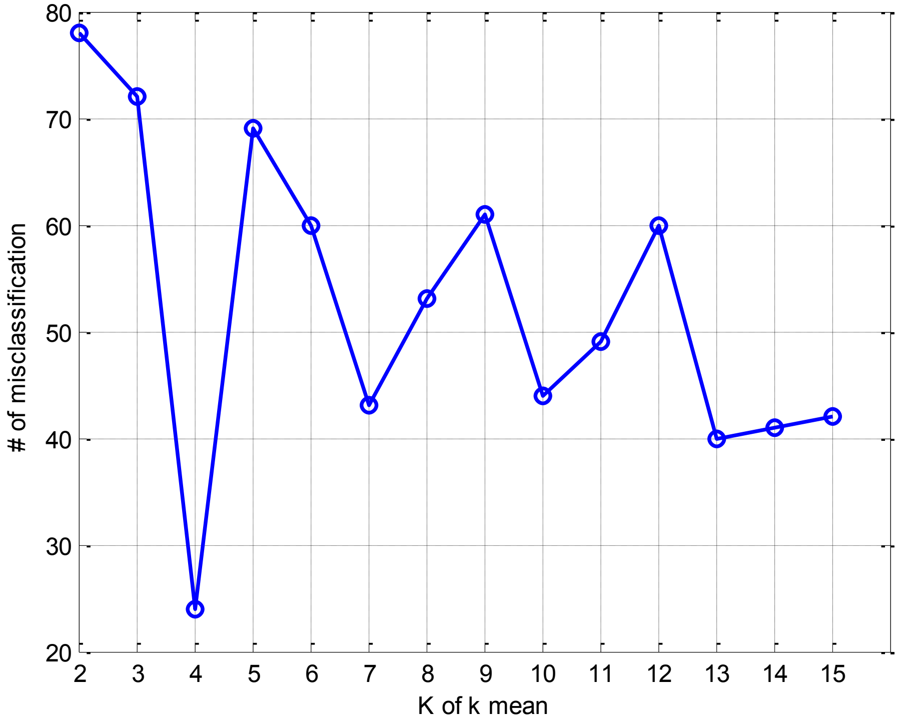


Figure 20 Misclassifications of the vocal and pure music vs. *k*

Figure 21 shows the multiple comparison of variance of MFCC1 and MFCC2. The lower order of the MFCCs should contain more information about the mel spectrum due to the property of the DCT in the final step of MFCC. However, it is very difficult to tell from Figure 21 which properties MFCC1 or MFCC2 can discriminate.

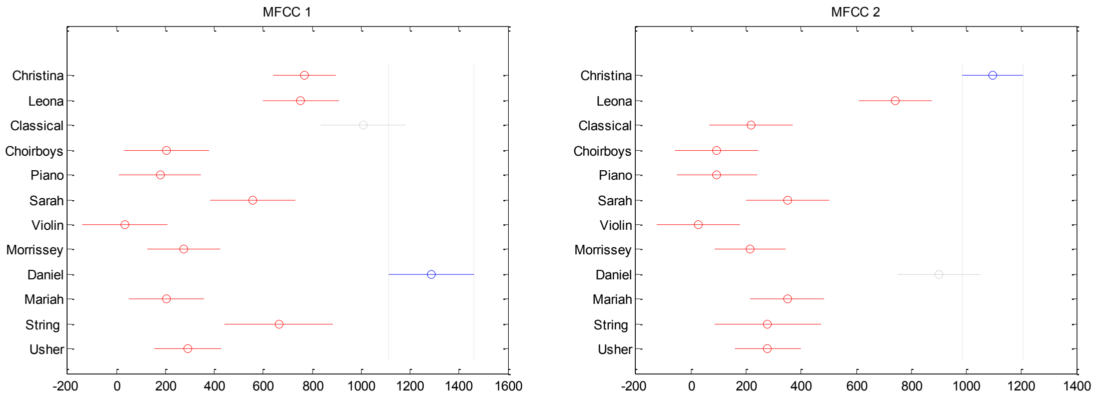
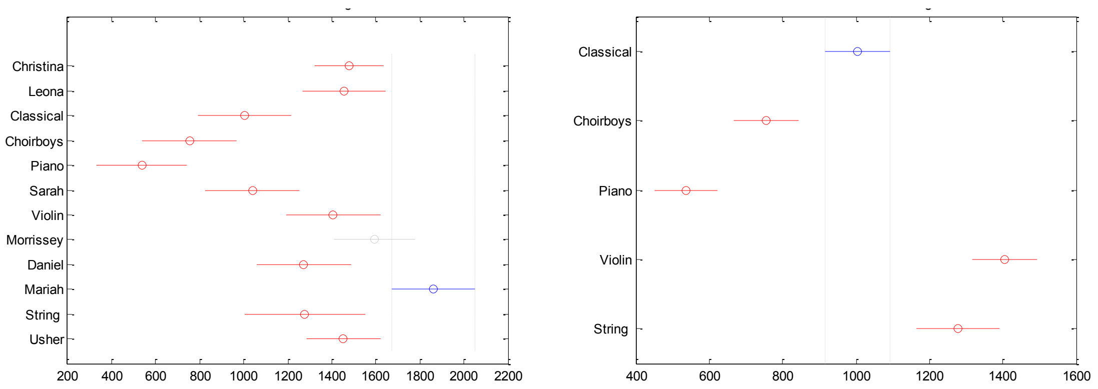


Figure 21 Multiple comparison of variance of MFCC1 and MFCC2

The multiple comparison of the mean ZCR in Figure 22(a) suggests that it discriminates between Symphony (labeled as Classical) melodies played by the Piano, and Violin well. Since the violin is a string instrument, it is reasonable that the mean ZCR cannot separate them well. Furthermore, the mean ZCR is not a suitable feature for separating vocal and instrumental music. In Figure 22 (b) only the multiple comparison of instrumental music and songs sung by Choirboys is shown. The figure implies that the mean ZCR can discriminate between the instruments, including Choirboys. Finally, the results of the k-means by setting the initial centroids at 500,700,1000, and 1300 for these selected albums show that ten songs are misclassified. The centroids are actually at 542.71, 789.2, 1082.3, and 1527.7. Almost all the misclassifications are of Violin and String, since the centroid of Violin and String (1527.7) is too high.



(a) All albums

(b) Selected albums only

Figure 22: Multiple comparison of the mean ZCR

Figure 23 shows the multiple comparison of variance ZCR. A comparison of Figure 22 and Figure 23 shows that the discriminability of instruments is better when using the mean ZCR than when using the variance ZCR.

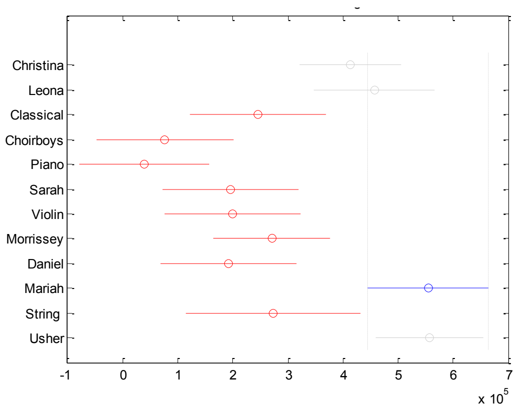


Figure 23 Multiple comparison of the variance ZCR

3.3.4.2 Fuzzy C-mean

In this subsection, 77 songs and 7 albums given in Table 6 in the Appendix are analyzed. The features and their dimension are given in Table 2.

Table 2 The dimension of the features

Feature	Dimension
Spectral Centroid Overall Standard Deviation	1
Root Mean Square Overall Standard Deviation	1
Zero Crossings Overall Standard Deviation	1
MFCC Overall Standard Deviation	13
Spectral Centroid Overall Average	1
Root Mean Square Overall Average	1
Zero Crossings Overall Average	1
MFCC Overall Average	13

Table 3 is the clustering result of fuzzy c-means with k = 6 and m = 2. The highest probability cluster (or hard decision) of each song is given in the “In cluster” column, which shows that cluster 1 tends to Violin, cluster 3 tends to Male vocal, cluster 2 tends to Female vocal, cluster 4 and cluster 6 tend to Symphony or Choirboys, and cluster 5 tends to Piano. In addition, some of the hard decisions of Sarah Brightman’s songs are in cluster 4, which tends to Symphony, but the probability is not as high as that of Symphony. The second highest probability of these songs (Sarah Brightman's) is in cluster 3, which corresponds to Male vocal. This may be because Sarah Brightman’s songs contain both vocals and background accompaniment similar to Symphony.

Table 3 Clustering result of fuzzy c-means with k = 6 and m = 2

Symphony

Song index	1	2	3	4	5	6	7	8	9	10	11
Cluster 1	0.043	0.0148	0.0158	0.0459	0.1085	0.0488	0.0625	0.0479	0.039	0.0336	0.0507
Cluster 2	0.0371	0.0115	0.0119	0.0183	0.0326	0.0157	0.0399	0.0261	0.0199	0.0136	0.0467
Cluster 3	0.0855	0.0279	0.0304	0.0286	0.0446	0.027	0.0825	0.0575	0.0359	0.0231	0.1081
Cluster 4	0.6176	0.8539	0.8557	0.1137	0.1125	0.1315	0.4844	0.6481	0.2593	0.1079	0.5264
Cluster 5	0.085	0.0365	0.0342	0.3004	0.3412	0.1454	0.108	0.076	0.2371	0.144	0.096
Cluster 6	0.1318	0.0554	0.0519	0.4931	0.3606	0.6316	0.2227	0.1444	0.4087	0.6778	0.1721
In Cluster	4	4	4	6	6	6	4	4	6	6	4
Piano											
Song index	12	13	14	15	16	17	18	19	20	21	22
Cluster 1	0.0196	0.0356	0.0275	0.1119	0.0114	0.0697	0.0243	0.0402	0.0931	0.0317	0.0365
Cluster 2	0.0086	0.0138	0.0103	0.037	0.0047	0.0271	0.0097	0.0194	0.0333	0.0147	0.0161
Cluster 3	0.0134	0.0199	0.0145	0.0472	0.0072	0.0367	0.0149	0.0334	0.0446	0.0233	0.0244

Cluster 4	0.0552	0.0594	0.0427	0.101	0.0266	0.0895	0.0537	0.1784	0.1042	0.0913	0.0789
Cluster 5	0.7563	0.713	0.6931	0.4345	0.871	0.5638	0.7127	0.4654	0.342	0.632	0.6812
Cluster 6	0.147	0.1583	0.212	0.2684	0.0792	0.2132	0.1847	0.2632	0.3828	0.207	0.1628
In Cluster	5	5	5	5	5	5	5	5	6	5	5
Violin											
Song index	23	24	25	26	27	28	29	30	31	32	33
Cluster 1	0.1315	0.2261	0.3515	0.167	0.7804	0.9039	0.8302	0.9449	0.8899	0.8631	0.8941
Cluster 2	0.05	0.0708	0.0526	0.0378	0.0296	0.0117	0.0236	0.0075	0.0136	0.0214	0.0136
Cluster 3	0.0995	0.122	0.0755	0.0622	0.0368	0.0132	0.0256	0.0084	0.0153	0.0225	0.0161
Cluster 4	0.3585	0.2518	0.153	0.2139	0.0507	0.0192	0.0341	0.0114	0.022	0.0275	0.0232
Cluster 5	0.1118	0.1129	0.111	0.1432	0.0409	0.0215	0.0352	0.0114	0.0244	0.0269	0.0205
Cluster 6	0.2486	0.2164	0.2565	0.3759	0.0617	0.0305	0.0514	0.0165	0.0348	0.0385	0.0325
In Cluster	4	4	1	6	1	1	1	1	1	1	1
Male vocal											

Song index	34	35	36	37	38	39	40	41	42	43	44
Cluster 1	0.0698	0.0278	0.0295	0.0384	0.0164	0.0276	0.0248	0.0373	0.0948	0.0436	0.0242
Cluster 2	0.3682	0.0808	0.2308	0.0903	0.1017	0.0798	0.1213	0.0839	0.5111	0.1885	0.2927
Cluster 3	0.3745	0.7197	0.6361	0.5281	0.8119	0.737	0.7389	0.5535	0.1864	0.5988	0.5961
Cluster 4	0.0899	0.1058	0.0535	0.228	0.0381	0.0934	0.0638	0.2193	0.0836	0.0907	0.0447
Cluster 5	0.041	0.0267	0.02	0.0461	0.0128	0.0249	0.0194	0.0422	0.0531	0.032	0.0169
Cluster 6	0.0567	0.0392	0.0301	0.0691	0.0191	0.0372	0.0318	0.0638	0.071	0.0464	0.0254
In Cluster	3	3	3	3	3	3	3	3	2	3	3
Female vocal											
Song index	45	46	47	48	49	50	51	52	53	54	55
Cluster 1	0.0189	0.1036	0.0375	0.0349	0.0515	0.117	0.0906	0.0149	0.041	0.0182	0.0236
Cluster 2	0.0883	0.4824	0.6453	0.2738	0.5779	0.2719	0.4689	0.0881	0.7053	0.7711	0.2296
Cluster 3	0.7799	0.183	0.1995	0.5317	0.2081	0.2197	0.1952	0.8242	0.1516	0.1535	0.6482

Cluster 4	0.0656	0.089	0.0523	0.0818	0.0712	0.1488	0.0955	0.0405	0.0446	0.0271	0.052
Cluster 5	0.0186	0.061	0.0261	0.0296	0.0362	0.0871	0.0569	0.0122	0.0237	0.0122	0.0185
Cluster 6	0.0288	0.0809	0.0393	0.0483	0.0551	0.1555	0.0929	0.0199	0.0338	0.0179	0.0281
In Cluster	3	2	2	3	2	2	2	3	2	2	3
Choirboys											
Song index	56	57	58	59	60	61	62	63	64	65	66
Cluster 1	0.0317	0.0237	0.0249	0.029	0.0287	0.044	0.0234	0.025	0.0486	0.0295	0.0156
Cluster 2	0.0108	0.0092	0.0112	0.0195	0.0105	0.0239	0.0102	0.0097	0.0197	0.02	0.0114
Cluster 3	0.0161	0.0135	0.0171	0.0443	0.0163	0.0509	0.0166	0.015	0.0271	0.0449	0.0266
Cluster 4	0.0559	0.0447	0.0607	0.7497	0.0633	0.594	0.0718	0.0546	0.0755	0.6951	0.8328
Cluster 5	0.4357	0.1487	0.1092	0.0628	0.1292	0.1203	0.2754	0.6325	0.2605	0.0846	0.0449
Cluster 6	0.4497	0.7602	0.7768	0.0947	0.7521	0.1669	0.6026	0.2633	0.5685	0.1259	0.0686
In Cluster	6	6	6	4	6	4	6	5	6	4	4
Sarah Brightman											

Song index	67	68	69	70	71	72	73	74	75	76	77
Cluster 1	0.0465	0.0443	0.0458	0.0469	0.0533	0.0443	0.0747	0.031	0.0436	0.0321	0.0545
Cluster 2	0.078	0.1251	0.0213	0.0724	0.1282	0.0706	0.032	0.0364	0.0677	0.4662	0.0408
Cluster 3	0.3533	0.6385	0.0331	0.2089	0.5824	0.246	0.0438	0.112	0.2519	0.3964	0.0756
Cluster 4	0.3728	0.111	0.1237	0.4565	0.1374	0.4561	0.1037	0.6713	0.4669	0.0522	0.3703
Cluster 5	0.0584	0.0328	0.2397	0.0865	0.0367	0.0643	0.2035	0.0579	0.0637	0.0205	0.1357
Cluster 6	0.091	0.0482	0.5365	0.1288	0.0621	0.1187	0.5423	0.0913	0.1062	0.0326	0.3232
In Cluster	4	3	6	4	3	4	6	4	4	2	4

3.3.4.3 SOM

In the following section, the analysis of 15 albums and 188 songs is described. The information of the albums is in Table 8 in the Appendix. Compared to Table 7, in Table 8, two jazz and one rap album are added.

The results of clustering the songs by MFCC, ZCR, unwrapped chromagram, and fluctuation pattern are discussed in sections 3.3.4.3.1 to 3.3.4.3.4. The best feature set for clustering the songs is a combination of MFCC and ZCR as shown in section 3.3.4.4.

3.3.4.3.1 MFCCs

Figure 24 and Figure 25 show the unified distance matrix (U-matrix) of the MFCC mean. The labels in Figure 24 and the numbers in Figure 25 are the albums' abbreviations and the song indexes, respectively. Hence, Figure 24 shows the albums' distribution on the map, whereas the distributions of the songs are shown in Figure 25. In Figure 24, the instrumental music and choirboys' songs are congregated in the upper left. In addition, the songs of the violin and piano are not far apart. The genre of jazz gathers at the lower left. Classical and string songs are aggregated at the center. The songs sung by females tend to gather in the upper part of the U-matrix, while those sung by males tend to gather in the lower part. Furthermore, the variation or the distance among the neurons is larger for the vocal than for the instrumental music. In addition, a clear edge is shown at the upper right of the U-matrix. This suggests that the MFCC mean may distinguish between vocal and instrumental music.

The U-matrix of the MFCC variance is shown in Figure 26. In the figure, according to the distribution of the rap songs, the MFCC variance seems to have a better ability to classify the songs than the MFCC mean. The songs played by the piano and violin form a group. Choirboy's songs gather together. Classical music is also congregated. The rap songs are also captured. Female 1 and 2 are at the lower left. Male 2 is at the lower right. Similarly, Male 1 and Female 4 are gathered at the center, and so on. However, Figure 24, Figure 25, and Figure 26 show that MFCC seems to fail to identify the songs played by the piano and violin. Figure 27 and Figure 28 are also the U-matrix of the MFCC mean and variance, respectively, but the size of the U-matrix is reduced from 20×10 to 8×5 to show more clearly that it is not sufficiently accurate to distinguish the piano and violin songs.

The component planes of the MFCC variance are shown in Figure 29. The songs sung by females tend to have a higher value of the variance, and the instrumental music and songs sung by males tend to have a lower value.

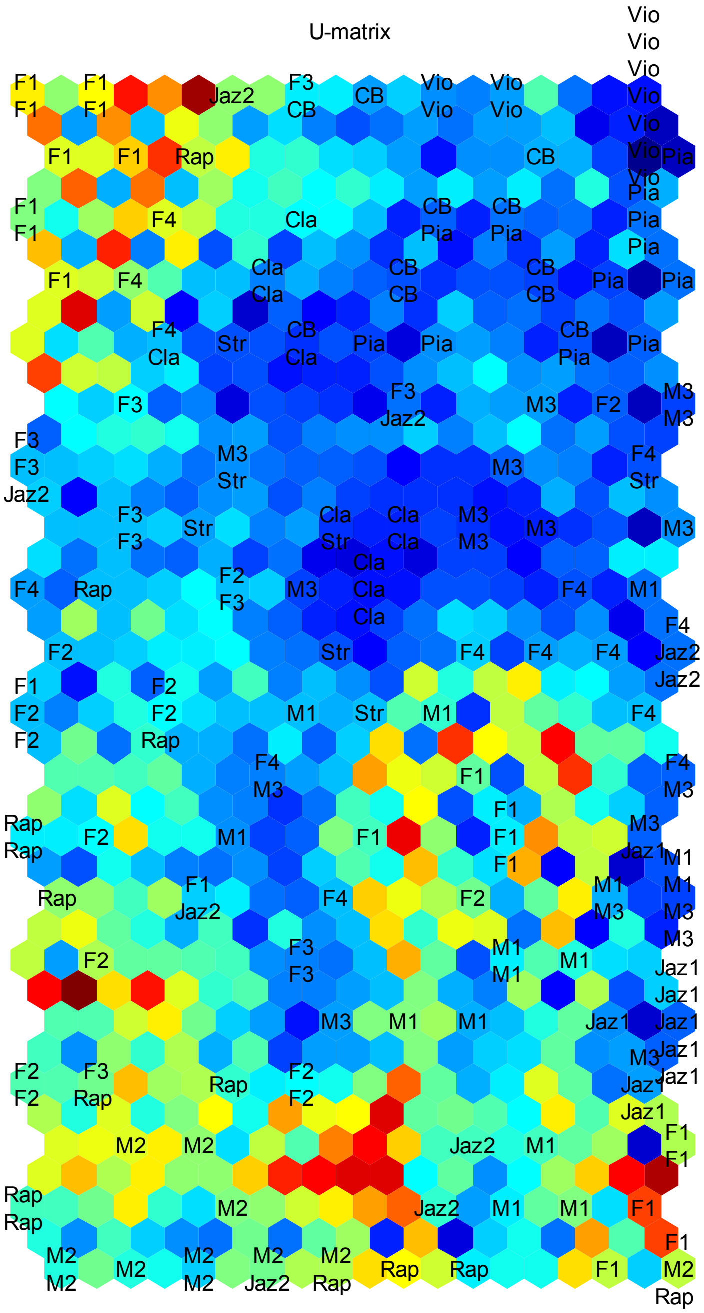


Figure 24 20 × 10 U-matrix of the MFCC mean with the album labels

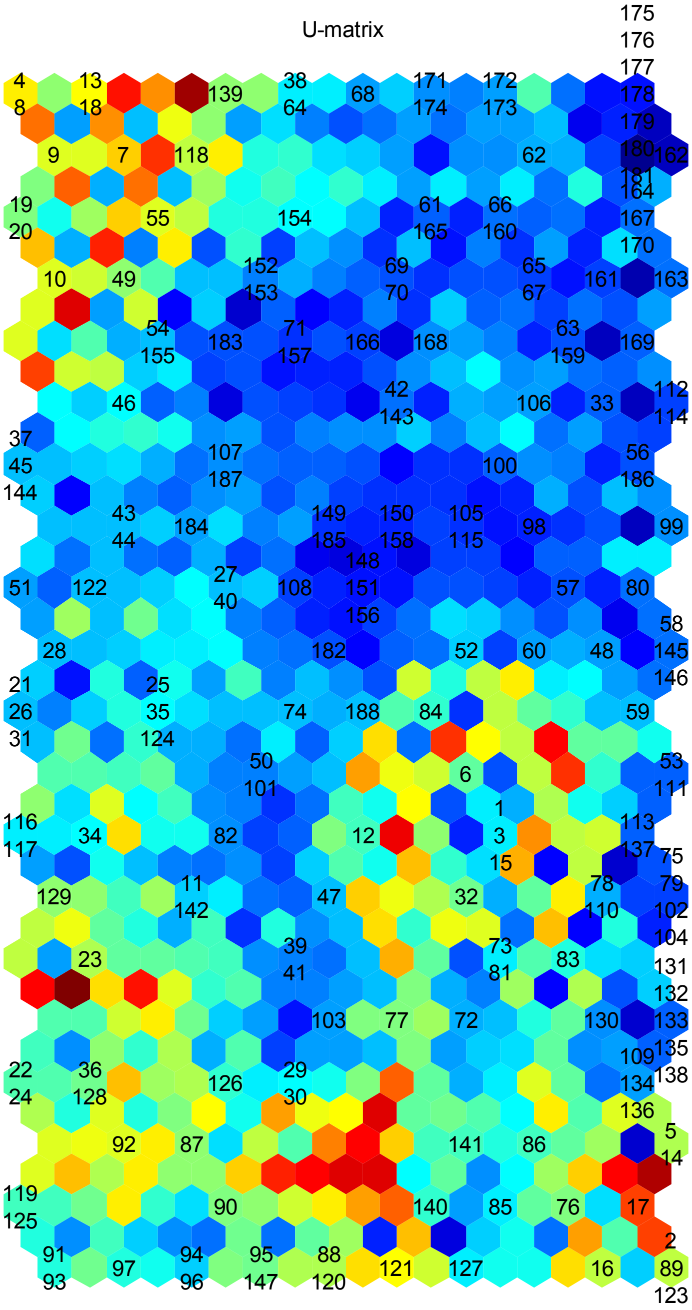


Figure 25 20 × 10 U-matrix of the MFCC mean with song index

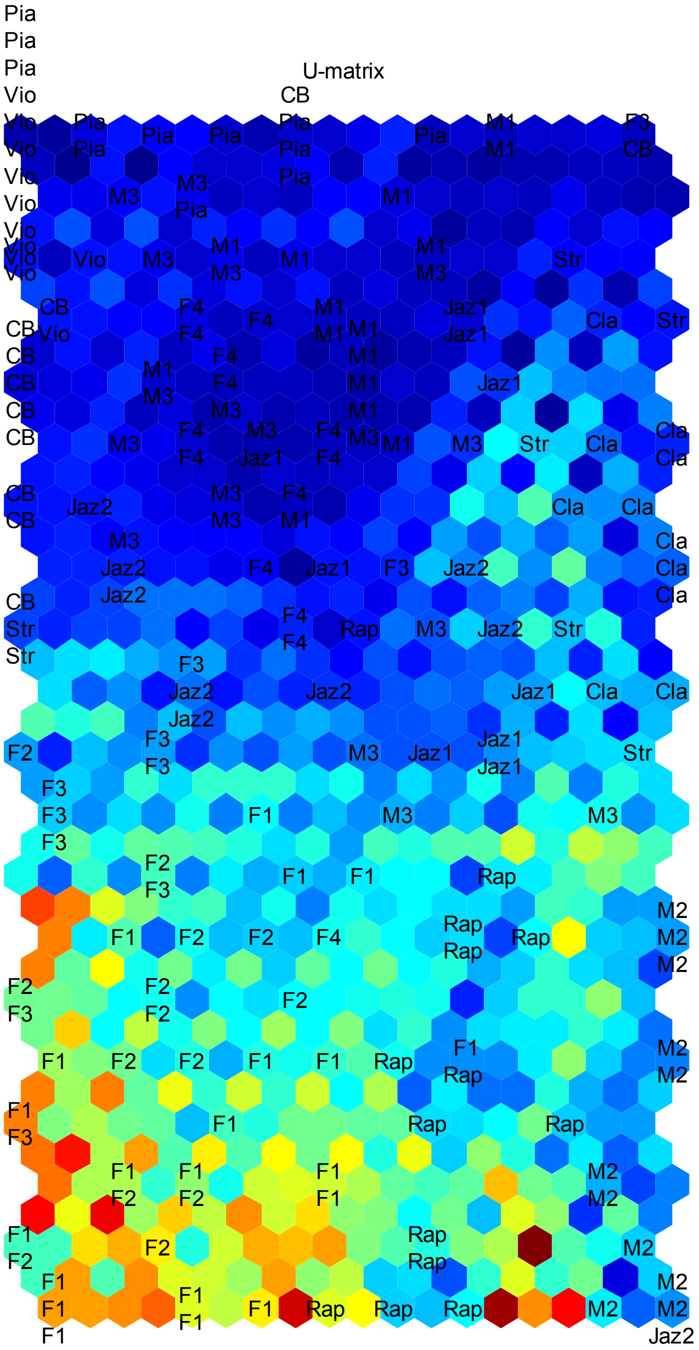


Figure 26 U-matrix of the MFCC variance with the album labels

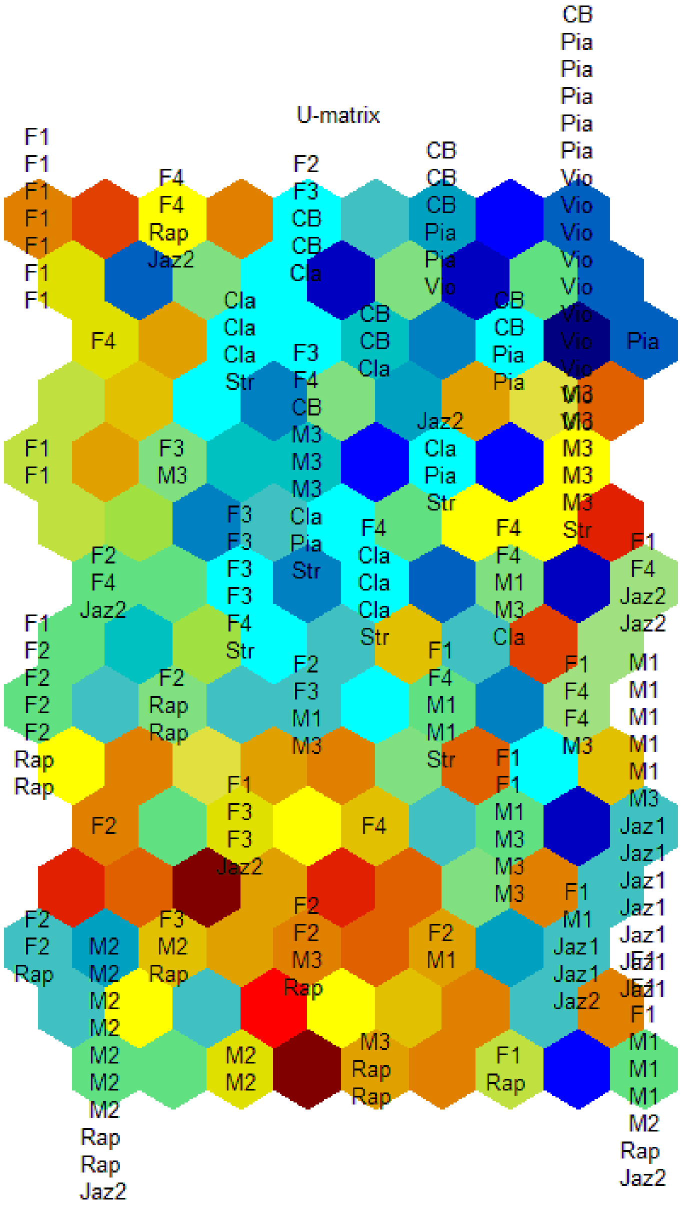


Figure 27 8 × 5 U-matrix of the MFCC mean with the album labels

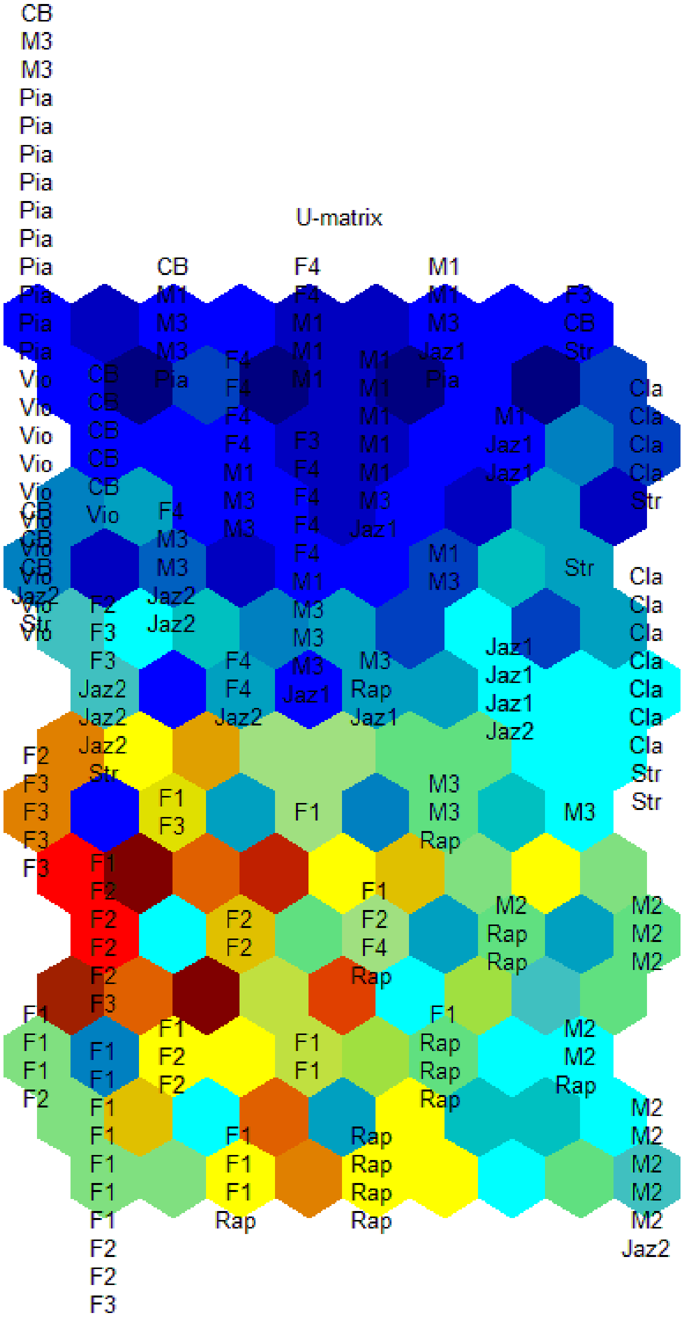


Figure 28 8×5 U-matrix of the MFCC variance with the album labels

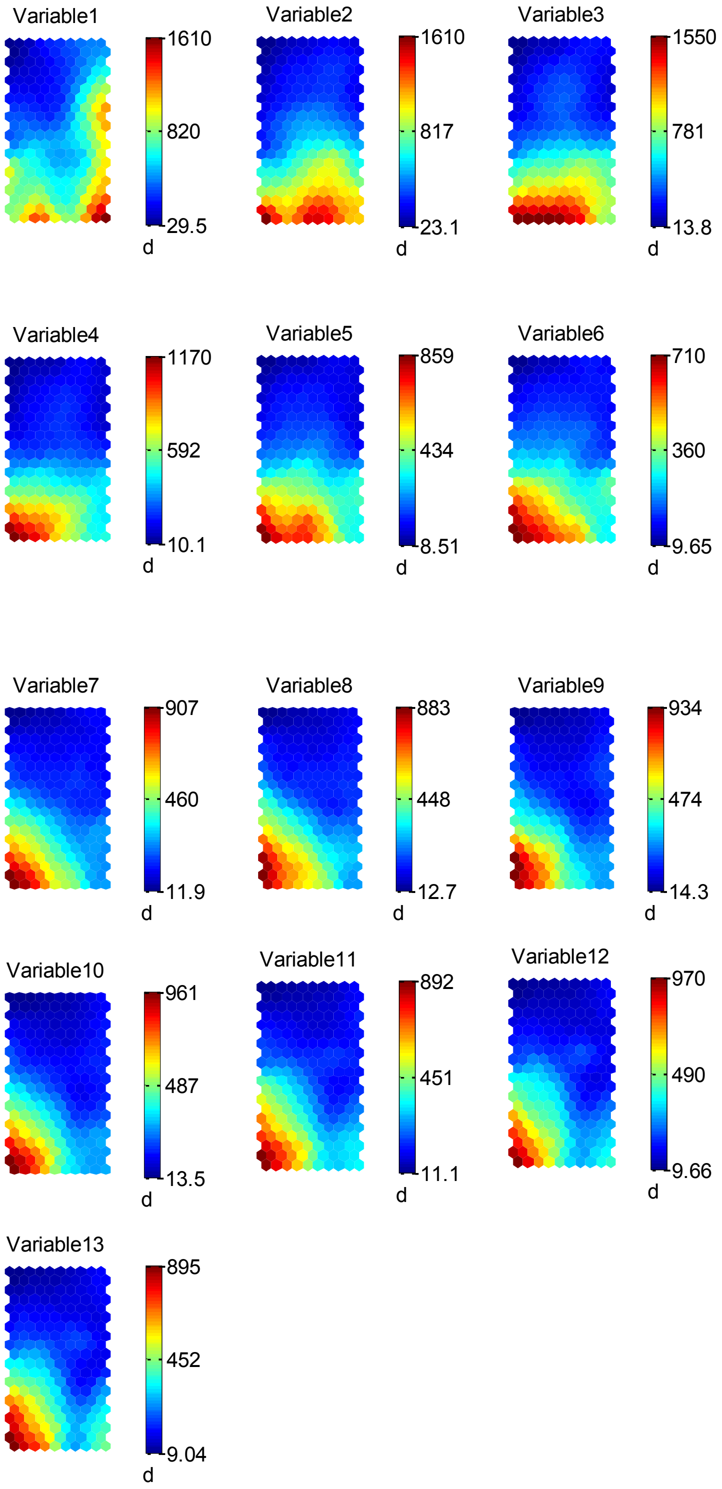


Figure 29 The component planes of the MFCC variance show that the songs sung by a female tend to have higher value of variance, but the instrumental music and songs sung by a male have a lower value.

3.3.4.3.2 Zero-Crossing Rate

The U-matrix of the mean ZCR of the instrumental music is shown in Figure 30. The songs of piano are at the upper left, the classical music is at the center, and the songs of the violin and string are distributed in the lower part of the figure. These results are coincident with the result in previous section: the mean ZCR is a suitable feature for identifying instruments. In addition, a comparison of the mean ZCR and the variance ZCR in Figure 31 shows that the mean ZCR clusters instrumental music better, especially in the case of classical music.

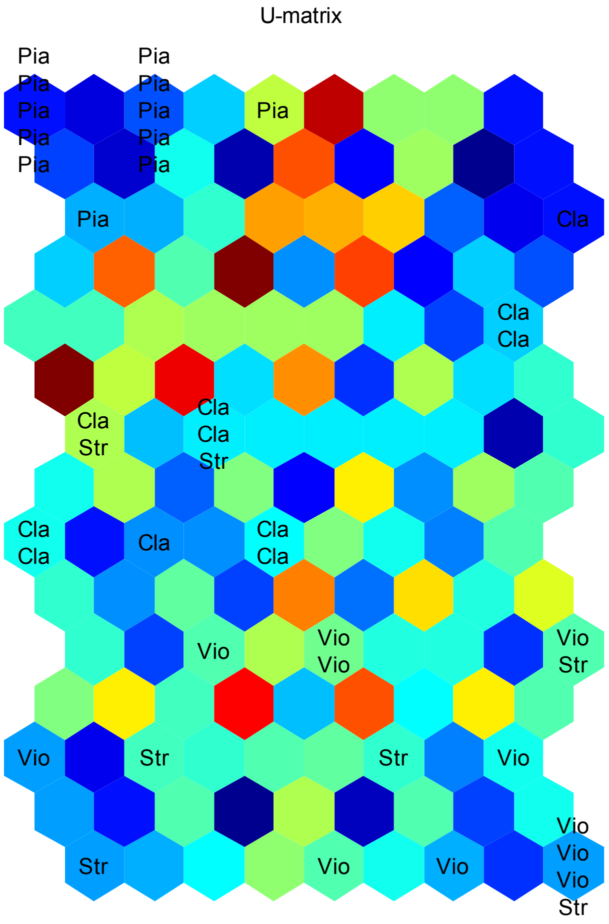


Figure 30 U-matrix of the mean ZCR for albums of instrumental music

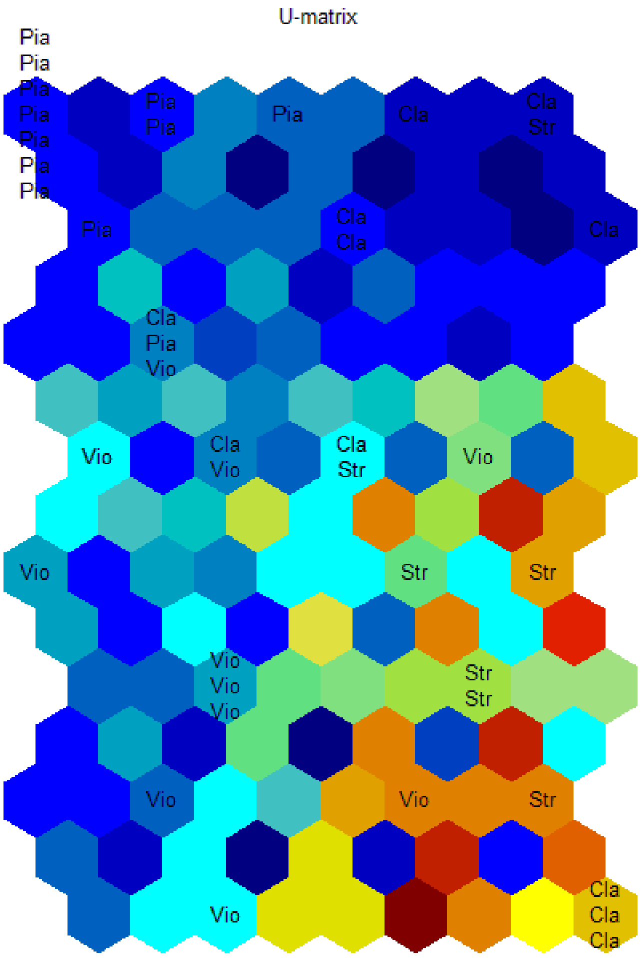


Figure 31 U-matrix of the ZCR variance for albums of instrumental music

Figure 32 also shows the U-matrix of the mean ZCR, but the albums of Jazz 1 and Jazz 2 are included. The figure shows that the mean ZCR fails to cluster jazz music. Figure 33 shows the U-matrix with the song indexes on the map. Most of Jazz 2's songs contain the sounds of a trumpet, and hence these results may suggest that the mean ZCR fails to identify the trumpet. Songs 131, 133, and 134 in Jazz 1 are played by a piano with some electrical instruments in the background, and hence it is reasonable that they are mapped close to the piano. Songs 135 and 136 in Jazz 1 contain more electrical instruments than songs 131, 133, and 134. The timbre of songs 131, 133, and 134 is different from that of songs 135 and 136, and song 138 in Jazz 1 contains a small amount of vocals. Hence, they are far apart on the map.

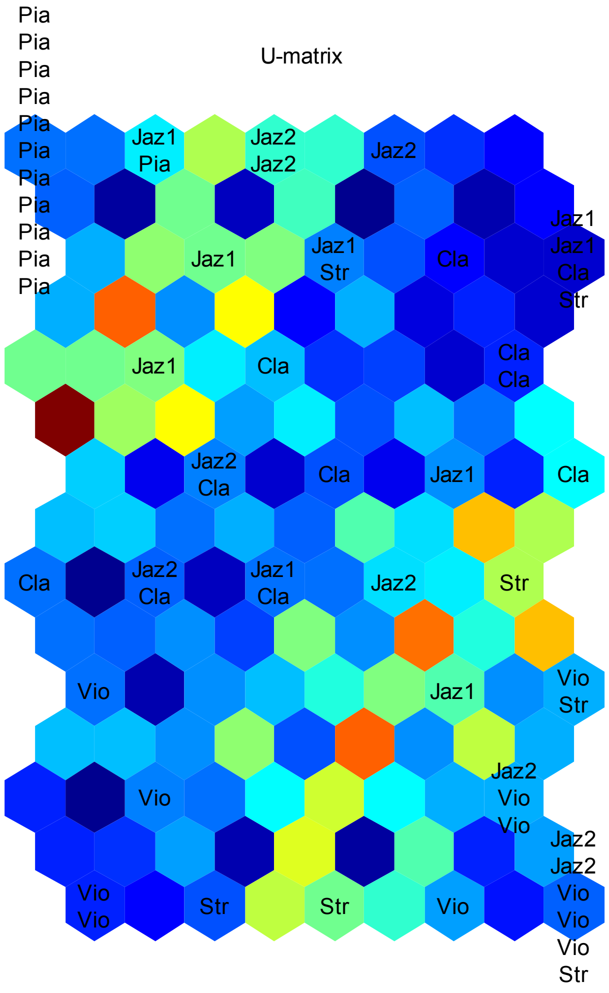


Figure 32 U-matrix of the mean ZCR including the album of instrumental music, Jazz 1, and Jazz 2 with the album labels

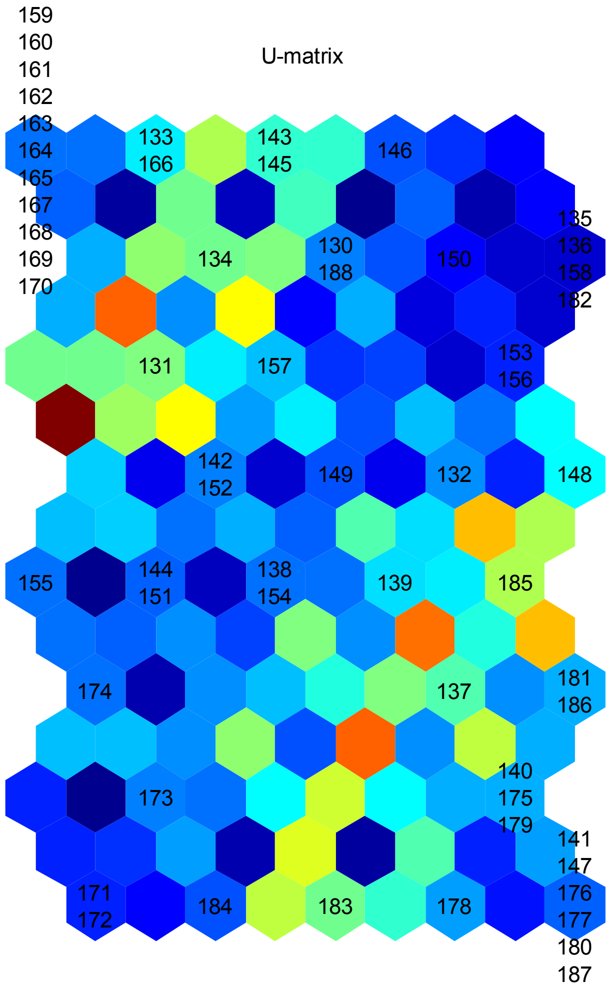


Figure 33 U-matrix of the mean ZCR including the album of instrumental music, Jazz 1, and Jazz 2 with the song index

Figure 34 and Figure 35 show the U-matrix of the mean and the variance ZCR containing all the songs, respectively. When the database containing all the songs is used, it is not very obvious which properties the mean ZCR can identify.

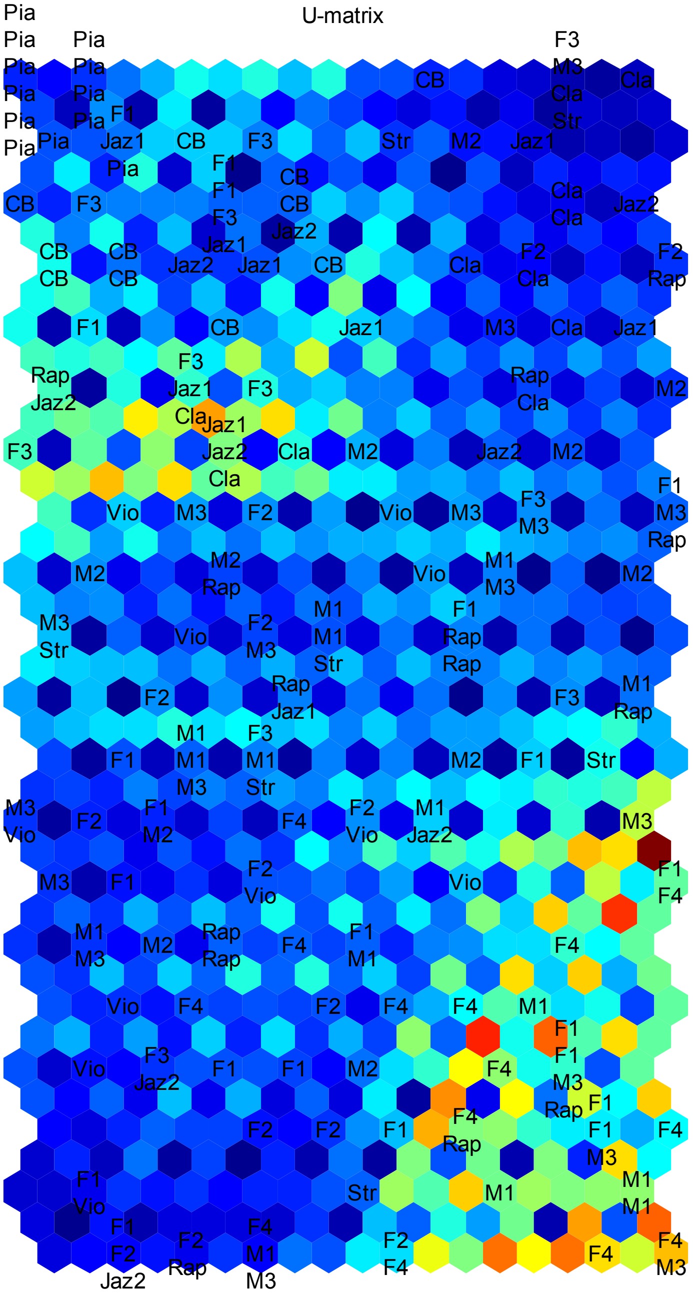


Figure 34 U-matrix of the mean ZCR including all the albums with the album labels

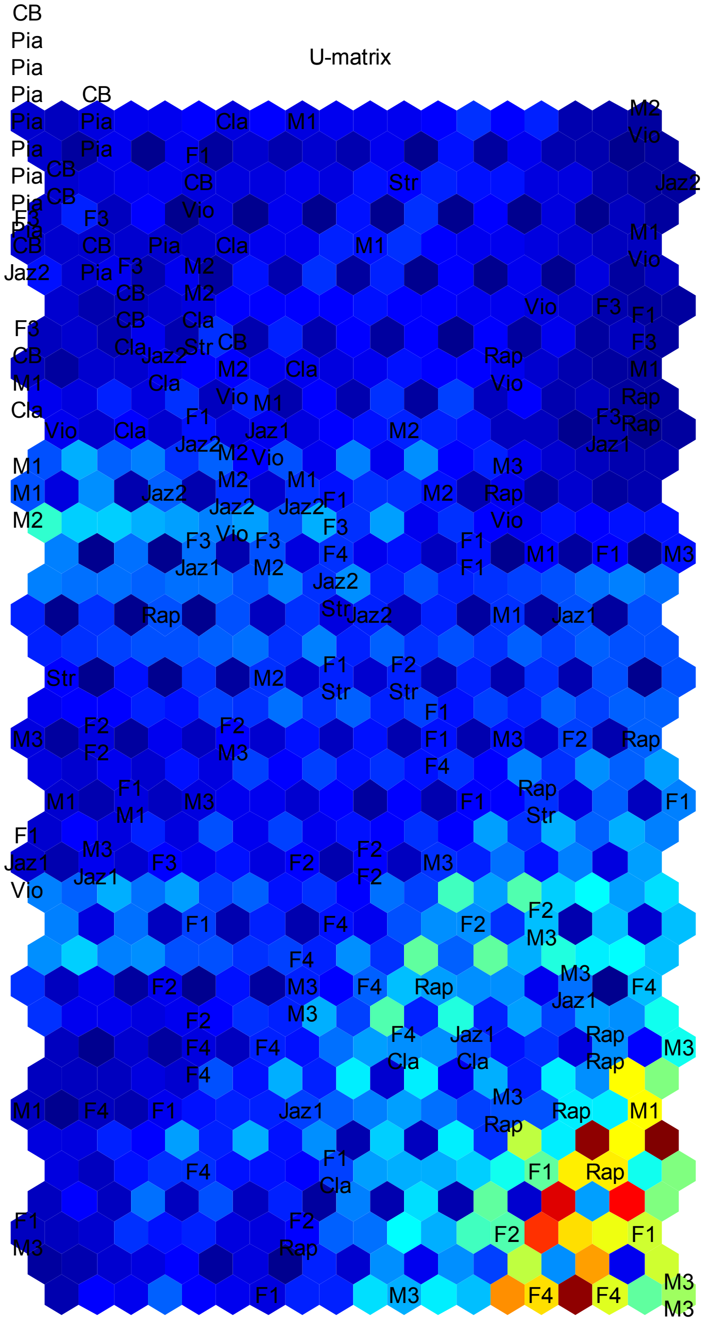


Figure 35 U-matrix of the variance ZCR including all the albums with the album labels

3.3.4.3.3 Unwrapped Chromagram

Figure 36 shows the 8 × 5 U-matrix of the mean unwrapped chromagram containing the vocal music. The songs sung by females are gathered at the left and the center; the rap songs are at the bottom and the songs sung by males are grouped at the upper and lower right.

Figure 37 is the U-matrix of the mean unwrapped chromagram of the instrumental music. It shows that the unwrapped chromagram is a good feature for identifying the instruments and genres of the instrumental music.

Figure 38 and Figure 39 are the U-matrix of the mean unwrapped chromagram of all the songs. The songs of the piano and Jazz 1 are distributed at the upper left; classical music and string and violin songs are at the upper right; the songs sung by a male are at the lower right; the songs sung by a female are at the lower left, and the rap songs are gathered at the bottom; however, the distributions of the songs of Jazz 2 are not well defined.

Although the unwrapped chromagram is designed to represent notes or chroma, it still contains information about the pitch. Hence, is not surprising that the unwrapped chromagram has the ability to cluster the genres or instruments.

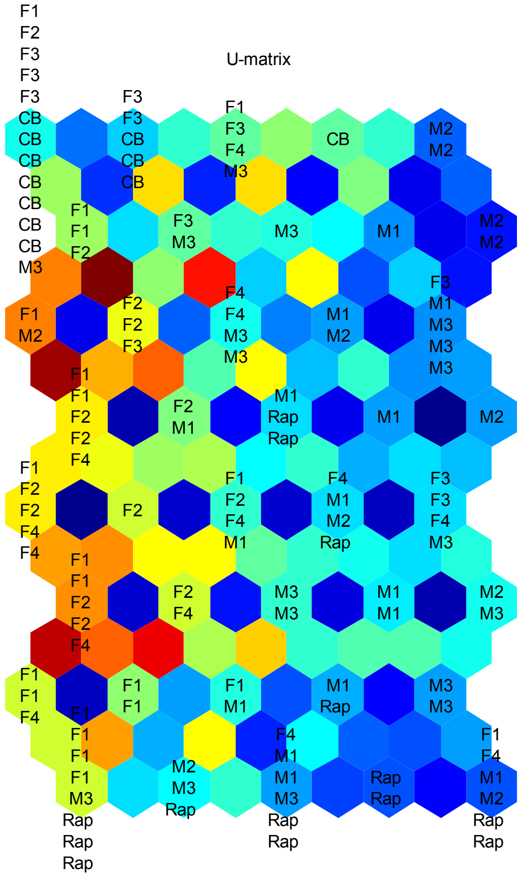


Figure 36 8 × 5 U-matrix of the mean unwrapped chromagram of vocal music

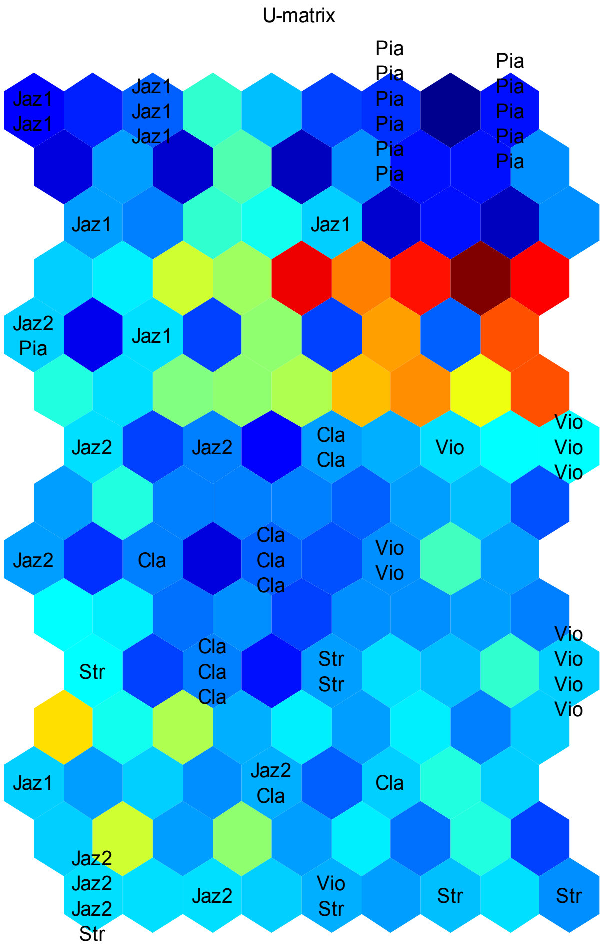


Figure 37 8×5 U-matrix of the mean unwrapped chromagram of instrumental music including Jazz

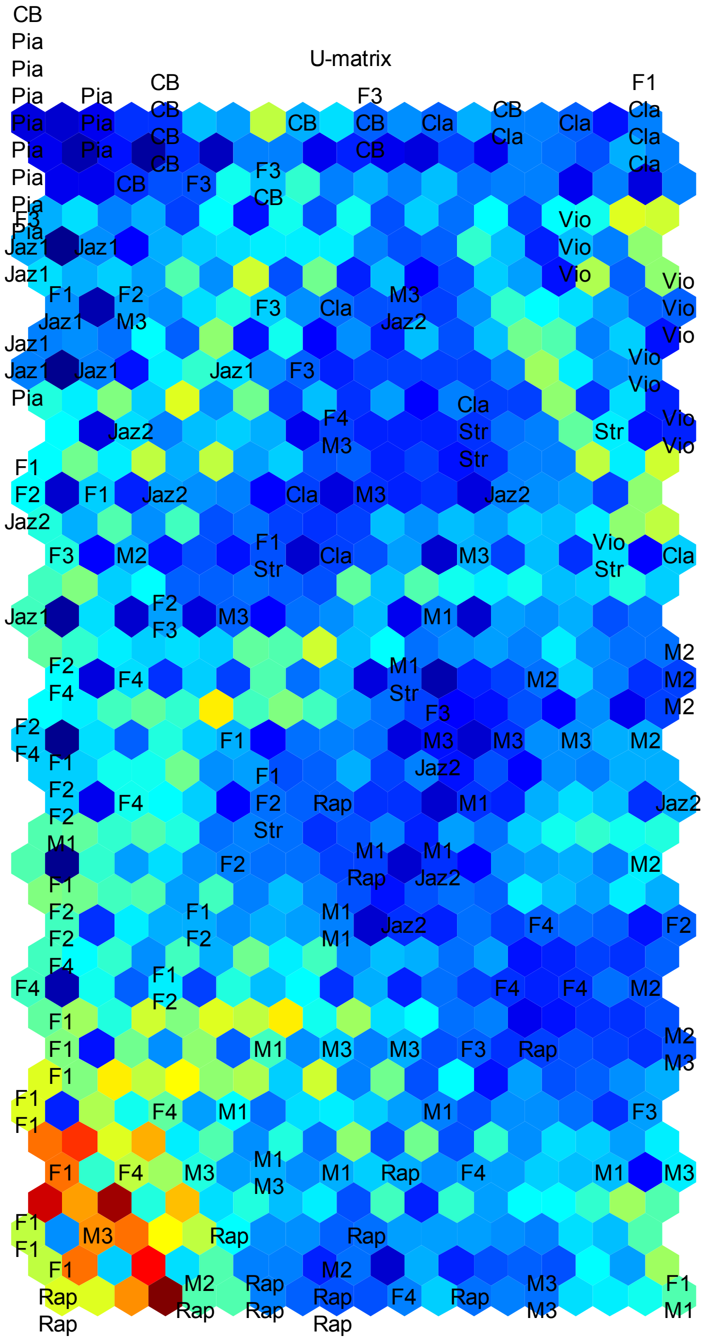


Figure 38 20×10 U-matrix of the mean unwrapped chromagram including all the songs

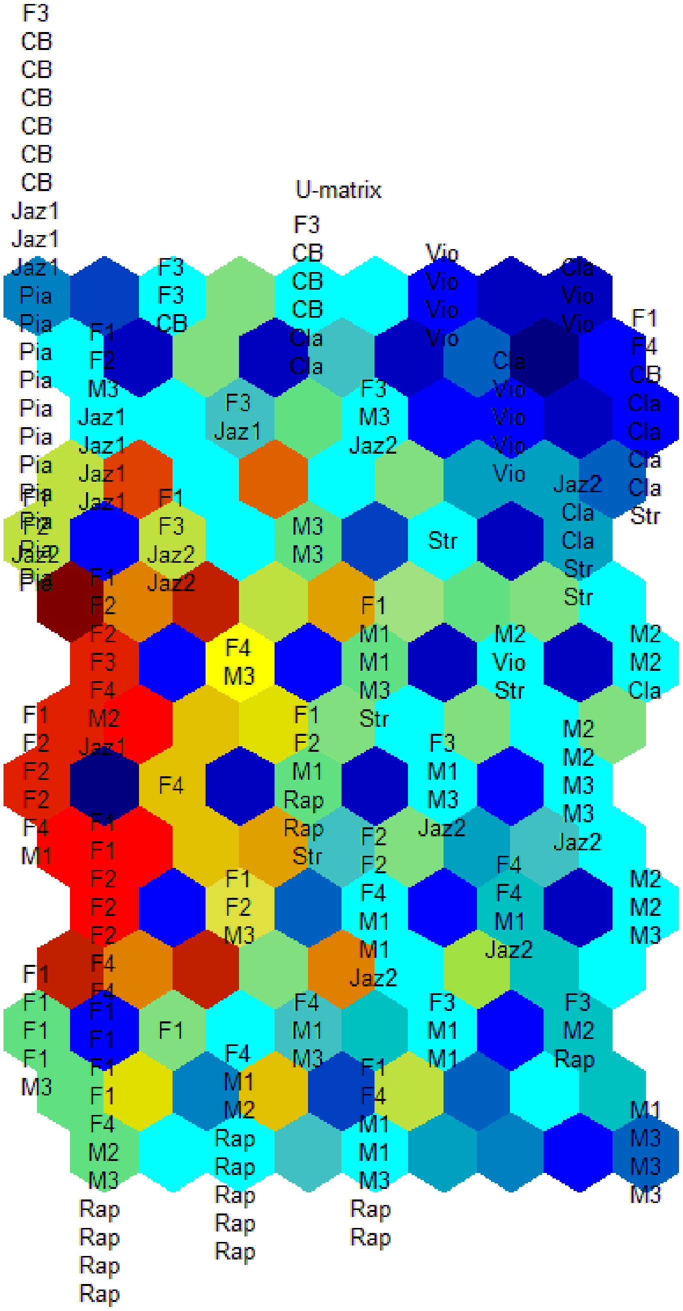


Figure 39 8 × 5 U-matrix of the mean unwrapped chromagram of the all songs

Figure 40 is the variance unwrapped chromagram U-matrix of the vocal music. In Figure 40, it can be seen that rap songs are grouped at the lower left. Most of the songs sung by women are at the lower right and center of the map, while those sung by males are at the upper left.

In Figure 41, the variance of the unwrapped chromagram of instrumental music shows that it can be used to classify genres or instruments.

The 20×10 U-matrix of the unwrapped chromagram of all the songs is shown in Figure 43. Roughly, the group of rap songs is at the lower left, the songs sung by females are at the lower right, those sung by males are at center left, and instrumental music is in the upper part of the map. The results can be read more easily by reducing the size of the U-matrix to 8×5, as in Figure 44.

By comparing Figure 40 and Figure 42, one can see that the mean unwrapped chromagram has a better ability to separate songs sung by females and males, while the variance unwrapped chromagram separates the rap songs better.

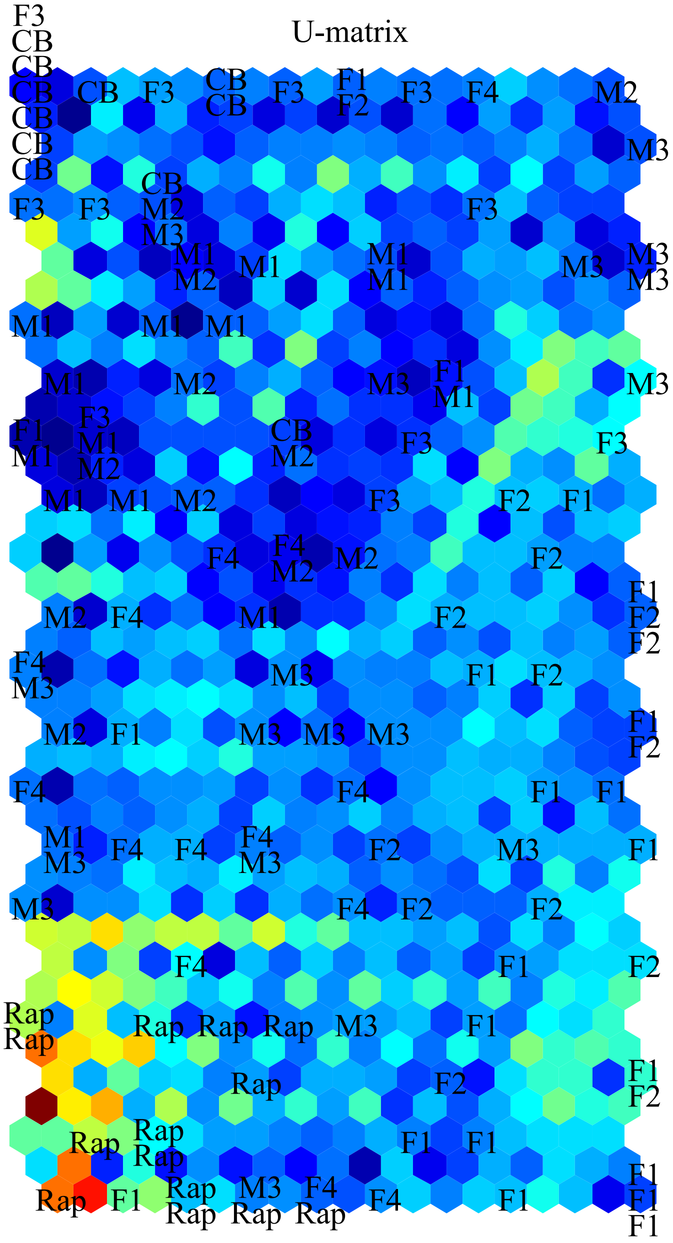


Figure 40 : 20×10 U-matrix of the variance unwrapped chromagram of the vocal music

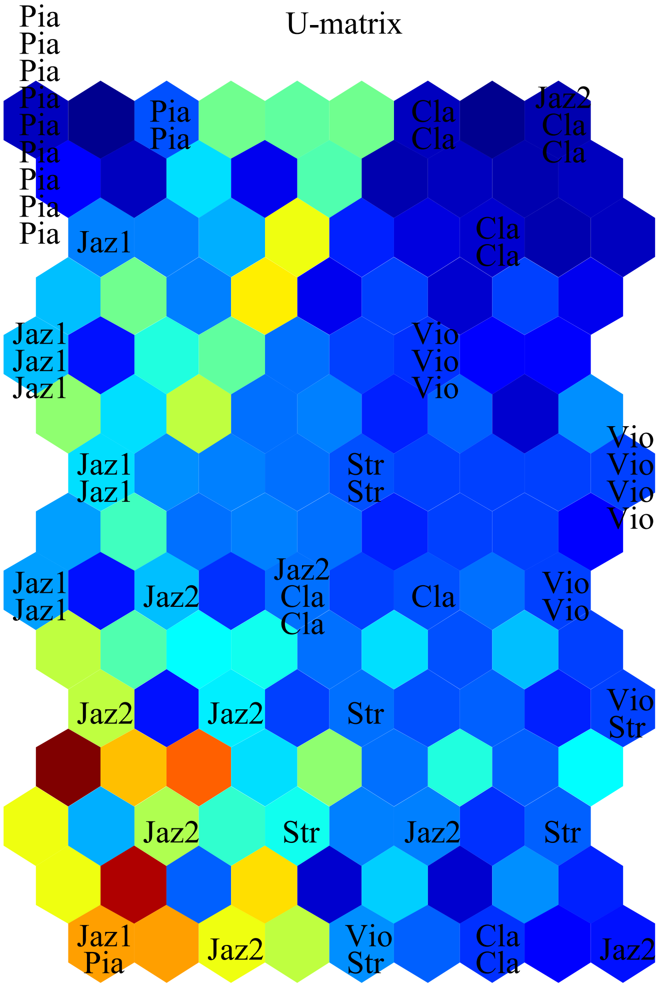


Figure 41: 8×5 U-matrix of the variance unwrapped chromagram of the instrumental music

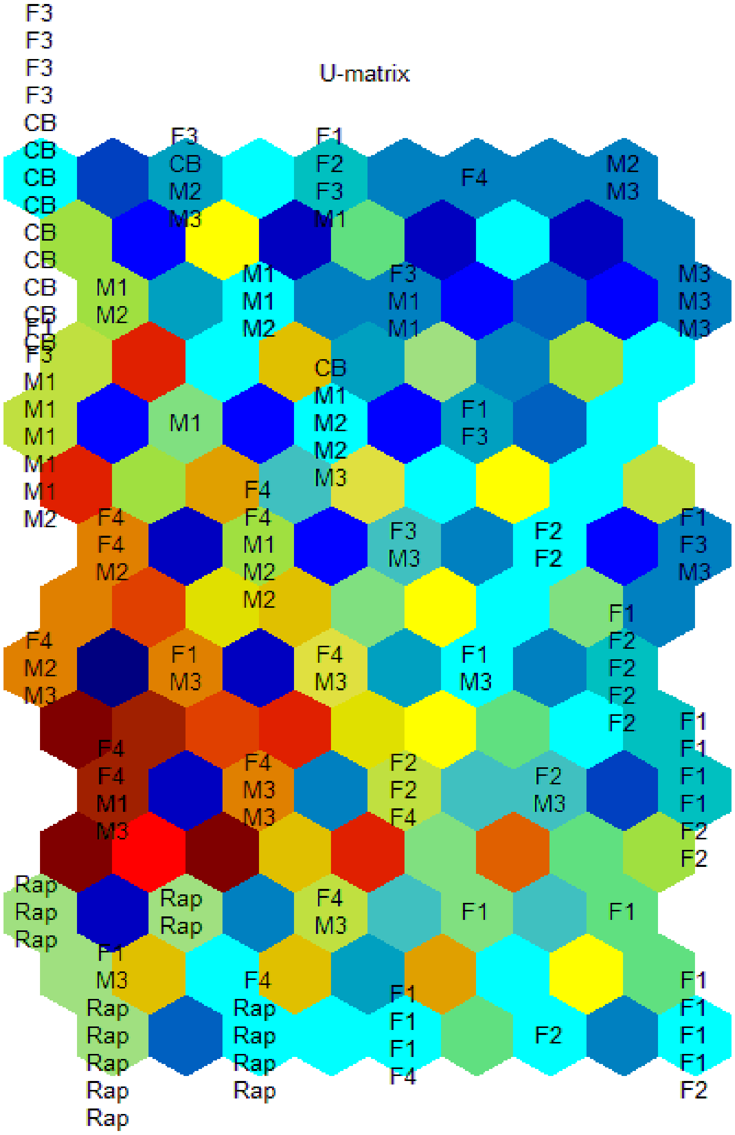


Figure 42 8×5 U-matrix of the variance unwrapped chromagram of the vocal songs

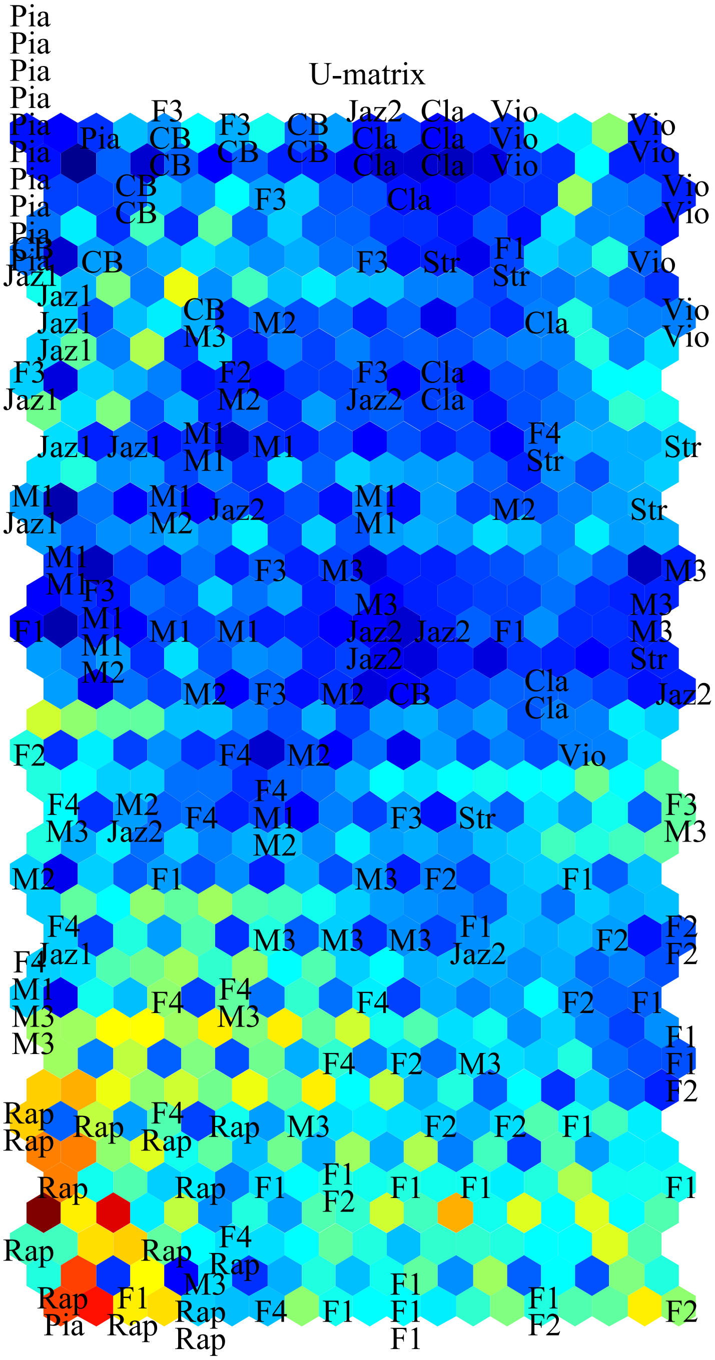


Figure 43 20×10 U-matrix of the variance unwrapped chromagram of all songs

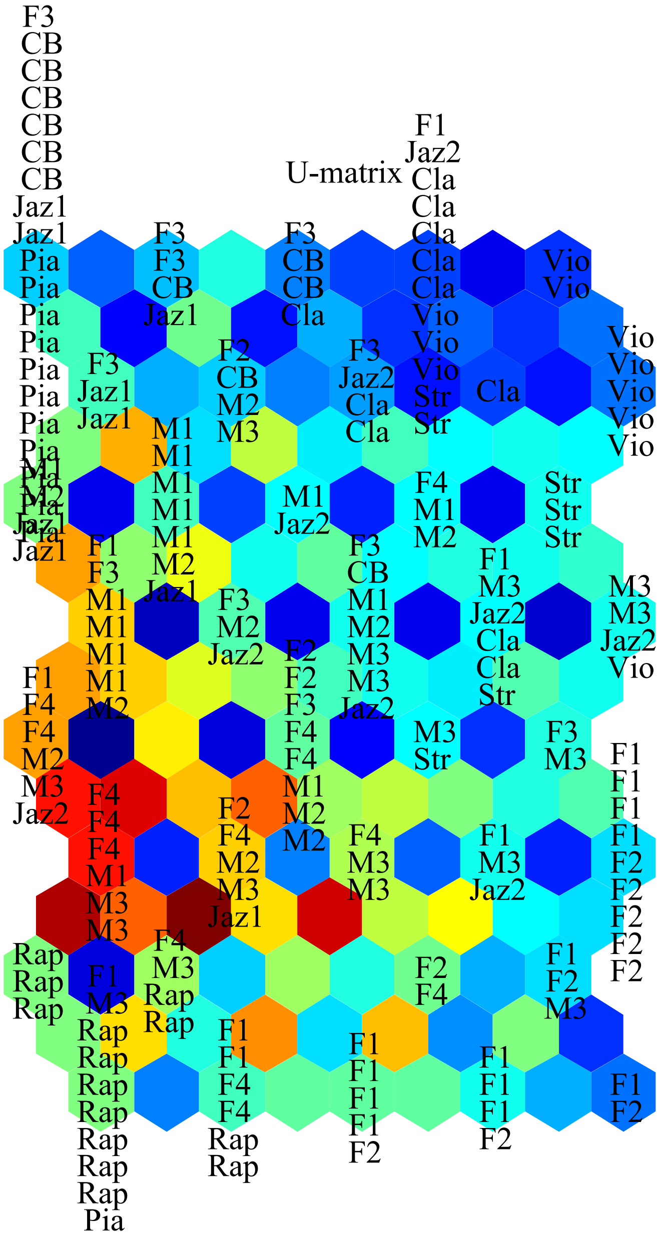


Figure 44 8×5 U-matrix of the variance unwrapped chromagram of all the vocal songs

3.3.4.3.4 Fluctuation pattern

Although fluctuation patterns are used to evaluate rhythm, different genres might have different rhythmic properties. Hence, the clustering of songs according to the fluctuation pattern was also examined, as suggested in [4]. Figure 45 and Figure 46 show the U-matrix of the mean and variance of the fluctuation pattern, and suggest that it can be used to discriminate between instrumental and vocal music. As compared with MFCC, the fluctuation pattern wraps the frequency domain into 40 critical bands in accordance with human perception. However, the KL transform of the final step of obtaining the MFCC preserves the most information about the frequency. As compared with the fluctuation pattern, the unwrapped chromagram has 70 frequency bands, which is more than the fluctuation pattern has. Accordingly, MFCC and the unwrapped chromagram retain more frequency information than the fluctuation pattern. Hence, this may be the reason that the discriminability of the songs sung by females and males when the fluctuation pattern was adopted is not as good as when MFCC or the unwrapped chromagram was adopted.

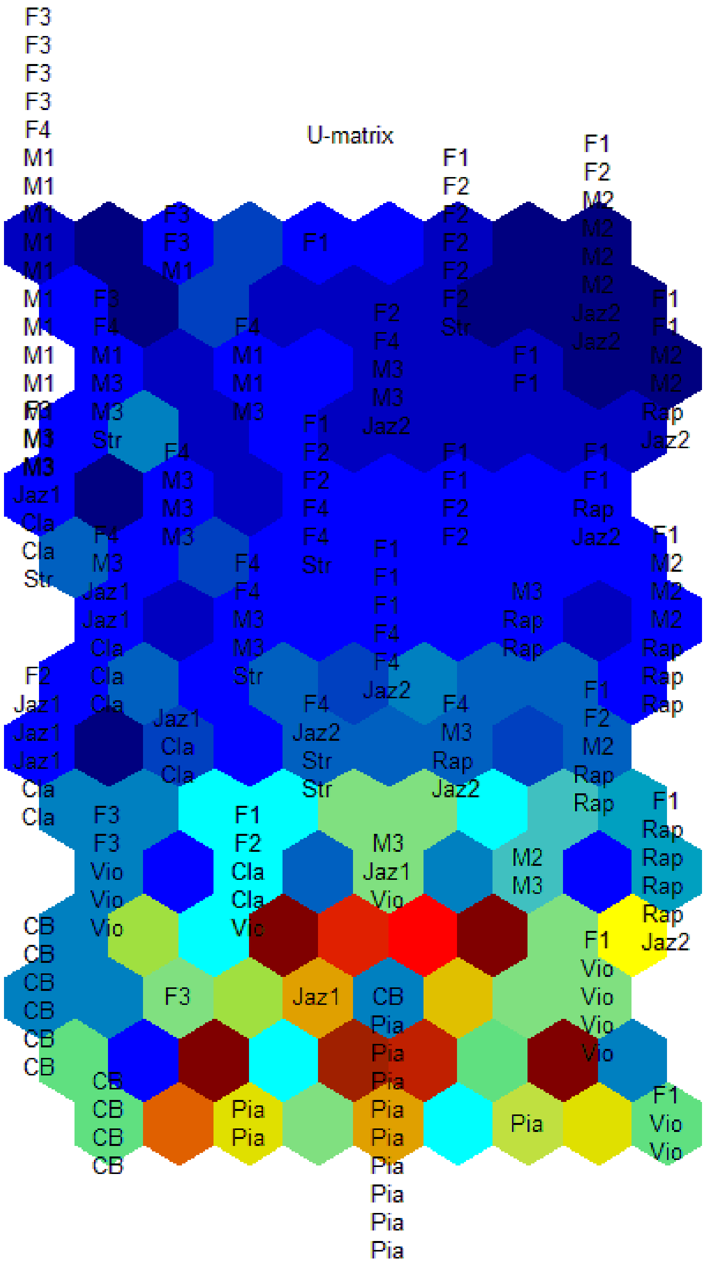


Figure 45 8×5 U-matrix of the mean fluctuation pattern

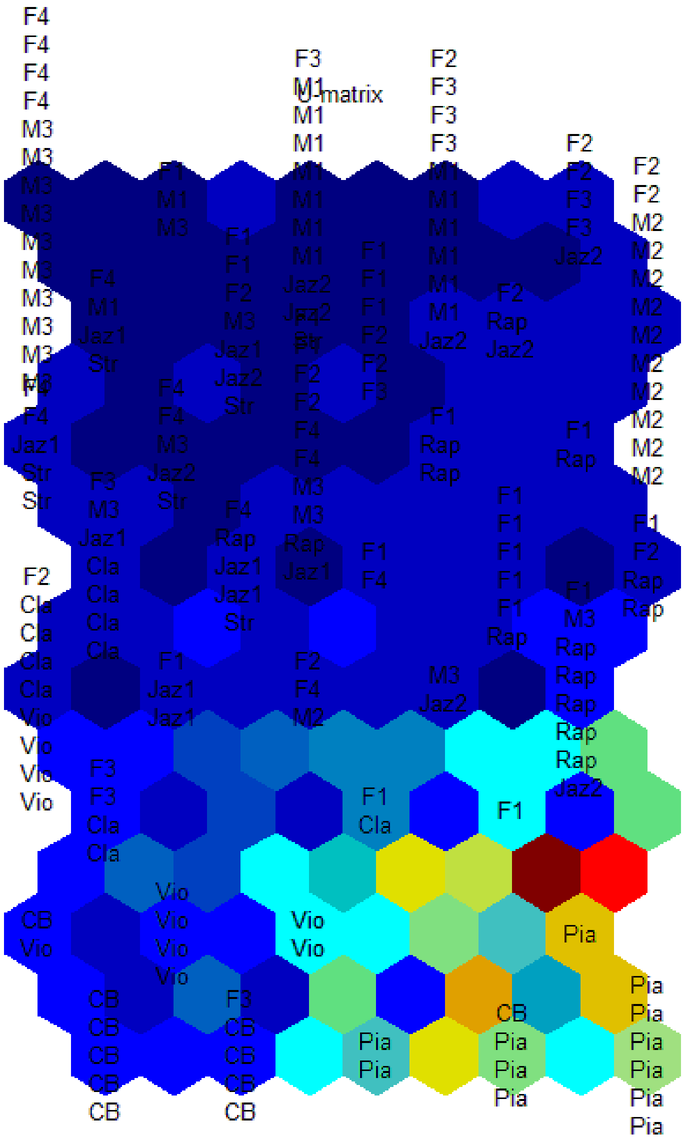


Figure 46 8×5 U-matrix of the variance fluctuation pattern

3.3.4.4 The combination of MFCC and ZCR

The discriminability of the features is summarized in Table 4. In addition, the previous discussion shows that variance of MFCC classifies the albums well, except for those containing piano and violin music, and the mean ZCR has a better ability to identify the instrument. Hence, the combination of the two features was evaluated for classifying the pieces of music using SOM. The results are shown in Figure 47 and Figure 48, which are 8×5 and 20×10 U-matrices, respectively. In Figure 47, it can be seen that the combination seems to separate these genres well, except for the songs of the piano and the choirboys. However, when the size of the U-matrix is increased to 20×10, the songs played by the piano and sung by choirboys are actually separated. Hence, we can conclude that a feature set in which MFCC and ZCR are combined outperforms a single feature.

Table 4 The discriminability of the features is summarized

.Feature	Discriminability
ZCR	Instruments
Variance of MFCCs	Fails to classify the piano and violin
Unwrapped chromagram: instrumental music	Classifies instrumental music well, including jazz
Unwrapped chromagram: vocal songs	Mean separates female and male songs better, but variance separates rap better
Unwrapped chromagram: all albums	Not good

Fluctuation pattern	Does not separate songs sung by females and males as well as MFCC
MFCC+ZCR	Best feature set for discriminate between the albums

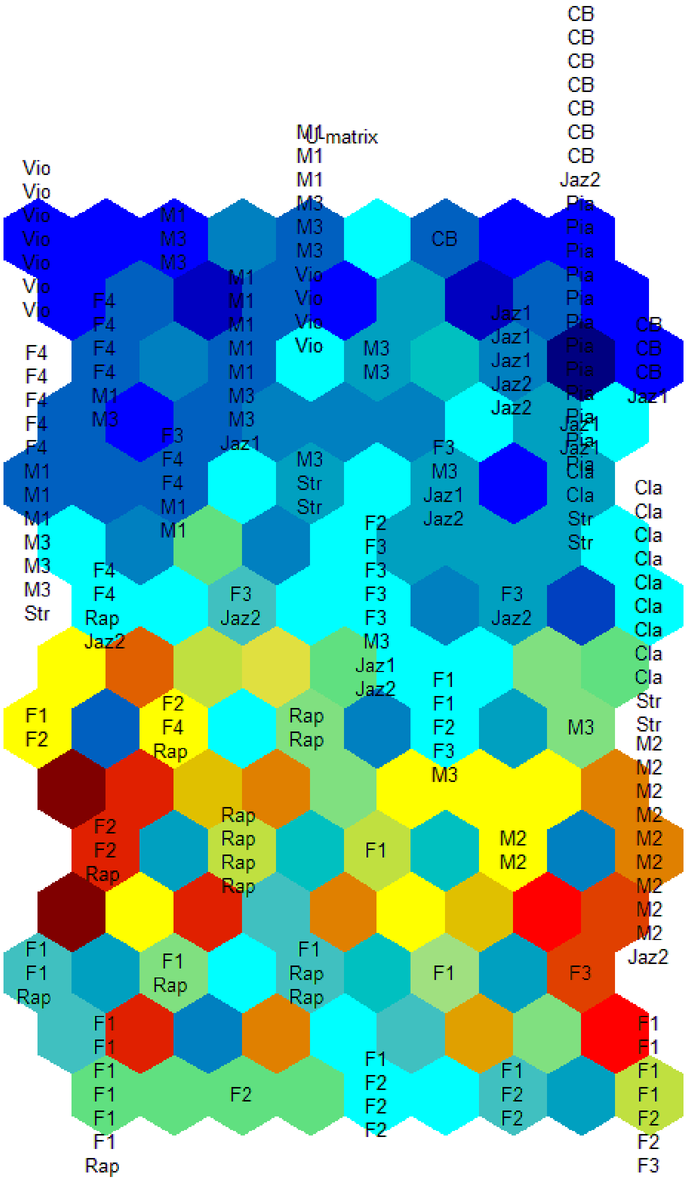


Figure 47 8×5 U-matrix of MFCC variance and mean ZCR

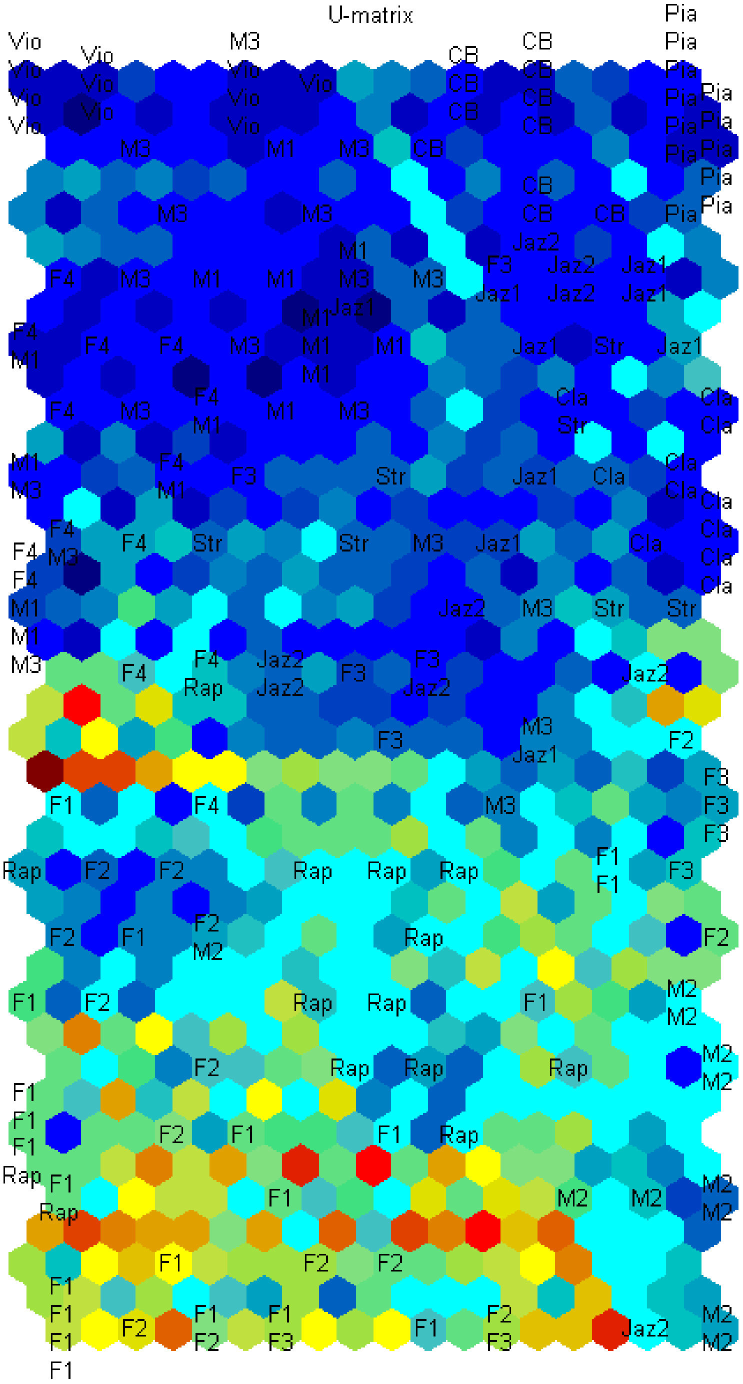


Figure 48 20×10 U-matrix of MFCC variance +mean ZCR

4 Applications

4.1 Structure of a Song

A song might contain several parts, such as the introduction, bridge, or the chorus. A pop song normally repeats the bridge or the chorus parts many times. The goal of the structure analysis here is to extract a good recurring section to constitute the thumbnail of a song, i.e., to perform music summarization. Figure 49 (a), (b), and (c) are the chromagrams of the repeating sections of Annie Lennox's "Don't Let It Bring You Down" corresponding to the sections 0-57 s, 57-114 s, and 114 s-187 s. In other words, the sections 0 s-187 s can be broken into three similar subsections. However, a proper beginning and ending point are normally not easy to find.

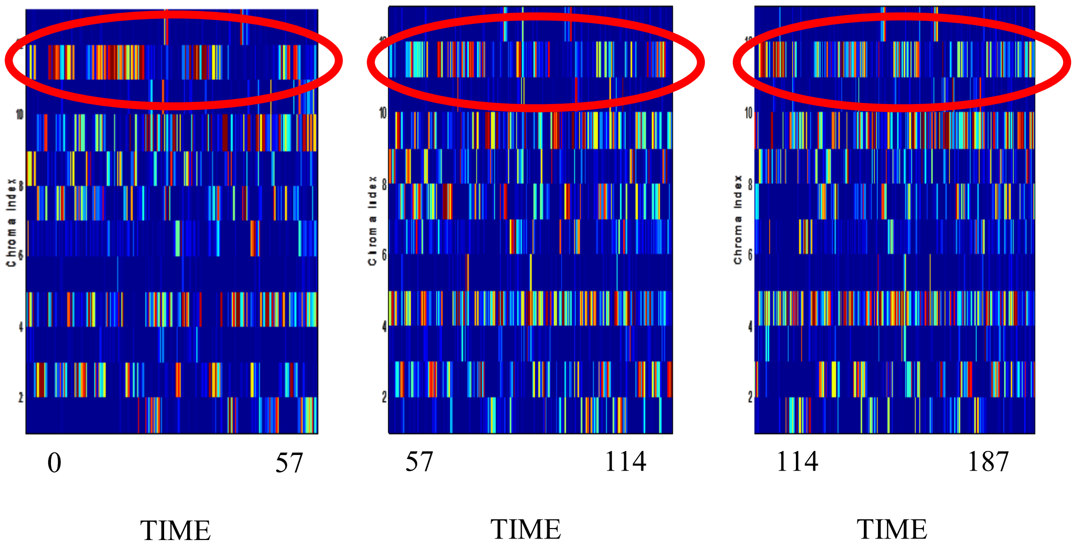


Figure 49 The chromagrams of the repeating sections of Annie Lennox's "Don't Let It Bring You Down." Please pay attention to trends that dominate chromas, which are the notes with the maximum strength.

Inspired by the sequence alignment of biotechnology, we think we can use any alignment algorithm to find the repeating parts of a song. However, since the chromagram is composed of 12-D chroma vectors, we need to convert the chromagram into a sequence of symbols or states. Each state can represent similar chroma vectors. In addition, we hope that by transferring similar chroma vectors into states, instead of comparing the chroma vectors directly, our method will be more robust against the variation of similar melodies. Finally, melodies are composed of a series of notes or chords. The note or chord that follows another is not arbitrary. Hence, this property makes HMM a good selection.

4.1.1 Hidden Markov Model Estimation

An HMM [44] consists of transition probabilities, emission probabilities, invisible hidden states, and observations. A state transition obeys the Markov property: given the present, the future does not depend on the past. Each state generates observations according to its emission probability. Given a set of observations, the parameters of an HMM, such as the state transition probabilities and the emission probabilities, can be derived by the Baum-Welch algorithm, which is a special case of the expectation-maximization algorithm. Conversely, given the HMM and the observations, the most likely state sequences can be obtained using the Viterbi algorithm.

In our case, depending on the applications, a 5-12 state HMM is used. In our experience, for most applications, five states are sufficient to represent a song. However, if precise cutting points are needed, the number of states can be increased up to 12. The emission distribution follows a single independent Gaussian distribution. In addition, instead of using a supervised learning method, the initial conditions are given by the statistics results of the k-means clustering of the chroma vectors with *k* equal to the number of states of an HMM, without any prior musical knowledge or training data.

4.1.2 State-Splitting State Hidden Markov Model (SSS HMM)

After the fixed-state step of the HMM described in section 4.1.1, we found that a five-state HMM state in our task can be further split into several states. For example, Figure 50 shows the chroma vectors when the state index is at state 1 in the sequence of the five-state HMM of The Beatles' "Day Tripper," which can be split into four sub-states, as shown in Figure 51.

The procedure of the SSS HMM is summarized in Figure 52. After the HMM and the most likely state sequence of a song are obtained by the procedure described in section 4.1.1, the chroma vectors corresponding to each state are extracted, and then, the k-means clustering analysis is run for each state, with the number of clusters, *k*, defined as the following:

1. The mean of a state's chroma vectors is calculated to produce a mean vector.
2. *k* is set to the number of local maxima of the mean vector that are above the average of the vector's elements.
3. Run the k-means analysis with *k* defined by step 2. If it fails to converge, then reduce *k* by 1.
4. Check the number of elements in each cluster. If the number of elements is less than a predefined threshold, reduce *k* by 1 and return to step 3. This step ensures there are enough frames in each cluster so that the following retrained HMM will be more stable.

After running the k-means analysis for each HMM state derived as in section 4.1.1, a new HMM of the song is trained with the number of states equal to the number of clusters defined by steps 1–4 above. Furthermore, the initial conditions of the new HMM, such as the mean and variance, are set according to the statistics results of the cluster. Finally, a new state sequence of the song is decoded using the Viterbi algorithm.

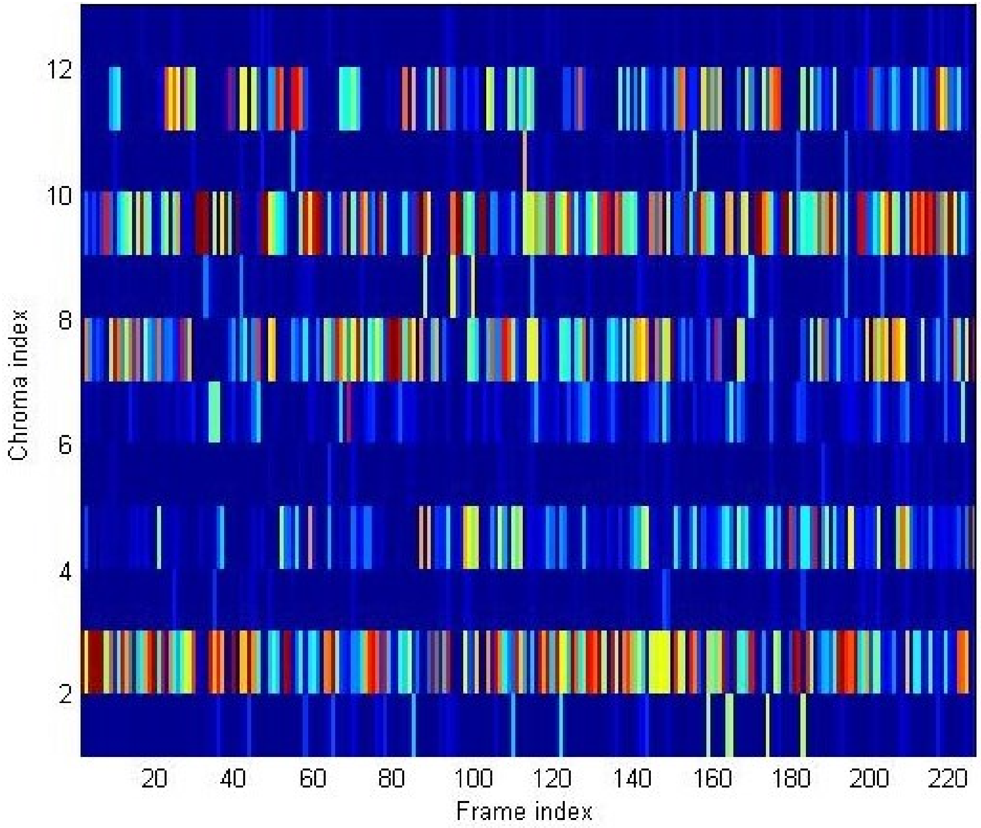


Figure 50 The chroma vectors with state index 1 in the sequence of the 5-state HMM of The Beatles' "Day Tripper"

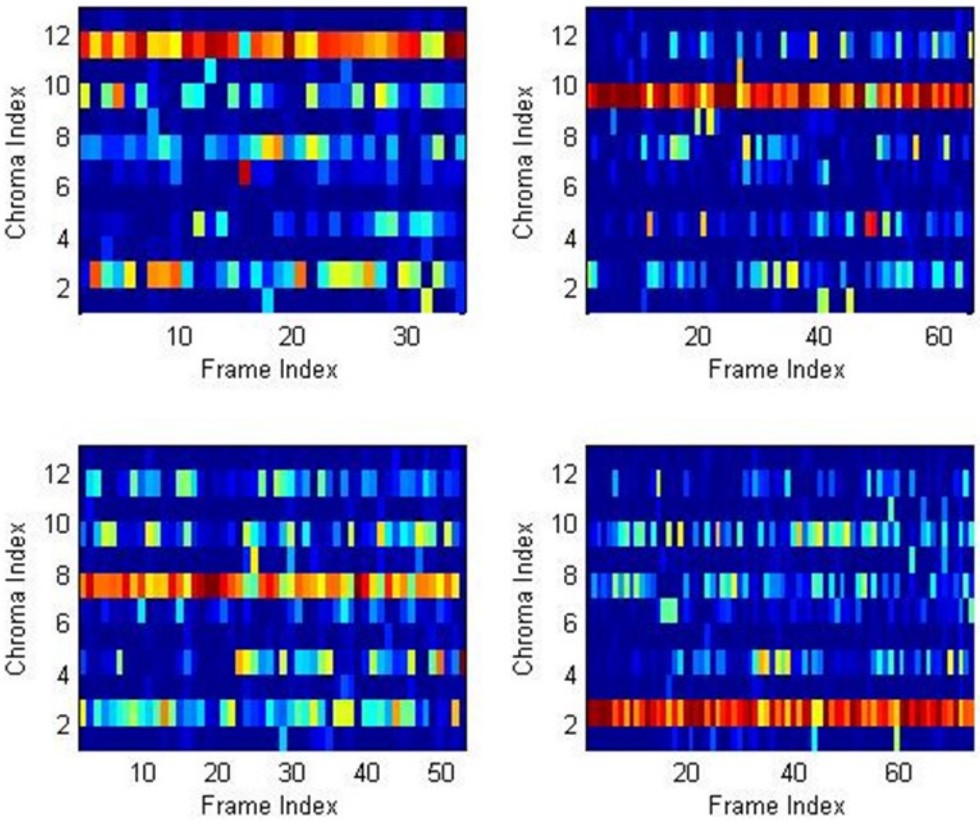


Figure 51 The chroma vectors shown in Figure 50 can be split into the four sub-states by k-means

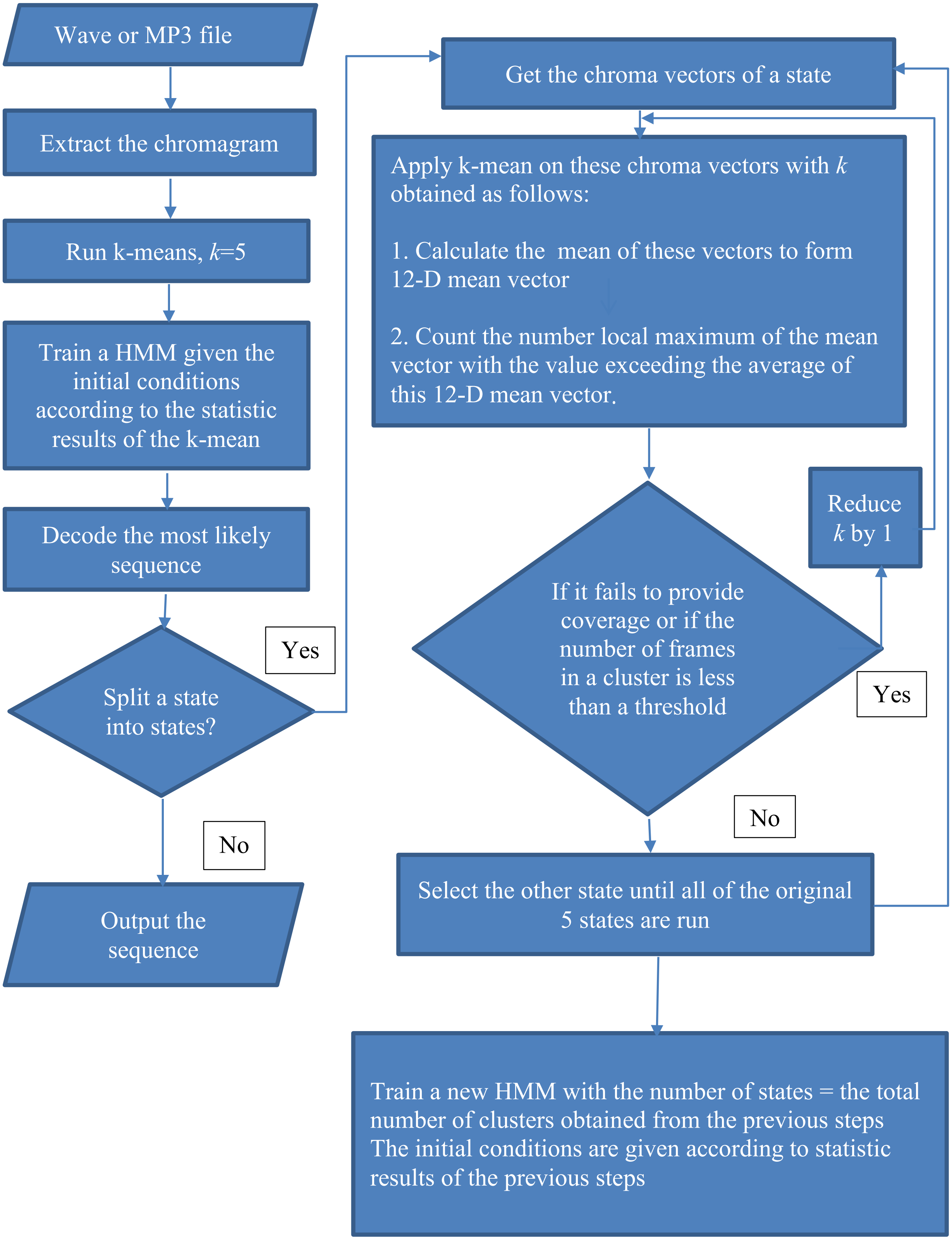

Wave or MP3 file
Extract the chromagram
Run k-means, *k*=5
Train a HMM given the initial conditions according to the statistic results of the k-mean
Decode the most likely sequence
Split a state into states?
Yes
No
Output the sequence
Get the chroma vectors of a state
Apply k-mean on these chroma vectors with *k* obtained as follows:
1. Calculate the mean of these vectors to form 12-D mean vector
2. Count the number local maximum of the mean vector with the value exceeding the average of this 12-D mean vector.
If it fails to provide coverage or if the number of frames in a cluster is less than a threshold
Yes
Reduce *k* by 1
No
Select the other state until all of the original 5 states are run
Train a new HMM with the number of states = the total number of clusters obtained from the previous steps The initial conditions are given according to statistic results of the previous steps

Figure 52 The method of State-Split-State HMM

4.1.3 State Sequence Alignment

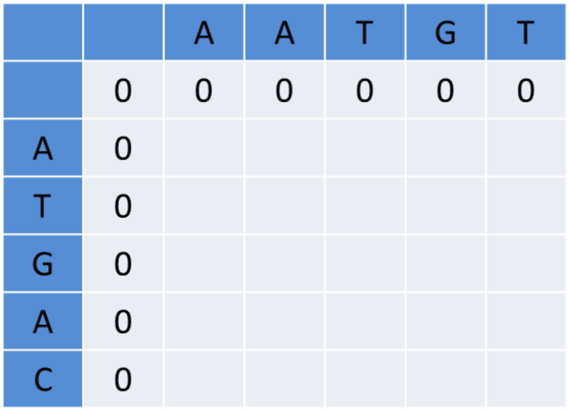
After transferring the chromagram into the states or symbols of a sequence, the next goal is to extract similar sub-sequences within a sequence or among sequences. Hence, the Smith-Waterman algorithm [45], a dynamic programming algorithm, is used to determine the optimal local alignment of two sequences with respect to the scoring matrix (*W*) and gap penalty (*G*). A similarity matrix (*S*) for the two sequences is built as

$$\mathrm{S}(i,j) = \max\begin{cases} S(i-1,j-1) + W\left(State_{seq1}, State_{seq2}\right) \\ S(i-1,j) - \mathrm{G} \\ S(i,j-1) - \mathrm{G} \\ 0 \end{cases},$$

where $W\left(State_{seq1}, State_{seq2}\right)$ is the score between the states of sequence 1 and sequence 2, and *G* is the gap penalty

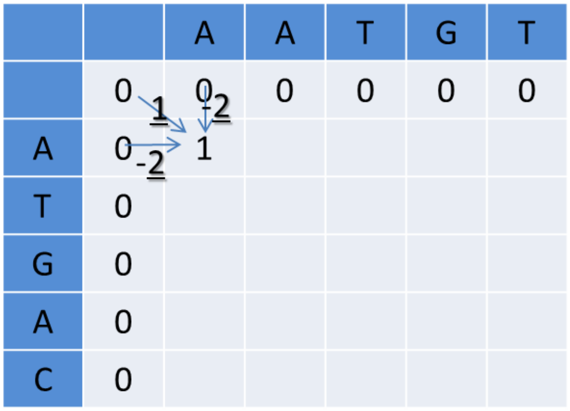
When S has been constructed, its negative components are set to 0. The optimal local-aligned section is obtained by backtracking from the largest component in *S* until the first 0 is encountered.

The Smith-Waterman algorithm is often used to align amino acid, protein, or nucleotide sequences in biotechnology. Figure 53 shows an example of finding the optimal local alignment of the two amino acid sequences, “A A T G T” and “A T G A C.” Assume the score of a matched label is 1 and a miss-matched label is -1, and the gap penalty is -2. First, the elements of the first row and column are set to 0 in Figure 53(a). Then, the first labels of the two sequences are “A.” Hence, $S(1,1) = \max\begin{cases} S(0,0) + 1 \\ S(0,1) - 2 \\ S(1,0) - 2 \\ 0 \end{cases} = 1$ as shown in Figure 53(b). Moreover, the second label of the sequence “A A T G T” which is “A” is matched to the first label of the sequence “A T G A C”; therefore, $S(1,2) = \max\begin{cases} S(0,1) + 1 = 1 \\ S(0,2) - 2 = -1 \\ S(1,1) - 2 = -2 \\ \qquad\quad 0 \end{cases} = 1$ in Figure 53(c). Furthermore, the third label of the sequence “A A T G T,” which is “T,” is miss-matched to the first label of the sequence “A T G A C,” which is “A.” Hence, $S(1,3) = \max\begin{cases} S(0,2) - 1 = -1 \\ S(0,3) - 2 = -2 \\ S(1,2) - 2 = -1 \\ \qquad\quad 0 \end{cases} = 0$ in Figure 53(d). The matrix (S) can be filled by following the procedure, and the highest value in the matrix is 3. Hence, the optimal local alignment can be obtained by backtracking from the highest value cell, until the first cell with zero value is encountered in Figure 53(h). The optimal local alignment of the two sequences is “A T G.”



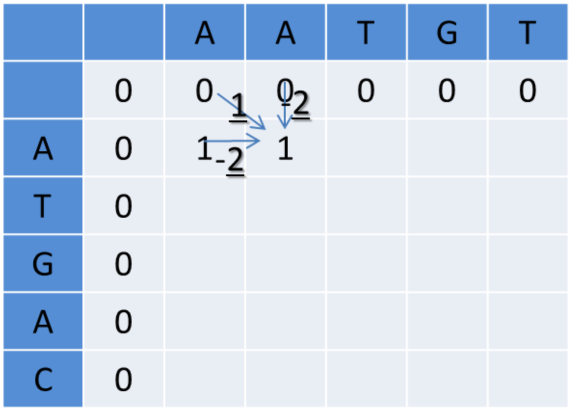
		A	A	T	G	T
	0	0	0	0	0	0
A	0					
T	0					
G	0					
A	0					
C	0					

(a)



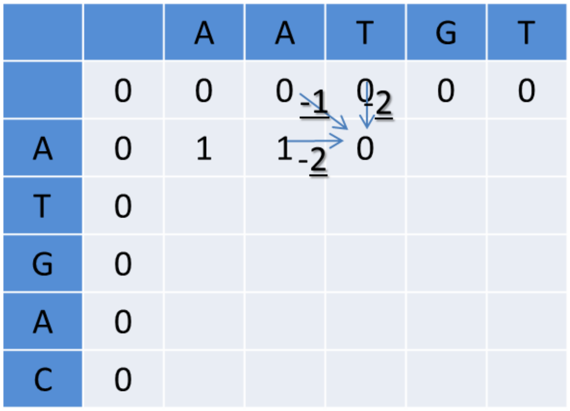
		A	A	T	G	T
	0	0	0	0	0	0
A	0	1				
T	0					
G	0					
A	0					
C	0					

(b)



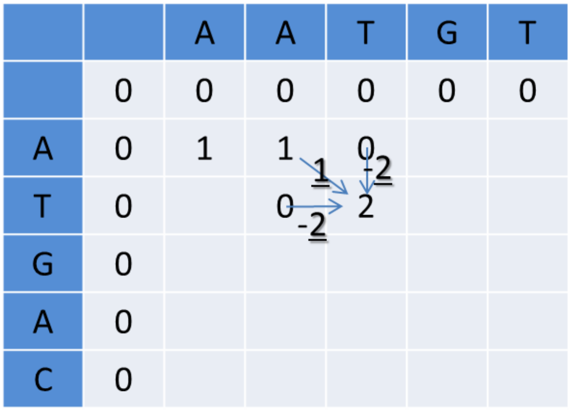
		A	A	T	G	T
	0	0	0	0	0	0
A	0	1	1			
T	0					
G	0					
A	0					
C	0					

(c)



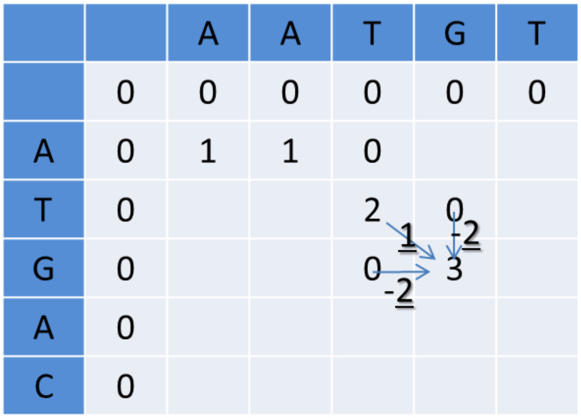
		A	A	T	G	T
	0	0	0	0	0	0
A	0	1	1	0		
T	0					
G	0					
A	0					
C	0					

(d)



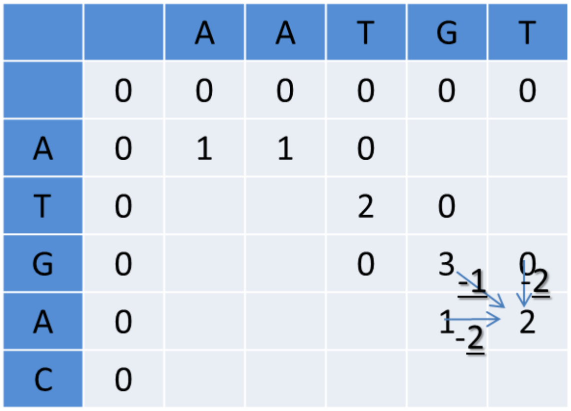
		A	A	T	G	T
	0	0	0	0	0	0
A	0	1	1	0		
T	0		0	2		
G	0					
A	0					
C	0					

(e)



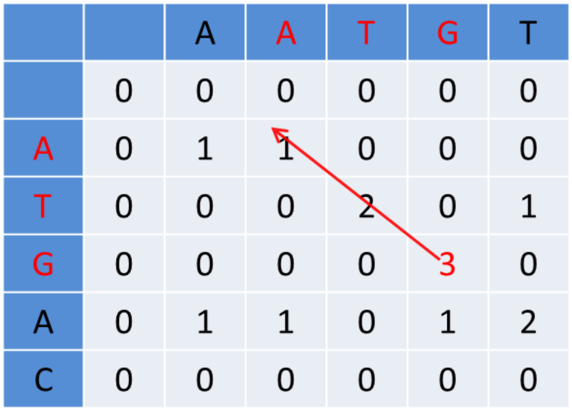
		A	A	T	G	T
	0	0	0	0	0	0
A	0	1	1	0		
T	0			2	0	
G	0			0	3	
A	0					
C	0					

(f)



		A	A	T	G	T
	0	0	0	0	0	0
A	0	1	1	0		
T	0			2	0	
G	0			0	3	0
A	0				1	2
C	0					

(g)



		A	A	T	G	T
	0	0	0	0	0	0
A	0	1	1	0	0	0
T	0	0	0	2	0	1
G	0	0	0	0	3	0
A	0	1	1	0	1	2
C	0	0	0	0	0	0

(h)

Figure 53 An example of finding the optimal local alignment of the two amino acid sequences, "A A T G T" and "A T G A C"

4.1.4 Score for the Local Alignment

For each state of a song, the empirical mean over frames of the chroma vectors in the state is used to represent the state. The pairwise Euclidian distance (P) between the states' mean vectors are then calculated. In order to increase the score (W) of the aligned states and decrease the score of the misaligned states, the scoring matrix (W) is defined as

$$W = diag(1) - \frac{P}{max(P)}$$

Since the maximum score will be 1, the gap penalties from 0-1.3 are examined.

The best performance is normally achieved when the gap penalty is set to 1.

4.1.5 Method of Music Summarization

The goal of music summarization is to obtain the most distinctive section by which a listener can easily identify the song. The excerpt is normally a recurring section. To extract a repeating section, as shown in Figure 54, the song's sequence estimated from its own SSS HMM or 5-state HMM is used, and then, the local self-alignment is applied on its own state sequence. Of course, the perfect alignment from beginning to end always yields the highest score, whereas, due to local alignment of the Smith-Waterman algorithm, a section corresponding to a sub-optimal score, where the aligned sequences' length is less than half the length of the whole song, generally constitutes replication segments that can be marked as the summary of the song.

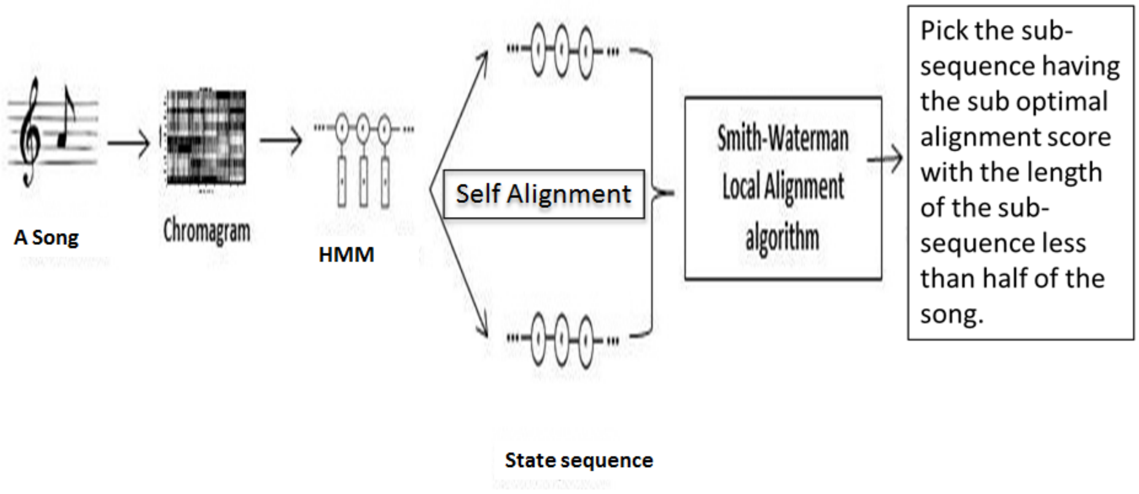

A Song
Chromagram
HMM
Self Alignment
State sequence
Smith-Waterman Local Alignment algorithm
Pick the sub-sequence having the sub optimal alignment score with the length of the sub-sequence less than half of the song.

Figure 54 Music summarization method. First, the chromagram of a song is extracted. Then, the HMM model is trained using the initial condition according to the statistics results of k-means, or SSS HMM is trained as described in 3.1.2. The most likely state sequence is decoded. Finally, the sequence is aligned to itself. The maximums score will be the whole alignment of the sequence. However, the sub-sequence corresponding to the suboptimal alignment score with the length of the sub-sequence less than half of the song is selected.

4.1.6 Cover Song Identification

A cover version is a new recording of an old song. Owing to the similar melodies of the cover version and the original song, our method can be applied to the task of cover song identification with some modifications. For the purpose of comparing our findings with published results, the performance of the system was first evaluated using the 80 pairs of songs (160 different songs) specified in [46]. For each pair, one song was sampled as the query song and the other reserved as the reference candidate. There were 80 query songs in total, and for every query song there were 80 reference candidates in the database. One of the 80 reference candidates was the cover version of the query song. Our goal was to determine the correct cover version from the set of reference candidates for a query song.

As shown in Figure 55, in our method, an HMM is first trained for each reference song. Each HMM contains five hidden states. The chromagram of a song is used as the observed output of the HMM. The probability is assumed to be a single Gaussian distribution with a full covariance matrix. The initial means and the covariance of the probability for each state are estimated by the empirical means and covariance of the k-means clustering with $k = 5$. In order to stabilize the HMM, if the length of a chromagram does not exceed a threshold (1500 frames in this case), the chromagram will be replicated and padded at the end of the original chromagram.

The chromagram of a query song is extracted and then cyclically shifted in the chromatic domain to capture the 12 key shifts. The set of the reference songs' HMMs are then applied to the shifted chromagrams of a query song in order to estimate the state sequences of the query song corresponding to the set of HMMs. This gives 12 × 80 state sequences to be estimated for every query song. Finally, the Smith-Waterman sequence alignment algorithm is applied to all 12 × 80 sequences to identify the correct reference for the query song. In consequence, a 12 × 80 alignment score vector is obtained for each query song. The reference song with the largest alignment score is selected as the cover version of the query song.

In general, the alignment of longer sequences tends to create a larger alignment score. Normally, the length of a pop song is about 4–5 min, and a portion of a song might repeat many times. Hence, the length of a query song's sequence in our case is limited to 700 frames, that is, about 175 s, to prevent the alignment algorithm from developing a score bias for longer sequences.

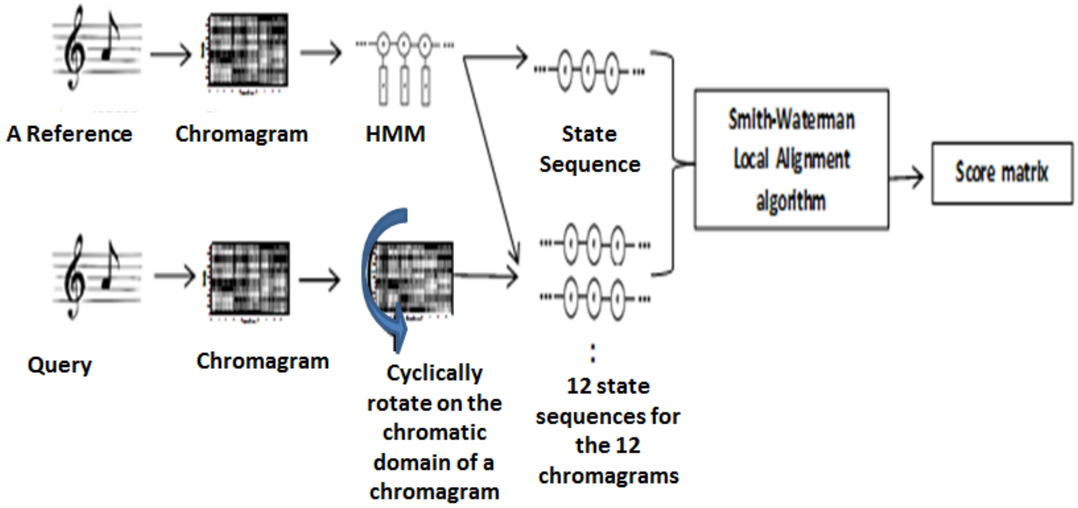

A Reference
Chromagram
HMM
State Sequence
Smith-Waterman Local Alignment algorithm
Score matrix
Query
Chromagram
Cyclically rotate on the chromatic domain of a chromagram
12 state sequences for the 12 chromagrams

Figure 55 Method of cover song identification

4.2 Extraction of Similar Melodies from Two Different Songs

In cover song identification, the two members of a pair are expected to have similar melodies. However, while the melodies of two different songs may be diverse, the two songs may have a short segment that is similar. The goal of this part of our study was to extract these segments. Similarly to cover song identification in section 4.1.5, an HMM or a SSS HMM was trained and the sequence was estimated for one song (called song 1); the other song's chromagram was rotated, and then the 12 sequences of the rotated chromagram, given song 1's HMM, were obtained. Finally, the local alignment method was applied to align the sequences of the two songs. The start and end points of the alignment were recoded, and mp3 or wav files of the segments were extracted so that by listening to them we could evaluate the performance of our method. We selected a song on YouTube and some songs on the Internet specified in [35] as our test songs.

4.3 Results

4.3.1 Data-Driven Chord Sequence Representation

Bello [28] claims that the relative semitone of a chord should be (*n*, *n*+4, *n*+7) for the major triad and (*n*, *n*+3, *n*+7) for the minor triad, where *n* is the root note. Since our method is unsupervised, the states in different songs represent different properties of chroma vectors, and hence, a chroma vector corresponding to a state in different songs is not comparable. Therefore, for each song ($j \in 1,2 \dots 160$), a mean chroma vector ($\tilde{\mu}_{i,j}$) is calculated where i is the state index ($i \in 1,2 \dots 5$)

$$\tilde{\mu}_{i,j} = \frac{\sum_k v_{k,i,j}}{n}$$

where $v_{k,i,j}$ is the chroma vector of frame k at state i of song *j* and *n* is the number of frames at the i^{th} state of the j^{th} song

Consequently, there are five mean chroma vectors representing five states for each song. Since there are 160 songs in our database, 800 mean chroma vectors ($\tilde{\mu}_{i,j}$) represent the states of the 160 songs.

We further grouped these chroma vectors ($\tilde{\mu}_{i,j}$) into 12 groups ($\tilde{\tilde{\mu}}_{i,j,c}$) according to the root note (c) which has the maximum strength among the 12 chroma of a chroma vector, and then calculated the average (μ_c) for each group, that is,

$$\mu_c = mean_{i,j}\left(\tilde{\tilde{\mu}}_{i,j,c}\right) \text{ where } \{c \in 1,2 \dots 12\}$$

Figure 56 and Figure 57 show the μ_c of the HMM and SSS HMM method, respectively. In the figures, it can be seen that, although *n*+3 and *n*+4 are not very obvious, the chroma indices *n* and *n*+7 tend to have large values, which is coincident with the musical knowledge. These tendencies are completely data-driven, without adopting any prior knowledge of music before training the HMMs. Thus, this property shows that an HMM state represents a chord instead of a harmonic.

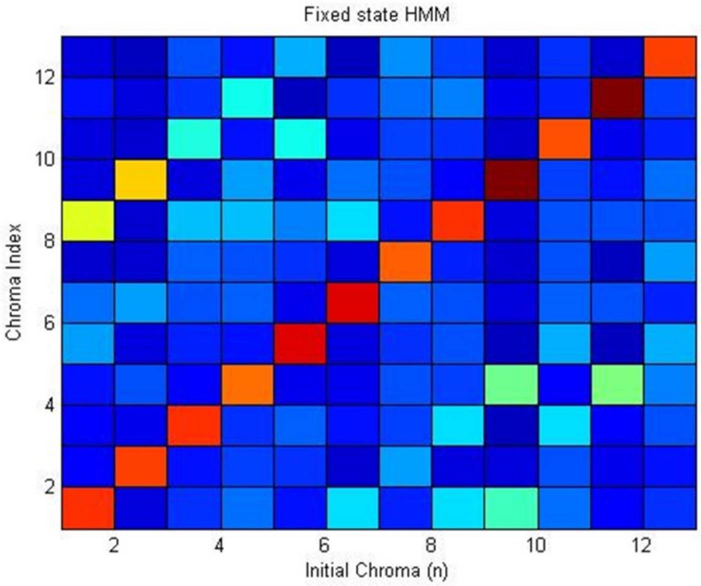


Figure 56 μ_c of the fixed-state HMM. The chroma indices *n* and *n*+7 tend to have a larger strength, which is coincident with the musical knowledge of a chord.

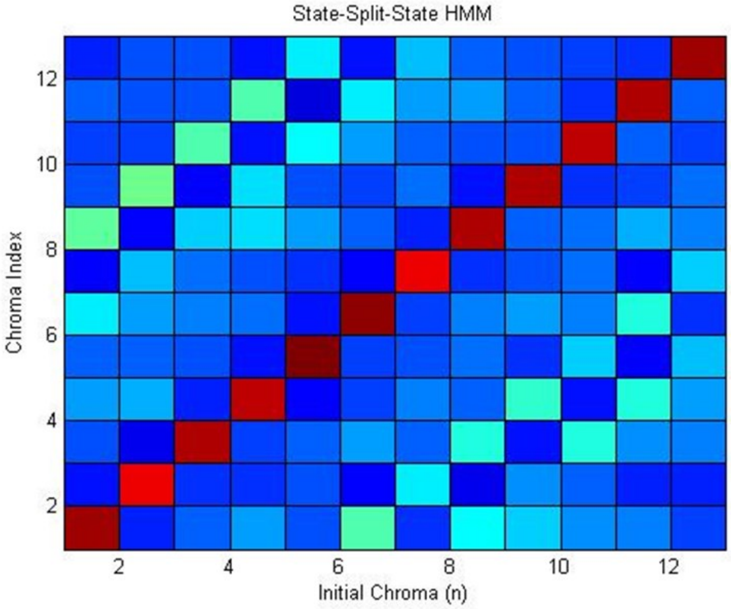


Figure 57 μ_c of the SSS HMM. The chroma indices *n* and *n*+7 tend to have a larger strength, which is coincident with the musical knowledge of a chord.

In addition to examining the chromagram, we asked Szu-Ying Wang to listen carefully to the tracks of a state to verify which common musical property is captured by a state. Szu-Ying Wang is an assistant professor at the National Hsinchu University of Education who was awarded a doctorate in Musical Arts by the University of Maryland. We found that the representation of a musical property for a state differs from state to state and from song to song. For example, state 1 of the Best of the Boomtown Rats' "I don't like Mondays" captures the glissando with E-e over three octaves, the melody with E-E-D-E, D-C by the piano and singer, and syncopation. The melody of state 2 modulates to a half step lower key with many G, F, D, and C notes. The sections in state 3 are flat, containing many lyrics with E-F-E-F. Most of the melody in state 4 is G-C-C-F and the register in the state is wider than in the other states. State 5 captures clapping C-B-A-G in a descending scale (bass line), the lyrics "Tell Me Why" with the melody E-D-C, and "I Don't Like Mondays" with C-C-D-C-G-C.

The other example, the Beatles' "Day Tripper," shows that state 1 covers the melody E-G-A-B-E-D played by the electronic guitar. We feel that the tracks in state 1 repeat a motif many times. The melody in state 2 has a wider register than in state 1. State 3 captures C- E_flat- G. State 5 captures the bass line notes of G-B-D-F, and the notes repeat many times. Finally, sometimes the harmony is captured, such as in state 1 of "A Whiter Shade of Pale." In a nutshell, the property captured by a state is still an open problem due to the unsupervised learning of our method.

4.3.2 Music Summary

The aim of automatic music summarization is to extract the most memorable portion of a song. Normally, the section of a pop song by which a listener can identify the song is repeated several times. The sub-optimal local self-alignment method is suitable for this task.

The self-alignment of the whole song always has the highest score. However, the section corresponding to the sub-optimal score, where the length of the aligned sequence is less than half of the whole song, is identified as a summary of the song. For example, Figure 58 shows the chromagram of Annie Lennox's "Don't Let It Bring You Down." The white line in Figure 59 represents the most likely state sequence derived from the HMM. Applying the Smith-Waterman algorithm in the state sequences indicates that section 0–111 s is aligned with section 51–183 s, as shown in Figure 60, which shows the two state sequences of sections 0-111 s and 51-183 s together, without applying the local alignment algorithm. The two sequences do not align well. However, if the local alignment algorithm is used and the gaps are inserted properly, as shown in Figure 61, the appearance of the two state sequences is similar. The representation of the alignment of the two state sequences is converted into the labels of amino acids in Figure 62. Listening to the overlap section in more detail, we found that the section from 51 s to 111 s is a replication of the sections from 0 s to 51 s and 111 s to 183 s. Owing to the gap insertion, the different lengths of these segments are reasonable.

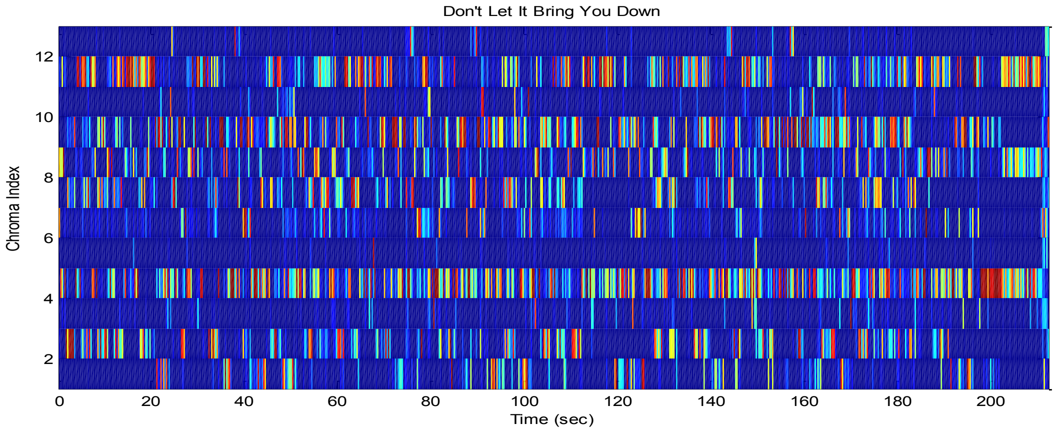


Figure 58 Chromagram of Annie Lennox's "Don't Let It Bring You Down" from 0 s to 200 s

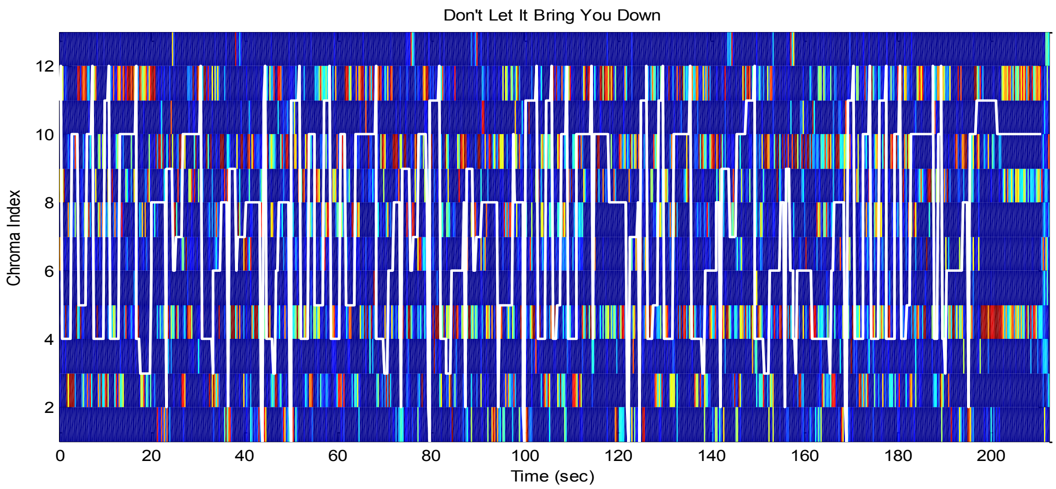


Figure 59 Chromagram of Annie Lennox's "Don't Let It Bring You Down" from 0 s to 200 s. The white lines in the figure represent the most likely state sequence derived from the HMM

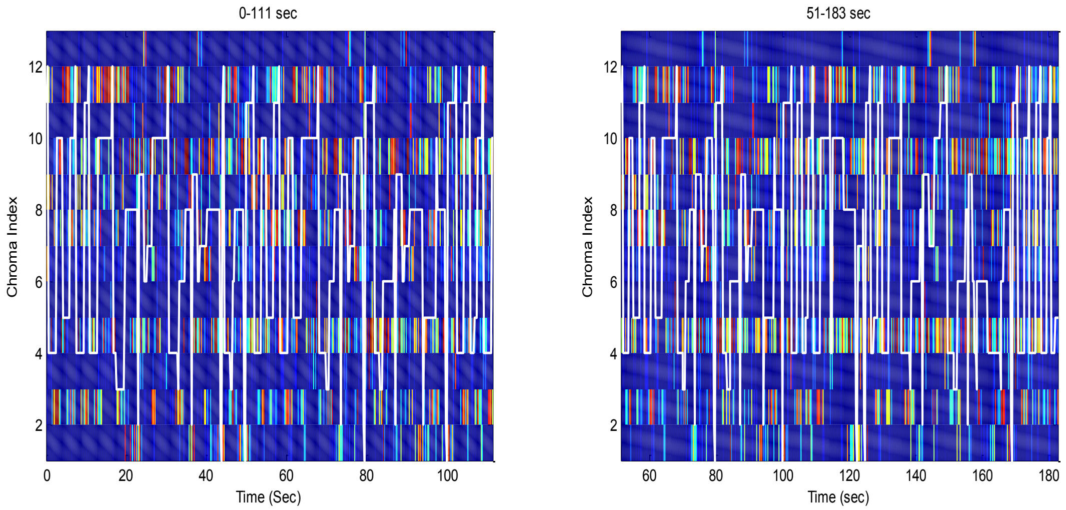


Figure 60 Chromagram of the repeating sections of Annie Lennox's "Don't Let It Bring You Down" extracted using our algorithm. The white line in the figure represents the state sequence of the HMM.

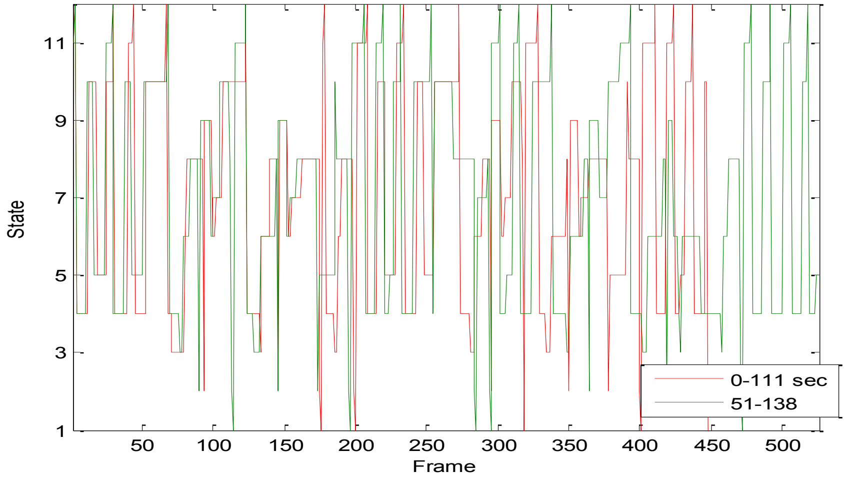


Figure 61 State sequence of 0-111 s and 51-138 s sections without applying the Smith-Waterman algorithm and inserting gaps

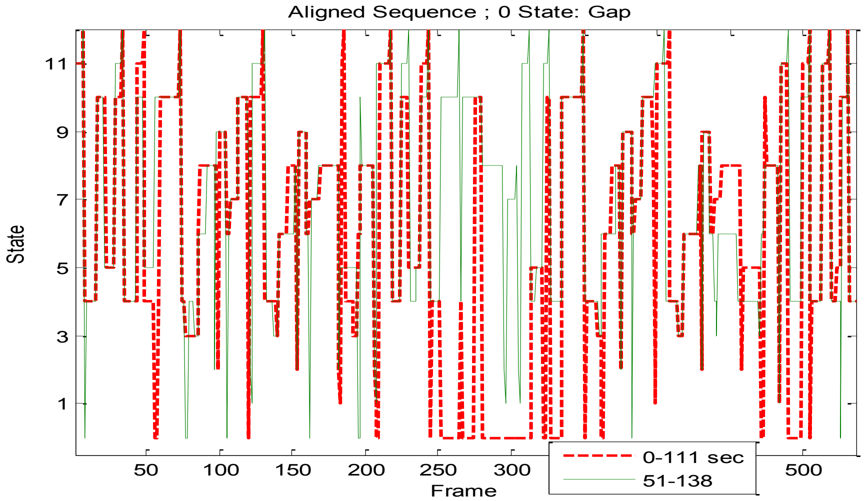
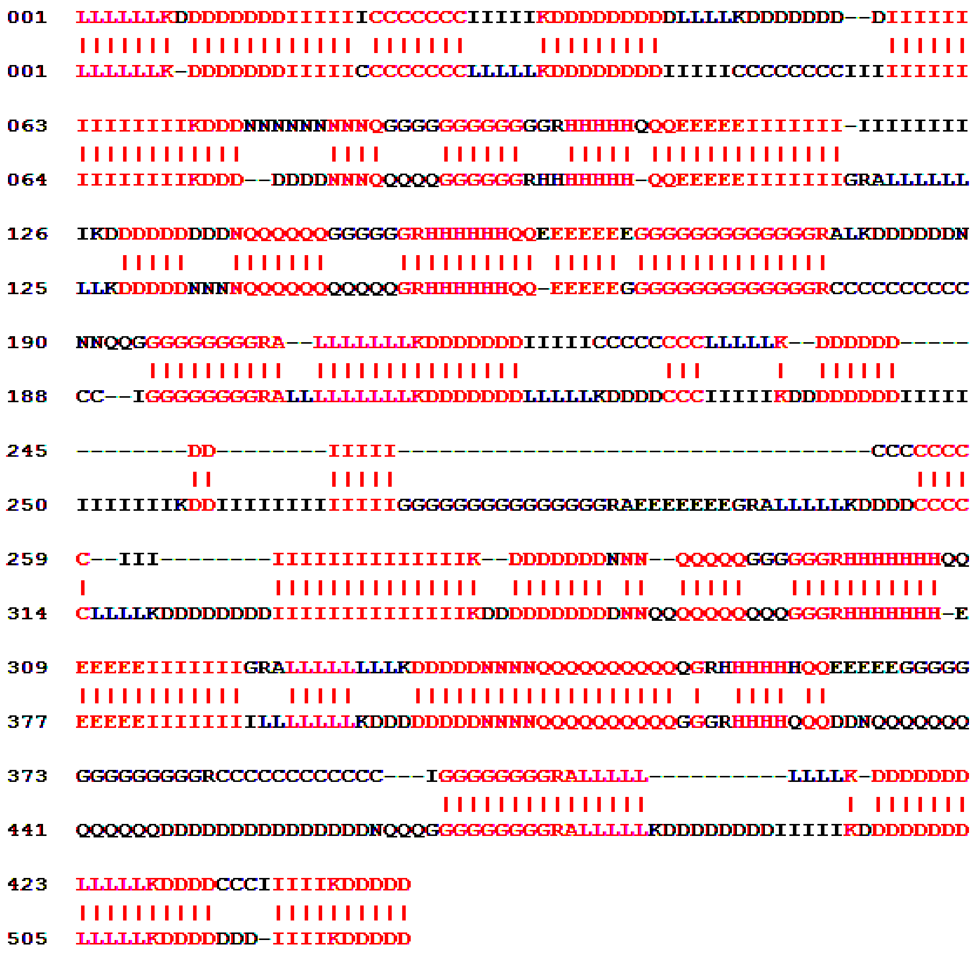


Figure 62 Aligned state sequence of the 0-111 s and 51-138 s sections applying the Smith-Waterman algorithm and inserting gaps



```
001   LLLLLLKDDDDDDDDIIIIIICCCCCCCIIIIIKDDDDDDDDDLLLLKDDDDDDD--DIIIIII
      ||||||| |||||||||||| |||||||     |||||||||                ||||||
001   LLLLLLK-DDDDDDDIIIIICCCCCCCCLLLLLKDDDDDDDDIIIIICCCCCCCIIIIIIIIII

063   IIIIIIIIKDDDNNNNNNNNNQGGGGGGGGGGGGRHHHHHQQQEEEEEIIIIIII-IIIIIIII
      ||||||||||||      ||||    ||||||   ||||| ||||||||||||||
064   IIIIIIIIKDDD--DDDDNNNQQQQQGGGGGGRHHHHHHH-QQEEEEEIIIIIIIGRALLLLLL

126   IKDDDDDDDDDNQQQQQQGGGGGGRHHHHHHQQEEEEEEEGGGGGGGGGGGGGRALKDDDDDDN
         |||||   |||||||     |||||||||| ||||| ||||||||||||||
125   LLKDDDDDNNNNQQQQQQQQQQQGRHHHHHHQQ-EEEEEGGGGGGGGGGGGGGRCCCCCCCCCC

190   NNQQGGGGGGGGGRA--LLLLLLLKDDDDDDDIIIIICCCCCCCCLLLLLK--DDDDDD-----
           ||||||||||  |||||||||||||||          |||     |  ||||||
188   CC--IGGGGGGGGRALLLLLLLLLKDDDDDDDLLLLLKDDDDCCCIIIIIKDDDDDDDDIIIII

245   --------DD--------IIIII----------------------------------CCCCCCC
              ||        |||||                                    ||||
250   IIIIIIIKDDIIIIIIIIIIIIIGGGGGGGGGGGGGGGRAEEEEEEEGRALLLLLKDDDDCCCC

259   C--III--------IIIIIIIIIIIIIIK--DDDDDDDNNN--QQQQQGGGGGGRHHHHHHHQQ
      |             |||||||||||||||  ||||||| ||  |||||   |||||||||||
314   CLLLLKDDDDDDDDIIIIIIIIIIIIIIKDDDDDDDDDDNNQQQQQQQQQQGGGRHHHHHHH-E

309   EEEEEIIIIIIIGRALLLLLLLLKDDDDDNNNNQQQQQQQQQQQGRHHHHHHQQEEEEEGGGGG
      ||||||||||||   |||||    ||||||||||||||||||| |  |||| ||
377   EEEEEIIIIIIIILLLLLLLKDDDDDDDDNNNNQQQQQQQQQQGGGRHHHHQQQDDNQQQQQQQ

373   GGGGGGGGGRCCCCCCCCCCCC---IGGGGGGGGRALLLLL----------LLLLK-DDDDDDD
                                |||||||||||||||              | |||||||
441   QQQQQQDDDDDDDDDDDDDDDNQQQGGGGGGGGGRALLLLLKDDDDDDDDIIIIIKDDDDDDDD

423   LLLLLKDDDDCCCIIIIIKDDDDD
      ||||||||||    ||||||||||
505   LLLLLKDDDDDDD-IIIIKDDDDD
```

Figure 63 Convert the representation of the alignment of the state sequence into the amino acid sequence. A vertical bar indicates that the two labels match and a horizontal bar among the sequences means a gap is inserted.

Another example is Annie Lennox’s “A Whiter Shade of Pale.” The fixed-state HMM shows that the section from 105 s to 289 s is a replication of the section from 3 s to 181 s, and by applying the SSS HMM method, the section from 7 s to 140 s is found to align with the section from 113 s to 248 s. The similarity of the chromagrams of the aligned sections is shown in Figure 64, in which the upper two figures correspond to the chromagram of the sub-sections extracted by fixed-state HMM, and the lower figures correspond to the sub-sections obtained by the SSS method. Both fixed-state HMM and SSS HMM capture the recurring parts well.

However, SSS HMM is preferred, as the sections derived from the state are more compact and have less overlap.

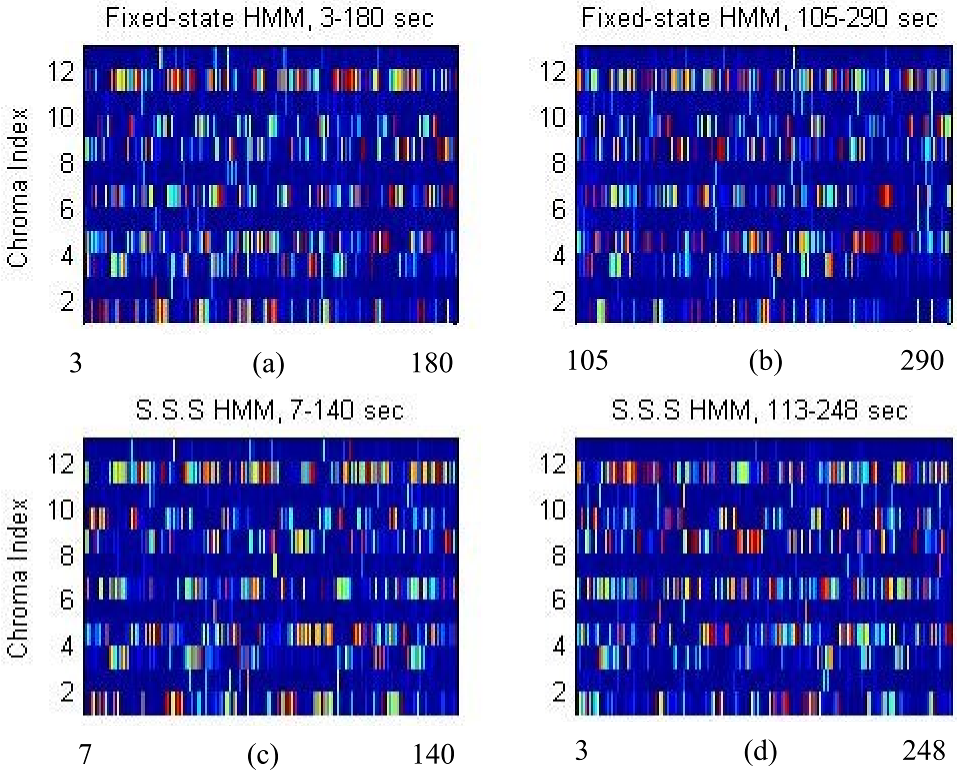


Figure 64 Chromagrams of repeating sections' of Annie Lennox's "A Whiter Shade of Pale": (a) 3 s–180 s and (b) 105 s–290 s using fixed-state HMM; (c) 7 s–140 s and (d) 113 s–248 s using SSS HMM

We found that the extracted repeating sections sometimes contain an overlap section. In general, there are two kinds of overlap, as shown in Figure 65 and Figure 67, in which section A and section B represent the aligned sections of the sub-optimal self-alignment. First, if the length of the overlap is short in Figure 65, subsections 1, 2, and 3 are aligned with subsections 3, 4, and 5, respectively, and subsection 3 is the overlap. Only subsections 1 and 2 are retained for music summarization owing to the similarity of subsections 1 and 3. Annie Lennox's "A Whiter Shade of Pale" is used as an example; the chromagrams of each subsection are shown in Figure 66. When the length of the overlap is around half the length of the aligned sections, as in Figure 67, subsections 1, 2, and 3 are similar; hence, only subsection 1 has to be retained. Japan's Quiet Life's "All Tomorrow's Parties" and Annie Lennox's "Don't Let It Bring You Down" are examples of such a case; the chromagrams of Annie Lennox's "Don't Let It Bring You Down" corresponding to the subsections 1, 2 and 3 in Figure 67 are shown in Figure 68. For more illustrations, please refer to [47].

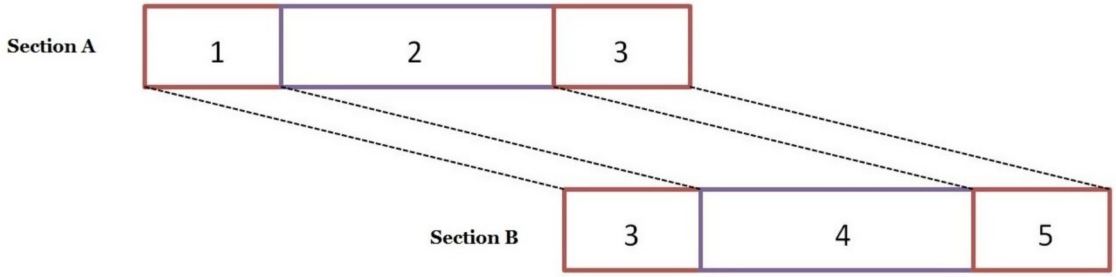
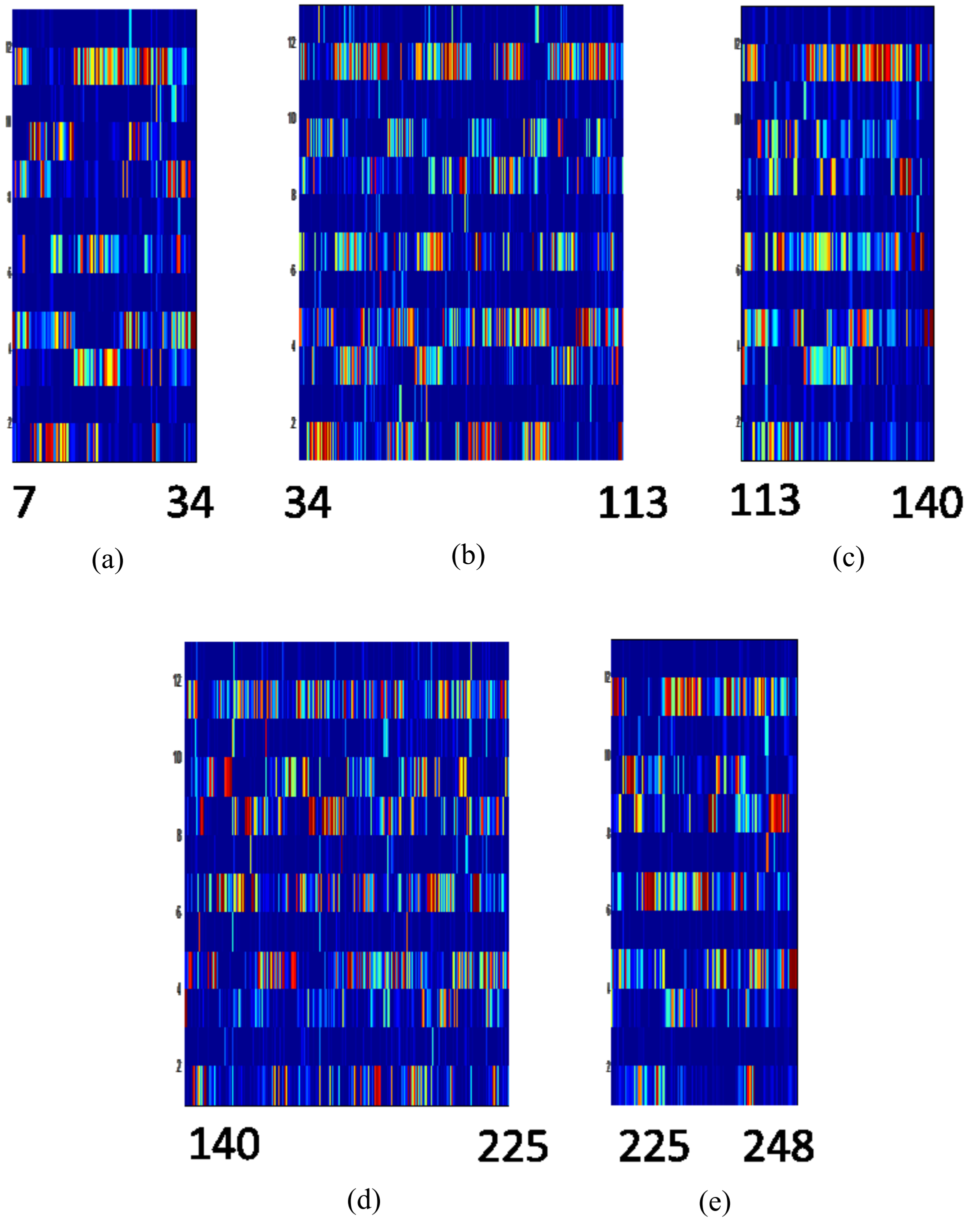


Figure 65 Assume Section A is aligned with Section B and Subsection 3 is the overlap.
The length of the overlap is short. Subsections 1, 2, and 3 are aligned with subsections 3, 4, and 5, respectively, and subsection 3 is the overlap. That subsection 1 of the section A is aligned with subsection 3 of section B and subsection 3 of section B and section A is the overlap implies that subsections 1 and 3 are similar. In addition, subsections 3 and 5 are similar due to subsection 3 of section A being aligned with subsection 5 of section B. Hence only subsections 1 and 2 are retained for music summarization owing to the similarity of subsections 1, 3, and 5.



(a) (b) (c)

(d) (e)

Figure 66 (a), (b), (c), (d), and (e) Chromagrams of Annie Lennox's "A Whiter Shade of Pale" corresponding to subsections 1, 2, 3, 4, and 5, respectively, in Figure 65

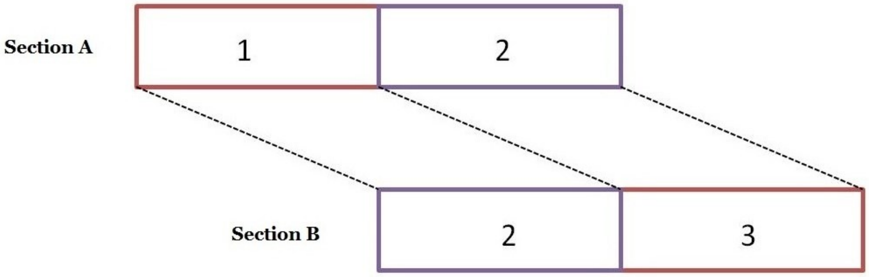


Figure 67 Assume Section A is aligned with Section B and subsection 3 is the overlap.
The length of the overlap is around half the length of the aligned section. Similar to (a), we can conclude that subsections 1, 2, and 3 are similar; hence, only subsection 1 has to be retained.

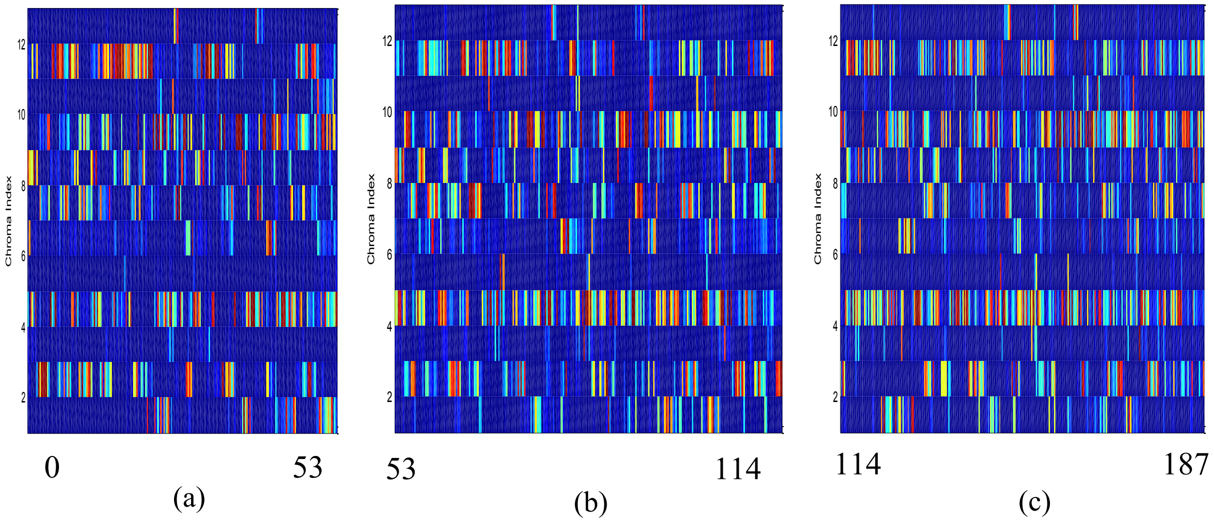


Figure 68 (a), (b), and (c) Chromagrams of Annie Lennox's "Don't Let It Bring You Down," corresponding to the subsections 1, 2, and 3 in Figure 67.

4.3.3 Cover Song Identification

To compare our results with published data, we evaluated our system's performance using the 80 pairs of song specified in [46]. In each pair, one song was selected as the reference song and the other reserved as a query song. In other words, the 160 songs were separated into two sets, namely a reference set and a query set. Each set contained 80 songs. For each query song, one of the songs in the reference set was the correct cover version of the query song. The purpose of the task was to identify the correct cover version from the reference candidates for every query song.

The fixed-state HMM method identified 41 out of 80 query songs (51.25%), giving a mean reciprocal rank (MRR) of 0.559 at a gap penalty of 1, and SSS HMM identified 40 out of 80 query songs (50%) to give an MRR of 0.556 at a gap penalty of 0.5. The performance of the different methods is summarized in Table 5. The precision and recall data of our method, with the alignment score normalized such that the sum of the score vector for each query song is equal to 1, are shown in Figure 69. When the normalized score is above 0.028, the precision is about 85%. In general, the sequence alignment methods perform better than Ellis' method [48]. Despite the similarity in the performance of the sequence-alignment methods, the complexity of using CHC is lower than that of using the BSC or SSS HMM method. In addition, the gap penalty of SSS HMM has to be set at a smaller value than that of fixed-state HMM. We may conclude that the SSS method is more sensitive to tempo variations. Therefore, local alignment using fixed-state HMM is preferred for the task of identifying cover songs.

Table 5 Performance of different methods in the cover song identification task

Method	Sequence Alignment			Correlation
	CHC by HMM	CHC by SSS HMM	BSC	BSC
Precision	51.25%	50%	53.75%	38.75%
MRR	0.559	0.556	0.57	0.44

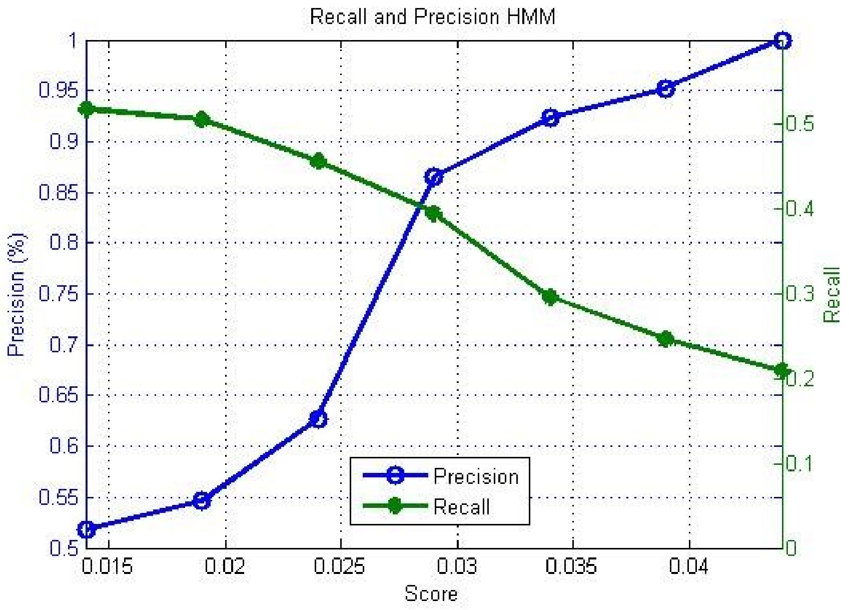


Figure 69 Recall and precision of our method. When the normalized score is above 0.028, the precision is about 85%.

4.3.4 Tolerance of Tempo Variations

In order to investigate the effect of tempo variations in our method, 15 songs, such as The Beatles' "We Can Work It Out," were chosen. The tempo of these songs was then altered by up to ±20% of the original tempo using "Audacity" software [49]. Following the procedure depicted in Figure 55, an alignment score between the tempo-changed version and the original song was obtained. Figure 70 shows that the percentage of the 15 songs' average score decreases as the tempo deviates from the original tempo at a gap penalty of 1. When the deviation is 0%, the song is self-aligning and always yields the highest score, and the gap penalty is set to 1. The more the tempo deviates from the original, the more the score decreases. When the tempo deviates to ±20%, the scores decrease to 60% of the original. As we do not know the ground truth of how similar the tempo of a query song and its cover version is, the tolerance of the tempo variation between a query song and its cover version cannot be reported. However, the margin between the average score of the correctly identified cover pairs and the average score of the misidentified songs is about 39%, as evaluated using Ellis' database [46]. This may suggest that our method should have at least 5% tolerance to the tempo variation on average.

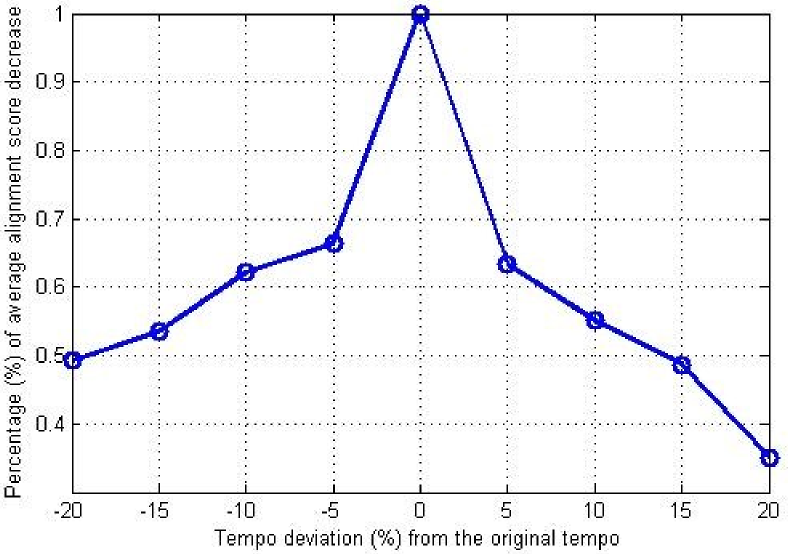


Figure 70 Percentage of the decrease in the alignment score as the tempo deviates from the original tempo.

4.3.5 Extraction of Similar Melodies from Two Songs

Recently, a YouTube video called "Singsing Rabbit" [50] caught our attention. This song was composed by combining sections of ten songs flawlessly. In order to check whether our proposed methods can extract the similar melodies from the original ten songs and Singsing Rabbit, the chromagrams of all of the songs were extracted. To overcome the key variation, the chromagrams of the ten original songs were cyclically rotated and then the state sequences of the songs were decoded by the given HMM of Singsing Rabbit (which is the only trained HMM in this case). Finally, the state sequence of Singsing Rabbit was aligned with the state sequences of its ten constituent songs. The aligned parts were recoded for listening. Four and six songs were identified correctly by fixed-state HMM and SSS HMM, respectively. We found that in "Singsing Rabbit" at least two sections taken from the original songs are short (one or two phrases), so that our method could not detect them. In addition, comparing the two HMM methods, the SSS HMM performs slightly better than the fixed-state HMM method. Finally, the start and end points of the correctly identified sections are more accurate using the SSS HMM than the fixed-state HMM method.

In order to compare the two HMM methods in more detail, we made a new song by connecting segments from 23 songs. We selected a short segment, normally a phrase of around 5 s in length, from each of the 23 songs. These sections were then connected to make the new song. The HMM of the new song was trained; thus, the most likely state sequences of the new song and the original 23 songs were obtained. Finally, the sequence alignment method was applied to the new song and the 23 original songs to test the ability of our methods to capture the matched melodies of the new song and the original songs. Unfortunately, the fixed-state HMM method was unable to identify correctly any section of the original songs in the new song, whereas the SSS method could identify 14 out of the 23.

Similar songs, unlike cover songs, are different songs with some sections that sound similar. The similar sections can be short. For example, the similarity between Coldplay's "The Scientist" and Taylor Swift's "Haunted" is limited to the introduction. Of the 26 pairs of similar songs listed on the "Sounds Just Like" website [35], 19 were selected as our test samples. Our SSS method was employed to extract the sections of these 19 pairs that resemble each other. The extracted sections were listened to carefully, and 15 of the 19 extracted sections were perceived to be similar. The four mismatched pairs are Green Day's "Waiting" and Petula Clark's "Downtown," REM’s “It’s the End of the World As We Know It” and Bob Dylan’s “Subterranean Homesick Blues,” Led Zeppelin’s “Babe I’m Gonna Leave You” and Chicago’s “25 or 6 to 4”, and Disney’s “A Spoonful of Sugar” and John Williams’ “The Imperial March” from the film The Empire Strikes Back. Although the author of the website claims these four pairs of songs sound similar, we believe that they are similar in terms only of atmosphere, not of melodies. An interesting result is worth mentioning. The similarity between The Beatles’ “Tomorrow Never Knows” and The Chemical Brothers’ “Let Forever Be” and between Fats Domino’s “I’m Walkin” and Will Smith’s “Switch” lies in the background drums, and the extracted excerpts are coincident with this property.

5 Conclusions

In this dissertation, we have described preliminary research on data-driven music chord-sequence analysis. We introduced some of the features used in content-based music analysis and showed that a combination of MFCC and ZCR is the best feature set for clustering different genres. We also presented a robust representation of music that is purely data-driven and utilizes melody (chromagram) as the basic feature. The representation is computed in a completely unsupervised manner allowing the data to dictate the nature of the “states” that are learned by HMM. This allows us to capture what is important in a song, i.e., the states, and how they interplay over time, i.e., the sequence. Our representation allows us to use a “local” alignment/matching algorithm that enables us to perform approximate matching and compute the similarity of music. We showed that for the cover song identification application our method performs better than the best state-of-the-art algorithm in the literature. We also presented two other applications where the “approximate matching” capability allows us to summarize a song or break up a song in terms of well-known melodies. Future work can apply our method to different genres and use complementary features, such MFCC, to augment the capabilities of the chroma-based representation that we have used.

6 Appendix

Table 6 Songs used in the section of 3.3.4.2

Symphony	
1	Herbert Von Karajan: Brahms Symphony No.1 - I Un poco sostenuto-Allegro
2	Herbert Von Karajan: Brahms Symphony No.1 - II Andante sostenuto
3	Herbert Von Karajan: Brahms Symphony No.1 - III Un poco allegretto e grazioso
4	Herbert Von Karajan: Brahms Symphony No.1 - IV Adagio-Allegro no troppo, ma con brio
5	Herbert Von Karajan: Haydn Symphony No.103, Drumroll - I Adagio-Allegro con spirito
6	Herbert Von Karajan: Haydn Symphony No.103, Drumroll - II Andante piu tosto allegretto
7	Herbert Von Karajan: Haydn Symphony No.103, Drumroll - III Menuetto
8	Herbert Von Karajan: Haydn Symphony No.103, Drumroll - IV Finale Allegro con spirito

9	Herbert Von Karajan: Haydn Symphony No.104, London - I Adagio-Allegro
10	Herbert Von Karajan: Haydn Symphony No.104, London - II Andante
11	Herbert Von Karajan: Haydn Symphony No.104, London - III Menuetto Allegretto
Piano	
12	Robert Casadesus_A la maniere de...Borodine (Valse)[1913]
13	Robert Casadesus_I. Modere
14	Robert Casadesus_I. Noctuelles
15	Robert Casadesus_I. Pavane de la Belle au bois dormant
16	Robert Casadesus_II. Mouvement de Menuet
17	Robert Casadesus_II. Oiseaux tristes
18	Robert Casadesus_II. Petit Poucet
19	Robert Casadesus_III. Anime
20	Robert Casadesus_III. Une Barque sur l'ocean
21	Robert Casadesus_IV. Alborada del gracioso
22	Robert Casadesus_Pavane pour une infante defunte [1899]
Violin	
23	Nicolo Paganini: 24 Caprices Op. 1 For Solo Violin No.1 In E
24	Nicolo Paganini: 24 Caprices Op. 1 For Solo Violin No.2 In B Minor
25	Nicolo Paganini: 24 Caprices Op. 1 For Solo Violin No.3 In E Minor
26	Nicolo Paganini: 24 Caprices Op. 1 For Solo Violin No.4 In C Minor

27	Vanessa Mae: Partita No. 3 in E For Solo Violin BWV 1006 I. Preludio
28	Vanessa Mae: Partita No. 3 i E For Solo Violin BWV 1006 II. Loure
29	Vanessa Mae: Partita No. 3 in E For Solo Violin BWV 1006 III. Gavotte en Rondeu
30	Vanessa Mae: Partita No. 3 in E For Solo Violin BWV 1006 IV. Menuet I
31	Vanessa Mae: Partita No. 3 in E For Solo Violin BWV 1006 V. Menuet II
32	Vanessa Mae: Partita No. 3 in E For Solo Violin BWV 1006 VI. Bourree
33	Vanessa Mae: Partita No. 3 in E For Solo Violin BWV 1006 VII. Gigue
Male	
34	Morrissey: All You Need Is Me
35	Morrissey: Everyday Is Like Sunday
36	Morrissey: First of the Gang to Die
37	Morrissey: I Have Forgiven Jesus
38	Morrissey: I Just Want to See the Boy Happy
39	Morrissey: In the Future When All's Well
40	Morrissey: Irish Blood, English Heart
41	Morrissey: Let Me Kiss You
42	Morrissey: Redondo Beach
43	Morrissey: Suedehead
44	Morrissey: That's How People Grow Up
Female:	
45	Leona Lewis: A Moment Like This

46	Leona Lewis: Angel
47	Leona Lewis: Better in Time
48	Leona Lewis: Bleeding Love
49	Leona Lewis: Footprints in the Sand
50	Leona Lewis: Here I Am
51	Leona Lewis: Homeless
52	Leona Lewis: I Will Be
53	Leona Lewis: I'm You
54	Leona Lewis: Take A Bow
55	Leona Lewis: The Best You Never Had
Choirboys	
56	The Choirboys_Corpus Christi Carol
57	The Choirboys_Danny Boy Carrickfergus
58	The Choirboys_Do You Hear What I Hear
59	The Choirboys_Ecce Homo
60	The Choirboys_He Ain't Heavy, He's My Brother
61	The Choirboys_In Paradisum
62	The Choirboys_Panis Angelicus
63	The Choirboys_Pie Jesu
64	The Choirboys_Tears in Heaven
65	The Choirboys_The Lord Bless You and Keep You
66	The Choirboys_The Lord Is My Shepherd (Psalm 23)

Sarah Brightman	
67	Sarah Brightman_Attesa
68	Sarah Brightman_Canto Della Terra (feat. Andrea Bocelli)
69	Sarah Brightman_Chanson D'enfance
70	Sarah Brightman_Heaven Is Here
71	Sarah Brightman_Let It Rain
72	Sarah Brightman_Pasion (feat. Fernando Lima)
73	Sarah Brightman_Pie Jesu From Requiem
74	Sarah Brightman_Schwere Traume
75	Sarah Brightman_Storia Damore
76	Sarah Brightman_Symphony
77	Sarah Brightman_The Music of the Night

Table 7 Songs used in the section of 3.3.4

Index	Abbreviation	Singer or player	Type	# of songs
1	'Christina'	Christina Aguilera	Female Pop	21
2	'Leona'	Leona Lewis	Female Pop	14
3	'Classical'	Herbert Von Karajan	Symphony	11
4	'Choirboys'	The Choirboys	Religious	11
5	'Piano'	Robert	Pure music	12

6	'Sarah'	Sarah Brightman	Female Pop & Classical	11
7	'Violin'	Nicolo Paganini & Vanessa Mae	Pure music	11
8	'Morrissey'	Morrissey	Male Pop & rock	15
9	'Daniel'	Daniel Powter	Male Pop	11
10	'Mariah'	Mariah Carey	Female Pop	14
11	‘String’	String Essentials	Pure music	7
12	'Usher'	Usher	Male R&B	18

Table 8 Songs used in the section of 3.3.4.3

Index	Abbr.	Singer or player	Type	Song index	# of songs
1	F1	Christina Aguilera	Female Pop	1-21	21
2	F2	Leona Lewis	Female Pop	22-35	14
3	F3	Sarah Brightman	Female Pop & Classical	36-46	11
4	F4	Mariah Carey	Female Pop	47-60	14
5	CB	The Choirboys	Religious	61-71	11
6	M1	Morrissey	Male Pop & Rock	72-86	15

7	M2	Daniel Powter	Male Pop	87-97	11
8	M3	Usher	Male R&B	98-115	18
9	Rap	Hurricane	Male Rap	116-129	14
10	Jaz1	Fourplay	Jazz	130-138	9
11	Jaz2	Sonny Rollins	Jazz	139-147	9
12	Cla	Herbert Von Karajan	Symphony	148-158	11
13	Pia	Robert	Pure music	159-170	12
14	Vio	Nicolo Paganini & Vanessa Mae	Pure music	171-181	11
15	Str	String Essentials	Pure music	182-188	7

The features are extracted by MIRtoolbox from University of Jyväskylä.

The SOM Toolbox by Esa Alhoniemi, Johan Himberg, Juha Parhankangas and Juha Vesanto is used.

7 References

[1] B. Logan, “Mel Frequency Cepstral Coefficients for Music Modeling,” *International Symposium on Music Information Retrieval*, 2000.

[2] E. Gómez, “Tonal Description of Polyphonic Audio for Music Content Processing,” *INFORMS Journal on Computing*, vol. 18, no. 3, pp. 294–304, Jun. 2006.

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

[4] E. Pampalk, A. Rauber, and D. Merkl, “Content-based Organization and Visualization of Music Archives,” 2002, pp. 570–579.

[5] V. Venkatachalam, L. Cazzanti, N. Dhillon, and M. Wells, “Automatic Identification of Sound Recordings,” *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 92– 99, Mar. 2004.

[6] M. Wells, V. Venkatachalam, L. Cazzanti, K. F. Cheung, N. Dhillon, and S. Sukittanon, “Automatic Identification of Sound Recordings,” U.S. Patent 732815305-Feb-2008.

[7] M. Wells, V. Venkatachalam, L. Cazzanti, K. F. Cheung, N. Dhillon, and S. Sukittanon, “Automatic Identification of Sound Recordings,” U.S. Patent 788193101-Feb-2011.

[8] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. Whitman, “A Large-Scale Evaluation of Acoustic and Subjective Music Similarity Measures,” *Computer music Journal*, vol. 28, 2003.

[9] J. Foote, “Visualizing Music and Audio Using Self-Similarity,” *ACM Multimedia*, pp. 77–80, 1999.

[10] J. Aucouturier and F. Pachet, “Music Similarity Measures: What’s The Use ?,” *Proceedings of the 3rd International Symposium on Music Information Retrieval*, 2002.

[11] M. Mandel, M. I. M, and D. Ellis, “Song-Level Features and Support Vector Machines For Music Classification,” *Proceedings of the 6th International Conference on Music Information Retrieval*, 2005.

[12] D. Pye, “Content-Based Methods for the Management of Digital Music,” in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000, vol. 6, pp. 2437–2440 vol.4.

[13] A. Berenzweig, D. P. W. Ellis, and S. Lawrence, “Using Voice Segments to Improve Artist Classification of Music,” 2002.

[14] B. Whitman, G. Flake, and S. Lawrence, “Artist Detection in Music with Minnowmatch,” in *Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop*, 2001, pp. 559–568.

[15] G. Williams and D. P. W. Ellis, *Speech/Music Discrimination Based On Posterior Probability Features*. 1999.

[16] T. Zhang and T. Zhang, “Semi-automatic Approach for Music Classification,” in *in Proc. SPIE Conf. on Internet Multimedia Management Systems*, 2003, pp. 81–91.

[17] Karin Kosina, “Music Genre Recognition,” Hagenberg, 2002.

[18] Ricardo Malheiro, “Classification of Recorded Classical Music using Neural Networks.”

[19] S. Haykin, *Neural Networks: A Comprehensive Foundation*. MacMillan Publishing Company, 1994.

[20] S. Z. Li and G. Guo, *Content-Based Audio Classification and Retrieval Using SVM Learning*. .

[21] J. Foote, “Automatic Audio Segmentation using a Measure of Audio Novelty,” in *2000 IEEE International Conference on Multimedia and Expo*, 2000, vol. 1, pp. 452–455 vol.1.

[22] J. T. Foote and M. L. Cooper, “Media Segmentation using Self-Similarity Decomposition,” in *In Proc. SPIE Storage and Retrieval for Multimedia Databases*, 2003, pp. 67–75.

[23] B. Logan and S. Chu, “Music Summarization using Key Phrases,” in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP ’00. Proceedings*, 2000, vol. 2, pp. II749–II752 vol.2.

[24] J. J. Aucouturier and M. Sandler, “Segmentation of Musical Signals Using Hidden Markov Models,” *Audio Engineering Society Convention*, 2001.

[25] A. Sheh and D. Ellis, “Chord Segmentation and Recognition Using EM-Trained Hidden Markov Models,” *Proc. ISMIR*, 2003.

[26] K. Lee, “Automatic Chord Recognition from Audio Using an HMM with Supervised Learning,” *Proc. ISMIR*, vol. 2006, 2006.

[27] S. Van De Par, M. Mckinney, and A. Redert, “Musical Key Extraction from Audio Using Profile Training,” *Proceedings of the 7th International Conf. on Music Information Retrieval*, 2006.

[28] J. Bello and J. Pickens, “A Robust Mid-Level Representation for Harmonic Content in Music Signals,” 2005.

[29] T. Fujishima, “Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music,” presented at the ICMA, editor, International Computer Music Conference, 1999, pp. 464–467.

[30] M. Wells, V. Venkatachalam, L. Cazzanti, K. F. Cheung, N. Dhillon, and Sukittanon Somsak, “Automatic Identification Ssound Recordings,” U.S. Patent 732815305-Feb-2008.

[31] D. P. . Ellis and G. Poliner, “Identifying ‘Cover Songs’ with Beatsynchronous Chroma Features,” *Music Information Retrieval Evaluation eXchange*, Jan. 2006.

[32] J. P. Bello, “Audio-based Cover Song Retrieval Using Approximate Chord Sequences: Testing Shifts, Gaps, Swaps and Beats,” *Int. Symp. on Music Information Retrieval (ISMIR)*, pp. 239–244, 2007.

[33] W. Tsai, H. Yu, and H. Wang, “A Query-By-Example Technique for Retrieving Popular Songs with Similar Melodies,” *Proceedings of ISMIR*, 2005.

[34] F. E. Soulez, F. Soulez, and I. C. Pompidou, “Improving Polyphonic and Poly-Instrumental Music to Score Alignment,” *ISMIR*, pp. 143–148, 2003.

[35] “Songs that Sound Like Other Songs - Sounds Just Like.” [Online]. Available: http://soundsjustlike.com/. [Accessed: 22-Aug-2011].

[36] J. Bello and J. Pickens, “A Robust Mid-Level Representation for Harmonic Content in Music Signals,” *ISMIR*, 2005.

[37] C. Roads, *The Computer Music Tutorial*. Mit Press, 1996.

[38] H. Fastl and E. Zwicker, *Psychoacoustics: Facts and Models*. Springer, 2007.

[39] M. A. Bartsch and G. H. Wakefield, “Audio Thumbnailing of Popular Music Using Chroma-Based Representations,” *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 96 – 104, Feb. 2005.

[40] H. Purwins, “A New Method for Tracking Modulations in Tonal Music in Audio Data Format,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN’00)-Volume 6 - Volume 6*, Washington, DC, USA, 2000, p. 6270–.

[41] W. Verhelst and M. Roelands, “An Overlap-Add Technique Based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech,” in *, 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1993. ICASSP-93*, 1993, vol. 2, pp. 554 –557 vol.2.

[42] J. V. Libella Oy, “SOM Toolbox for Matlab 5,” *SOM Toolbox*, 2000. [Online]. Available: http://www.cis.hut.fi/somtoolbox/.

[43] C. Xu, N. C. Maddage, and Q. Tian, “Support Vector Machine Learning for Music Discrimination,” in *Proceedings of the Third IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, London, UK, UK, 2002, pp. 928–935.

[44] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[45] T. F. Smite and M. S. Waterman, “Identification of Common Molecular Subsequences,” *Journal of Molecular Biology*, 1981.

[46] D. P. W. Ellis, “The ‘Covers80’ Cover Song Data Set.”.

[47] C.-L. Wang, “Examples of Music Summary.” [Online]. Available: chihliwang.bol.ucla.edu/index1.html.

[48] Dan Ellis, “Music Beat Tracking and Cover Song Identification.” [Online]. Available: http://labrosa.ee.columbia.edu/projects/coversongs/.

[49] “Audacity.” [Online]. Available: http://audacity.sourceforge.net/.

[50] “Singsing Rabbit,” *YouTube*, 20-Jul-2011. [Online]. Available: http://www.youtube.com/watch?v=4tM3ZNN6dNM&feature=youtube_gdata_player. [Accessed: 22-Aug-2011].