# Multimodal Content-based Structure Analysis of Karaoke Music

Yongwei Zhu
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
ywzhu@i2r.a-star.edu.sg

Kai Chen
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
kchen@i2r.a-star.edu.sg

Qibin Sun
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
qibin@i2r.a-star.edu.sg

## ABSTRACT

This paper presents a novel approach for content-based analysis of karaoke music, which utilizes multimodal contents including synchronized lyrics text from the video channel and original singing audio as well as accompaniment audio in the two audio channels. We proposed a novel video text extraction technique to accurately segment the bitmaps of lyrics text from the video frames and track the time of its color changes that are synchronized to the music. A technique that characterizes the original singing voice by analyzing the volume balance between the two audio channels is also proposed. A novel music structure analysis method using lyrics text and audio content is then proposed to precisely identify the verses and choruses of a song, and segment the lyrics into singing phrases. Experimental results based on 20 karaoke music titles of difference languages have shown that our proposed video text extraction technique can detect and segment the lyrics texts with accuracy higher than 90%, and the proposed multimodal approach for music structure analysis method has better performance than the previous methods that are based only on audio content analysis.

## Categories and Subject Descriptors

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

**General Terms:** Algorithms, Design, Experimentation

**Keywords:** Music information retrieval, multimodality, video text detection, music structure analysis, Karaoke

## 1. INTRODUCTION

Karaoke is a popular form of entertainment, by which an amateur singer can sing along the playback of the music accompaniment while the lyrics are displayed and synchronized to the music with text color changes. Plenty of karaoke music titles are now available in Karaoke lounges, K-Boxes, and can be purchased as VCDs or DVDs from music shops for home use. With the growing powers of computers and advancement of media processing technologies, new ways of interaction to the karaoke music data beyond merely playing back and singing along are in demand. For instance, automatic detecting the structures of the song, such as the choruses and verses, can help a singer get familiar with and practice singing the song. In a Karaoke lounge, a user may want to search for a particular song of interest by singing an excerpt of the melody, and then decide to choose a title by browsing visual summaries of the lyrics and video. We are currently developing a multimedia music retrieval system with such applications in mind.

There has been some work on content-based music structure detection by using audio signal analysis [1-6]. In the earlier work [1-5], the choruses are detected by finding the most repeated sections, based on similarity measure of pitch or chroma-based features. These methods have difficulty in precisely locating the boundaries of the repeated sections. In a recent work [6], Maddage proposed an approach that detects similarity regions of both melodies and audio lyrics contents, and utilizes the music knowledge in the structure analysis, which effectively improve the detection accuracy (to 80%).

In karaoke music video, the lyrics of songs are embedded in the video frames, and the time of text color changes is synchronized to the music. The lyrics text, once extracted and tracked from the video, can be utilized in analyzing lyrical contents and detecting song structures. From the best of our knowledge, there is no existing work that uses lyrics text extraction and color change tracking for such purposes.

Shao [7] proposed a music video summarization technique, which detects and recognizes lyrics on the key frames of video shots. However, in music videos as well as karaoke music the occurrences of lyrics text may not be related to video shots. For instance, one video shot may contain multiple lines of lyrics and one line of lyrics may also span multiple video shots. Wang [8] has proposed a technique to automatically align the lyrics text to the singing voices of music, which can be used in the production of karaoke music video. In this paper, we shall present a technique that in part does the inverse of Wang's technique.

We propose a novel and multimodality approach for content-based structure analysis of karaoke music. The contributions of this paper are: (1) a novel video text analysis technique to extract the bitmaps of lyrics text from the video frames and track the time of its color changes that are synchronized to the singing voice; (2) a novel music structure (chorus/verse/phrase) analysis technique based on multimodality contents: lyrics text from video and original singing audio as well as accompaniment audio from the two audio channels. The choruses and verses are detected by analyzing the patterns of repetition of both the lyrics text and the

*MM'05*, November 6–11, 2005, Singapore.

melody. The singing phrase segmentation is based on the time alignment of the lyrics text with the detected singing voice.

The rest of this paper is organized as follows. The overview of the proposed approach is presented in section 2. The technique for lyrics text extraction and text color change tracking are presented in section 3. Section 4 presents the multimodality approach for music structure detection. Section 5 presents the experimental results. Section 6 presents the conclusion and discussion.

## 2. OVERVIEW

We are currently developing a retrieval system of karaoke music, which is based on the multimedia contents of the karaoke music data. Figure 1 illustrates the overview of our proposed approach for structure analysis of karaoke music. The input from the karaoke music data consists of video frame sequence, the audio signals for both the music accompaniment and the original singing. A lyrics text analysis method extracts the bitmaps of lyrics text from the video frames and tracks the text color changes. Beat length and location is detected from the accompaniment channel. A singing voice analysis method detects the presence of the singing voice in the beats based on the audio signals in the dual channels. Melody analysis characterizes the harmonic and melodic contents of the original singing audio. Music structure detection operates on the lyrics and the audio contents, and derives the music structures, i.e. choruses, verses and phrases. Shaded boxes in Figure 1 indicate the major contribution of this paper. The detected music structures can be used for content-based indexing, retrieval and summarizing of karaoke music.

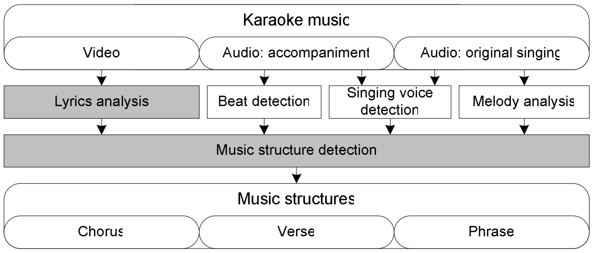


**Figure 1: Overview diagram of the multimodal content-based music structure analysis method**

## 3. LYRICS TEXT ANALYSIS

In this section, we present a technique to extract lyrics texts from the video frames and track the color changes of the text over time. In our approach, we localize and segment the text lines into bitmap images of characters. Text recognition (OCR) is not involved in this work, and so the technique is independent from languages. The color changes of lyrics text are represented as the time sequences of the extracted characters.

In the existing approaches for video text detection and segmentation [9-12], the texts are detected by analyzing the textures and edges in the image, and the characters are segmented by analyzing the colors of the text and the background. The redundancy of multiple video frames is then explored to refine the result, which is based on the assumption that the text keeps static over all the frames of its occurrence. This assumption, however, is not true for karaoke music video, in which the colors and contrasts of the lyrics texts can change over time while the background can contain large variation and dynamics. Thus the existing techniques can hardly handle the problem of lyrics extraction from Karaoke music video, and we have not seen any existing technique that tracks the color changes of lyrics text in karaoke music video.

In our proposed approach, we utilize the characteristics of the occurrence of lyrics text on the video frame: (1) the lyrics presents as text lines alternatively on two (or occasionally three) textboxes with fixed sizes and locations; and (2) the color of a text line changes in a way like a vertical bar scanning gradually from the left to the right during the occurrence of the text line (called text object in this paper), and (3) in each text object the text colors keep static (unchanged) before or after the color changes taking place.

Figure 2 illustrates textboxes, text lines, text objects and color changes of a text line. The subfigure (a) illustrates the textboxes that have fixed positions in the video frame. In most karaoke video, the text in the upper textbox is aligned to the left, whereas the text in the lower textbox is aligned to the right. The left half of the upper textbox and the right half of the lower textbox, as indicated by gray color, are mostly occupied by the texts. The subfigure (b) illustrates three text objects for a single textbox. Each text object contains only one text line, and the color of the text in a text line change from the left to the right as illustrated in subfigure (c).

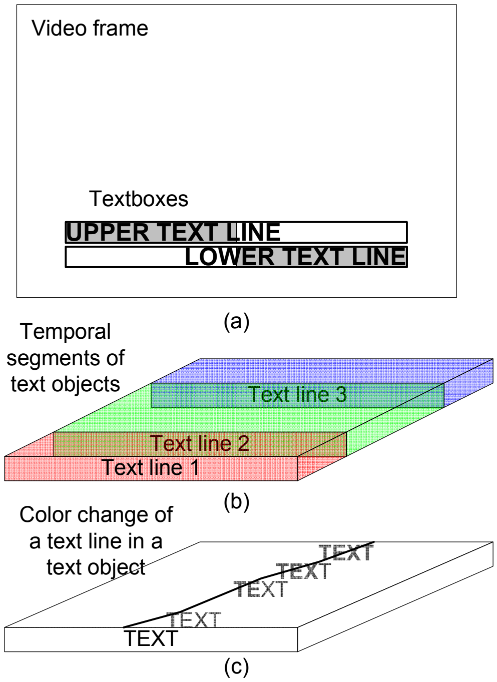


**Figure 2: Illustration of textbox, text line, text object, and text color change**

The steps for the proposed lyrics analysis technique are shown in Figure 3. In the first step, the positions of the textboxes are located for the whole Karaoke music title, and the presence of text lines in the textboxes for each video frame is detected. In the second step, the temporal boundaries of text objects are detected. In the third step, we track the color changes of a text line in a text object. In the final step, we extract the bitmaps of the text lines. The color change tracking is done before text bitmap extraction, because we utilize the characteristics of the text color changes for text extraction. The details are presented in the following.

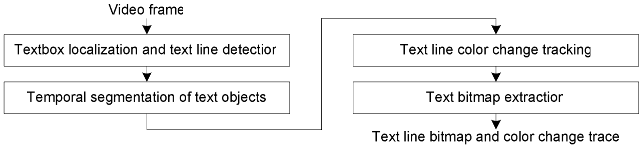

Video frame
Textbox localization and text line detectior
Temporal segmentation of text objects
Text line color change tracking
Text bitmap extractior
Text line bitmap and color change trace

**Figure 3: The steps in lyrics text analysis**

## 3.1 Textbox Localization and Text Line Detection

The textboxes have fixed locations in karaoke music video, thus it is convenient for analysis to initially localize the textboxes for the whole video title rather than individually for each video frame. Textbox localization has two aspects: the vertical positions and the horizontal positions. In this approach, these positions are determined by computing simple image texture features, the image gradient and its variances, based on the grayscale intensity images of the video frames.

We denote the intensity of pixels in a frame by $v(t,x,y)$, where $t \in [1,T]$, $x \in [1,X]$ and $y \in [1,Y]$, t, x, and y are the time, column and row indexes of the pixels and T, X and Y are the bounds of these indexes. The intensity *v* takes values between 0 and 1.

The horizontal gradient and vertical gradient are computed using equation (1) and (2).

$$G_h(t,x,y) = v(t,x+1,y) - v(t,x-1,y) \qquad (1)$$

$$G_v(t,x,y) = v(t,x,y+1) - v(t,x,y-1) \qquad (2)$$

We compute the approximate standard deviation of horizontal gradients across the pixels of each row of a frame efficiently by assuming the zero mean of gradients in a row, as shown in Equation (3).

$$D_h(t,y) = \sqrt{\sum_{x \in X} G_h^2(t,x,y)/(X-1)} \text{ ,where } y \in [1,Y] \qquad (3)$$

$D_h(t,y)$, $y \in [1,Y]$ defines a row-based texture profile of a frame. The profile typically has high values at the rows that are covered by texture busy objects like text lines.

We accumulate the row-based texture profile over all the frames of the video title to produce a general row-based texture profile. $\sum_{t \in [1,T]} D_h(t,y)$, $y \in [1,Y]$. Figure 4 illustrates an example of the general row-based texture profile. The vertical positions of textboxes are determined by detecting the boundaries of the salient lumps in the profile, using simple boundaries detection technique [9]. Regulation is imposed to ensure the heights of all the textboxes are equal.

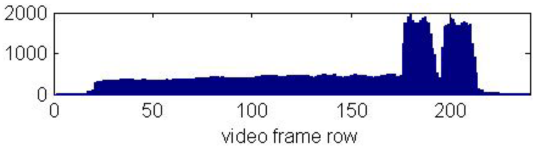


**Figure 4: General row-based texture profile for detecting vertical positions of textboxes**

Denoting the vertical bounds of the textboxes by $[Y_s^n, Y_e^n]$, where n ($n \in [1,2]$) is the index of the textboxes (e.g. upper or lower). The horizontal bounds of textboxes are detected by computing column-based texture profiles within the vertical bounds as in Equation (4).

$$D_v^n(t,x) = \sqrt{\sum_{y \in [Y_s^n, Y_e^n]} G_v^2(t,x,y)/(Y_e^n - Y_s^n)} \qquad (4)$$

And similarly the general column-based texture profile for a textbox is given by $\sum_{t \in [1,T]} D_v^n(t,x)$, $x \in [1,X]$. The general column-based texture profiles of the examples of two textboxes are illustrated in Figure 5. Subfigure (a) corresponds to the upper textbox and (b) corresponds to the lower textbox. Since the texts are left aligned in the upper textbox and right aligned in the lower textbox, the left and right bounds of textboxes are determined based on the upper and lower textbox respectively.

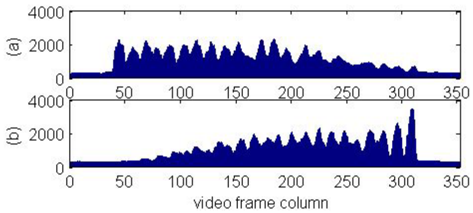


**Figure 5: General column-based texture profile for textboxes**

After textbox localization, the presence of text lines in the textboxes for each frame is detected in the left half or the right half of the textboxes, as indicated by the gray area in Figure 2 (a). This is because, for most of the time the texts occupy these areas. The text line presence detection is based on variation of the image gradients of the textbox area, as defined in Equation (5).

$$D(t) = \sqrt{\sum_{x \in [X_s, X_e]} \sum_{y \in [Y_s, Y_e]} \left(G_h^2(t,x,y) + G_v^2(t,x,y)\right) / (X_e - X_s) / (Y_e - Y_s)} \quad (5)$$

where t is the index of the video frame, $[X_s,X_e]$ and $[Y_s,Y_e]$ define the area of the corresponding detection area. The presence of text in the textbox is claimed if the value *D(t)* is above a threshold. Figure 6 illustrates the value of *D(t)* for the upper textbox of a karaoke title “paint my love”. It can be seen that text may be absent in the textbox in the middle of the song besides the very beginning and the end.

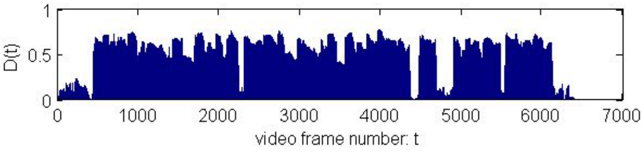


**Figure 6: Presence of text lines in a textbox for each frame**

## 3.2 Text Object Segmentation

Within the time duration that text lines are presented in textboxes, we temporally segment the frame sequence of text line occurrences into text objects (as illustrated in Figure 2(b)), each of which contains a single text line. So that color change tracking and bitmap extraction can be conducted for each text object in the later steps. The segmentation is based on the detection of abrupt change of the content in the detection area of a textbox. In this approach, the content change is computed based on the difference of texture profiles between consecutive frames as defined in Equation (6).

$$C(t)=\sum_{x\in[X_s,X_e]}\left(\left|D_v(t,x)-D_v(t-1,x)\right|\right) \qquad (6)$$

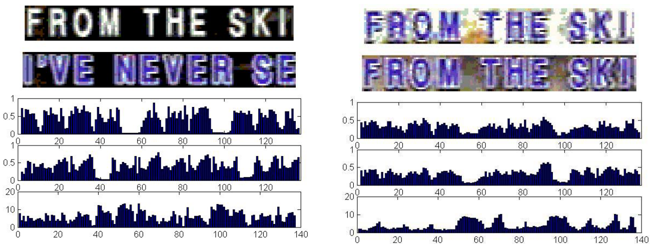
where D$_v$ is given in Equation (4), in which the textbox index n is omitted here for simplicity of presentation..

C(t) takes a high value if there is an abrupt content change in the detection area between two consecutive frames. We claim a content change based on the following criteria:

(a) $C(t)>threshold_1$; (b) $C(t)/\min(C(t-1),C(t+1))>threshold_2$. In (a), C(t) is above an absolute threshold value. In (b), C(t) is much higher than one of its adjacent values, whichever is lower. Condition (b) is used to eliminate false alarms that may rise when there are high continuous temporal dynamics in the background video.

Detection of content abrupt change, as mentioned above, can capture the sudden change of text content. However, it may also capture certain significant background changes, like background scene change, special editing effects or significant local motion. Thus it is desirable to distinguish these two types of content change: the text or the background.

We propose a method for the determination of the two types of content change by exploring their dynamical characteristics. For each detected content change, we compute three features: column-based texture profiles for the textbox before content change, the same feature for the textbox after content change, and a temporal block matching difference profile. The computation of these features is given later in this section. Our technique is based on the observation that in the case of background scene change the three features are still highly (positively or negatively) correlated, whereas for the case of text changes these three features exhibit uncorrelated or independent. Figure 7 illustrates the two cases, where (a) is for the case of text changes and (b) is for the case of background changes. In both of the two cases, the first two rows show the textbox after and before the detected content change, the third and forth rows show the corresponding column-based texture profiles and the last rows show the block matching difference profile. From subfigure (b), we can see that when there is only background scene change, the texture profiles before and after change are very similar to each other, and the texture profiles are also negatively similar to the block matching difference profile.



(a)

(b)

**Figure 7: Text change and background scene change**

Computations of the correlations of the profiles previously mentioned are shown in Equation (7-9).

$$Corr_1(t)=correlation\left(D_v(t,x),D_v(t-1,x)\right),\ x\in[X_s,X_e] \qquad (7)$$

$$Corr_2(t)=correlation\left(D_v(t,x)+D_v(t-1,x),B(t,x)\right),\ x\in[X_s,X_e] \qquad (8)$$

$$B(t,x)=\sum_{y\in[Y_s,Y_e]}\left|v(t,x,y)-v(t-1,x,y)\right|,\ x\in[X_s,X_e] \qquad (9)$$

In Equation (7) and (8) correlation() computes the correlation coefficients of the two variables. Corr$_1$ characterizes the correlation between the texture profiles and Corr$_2$ characterize the correlation between the mean texture profile and the block matching difference profile that is defined in Equation (9). The content change in the textbox detection area is claimed a background scene change, if Corr$_1$ is above a threshold with a positive value and Corr$_2$ is below a threshold with a negative value.

Figure 8 shows an example of text lines change detection. In the figure, (a) shows the profile difference *C(t)*; (b) shows the detected content change; (c) and (d) show the two correlation coefficient values Corr$_1$ and Corr$_2$ for the corresponding content changes; (e) shows the text change detection after removing the claimed background scene changes, and (f) shows the ground truth of text content change.

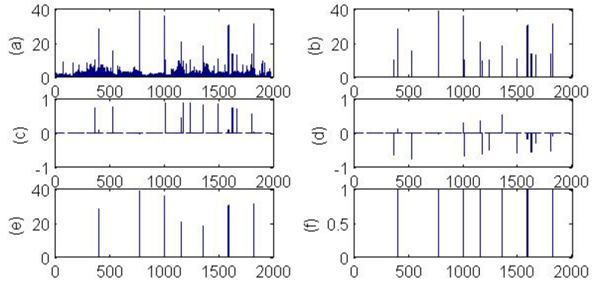


**Figure 8: Text line change and background change detection**

Temporal boundaries of a text object are defined by the detected text content change. Occasionally a text object may be over-segmented into more than one text objects. These cases can be corrected in the later text color change tracking step.

## 3.3 Text Color Change Tracking

In this step, we propose a technique to track the color change of a text line within a text object. In the tracking, we are interested in identifying the spatial locations (frame column number) and the temporal locations (frame number) of the occurrences of text color changes. This approach does not rely on detecting the text colors, since text color prototypes are difficult to build in video [9]. Instead, we utilize the characteristics and the spatial temporal constraints of the text color changes of karaoke music video: (1) the change of intensity of pixels of the text line is quite significant; (2) the color change occurs simultaneously for a column in the textbox; and (3) the color changes occur sequentially from the left to the right.

The proposed color change tracking technique takes 3 steps: (1) computing a spatial temporal textbox color change matrix; (2) finding the trace in the matrix that has maximal accumulated change values using a dynamic programming technique; and (3) finalizing the trace by value interpolation.

The color change matrix *M* characterizes the changes taking places in the spatial temporal range of the text object. The matrix has the dimension of the number of frames $N_f$ and number of columns $N_c$ of the text object. The values of the matrix are computed as follows. At each frame, the image difference between the current frame and the previous frame is computed within the textbox. For each row of the textbox image difference, the column with maximal absolute difference value across the row is located. A pixel $(x,y)$ is picked up from the textbox image difference, if the following two conditions are met: (1) the difference value at $(x,y)$ is maximal across the whole textbox; and (2) there are at least $n$(=3) rows in the textbox, which have their maximal difference values at the column $y$. The condition (2) is set, because the color changes occur simultaneously for the whole column of the textbox, and this can filter out many cases of background content change. The difference value at $(x,y)$ is set to the color change matrix $M(t,x)$, where $t$ is the time index of the current frame and $x$ is the column index of the picked up pixel in the textbox. $M(t,x)$ is set to zero, at positions where the above two conditions cannot be met. Figure 9 shows the first frame and last frame of a text object, as an example. The color change matrix is shown in Figure 10. It can be seen that many of the color changes due to lyrics text color change are identified. But some changes due to background changes are also presented.

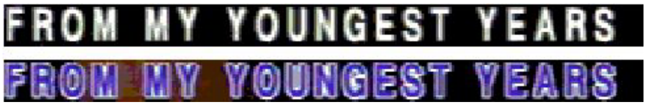


**Figure 9: The first and last frame of a text object**

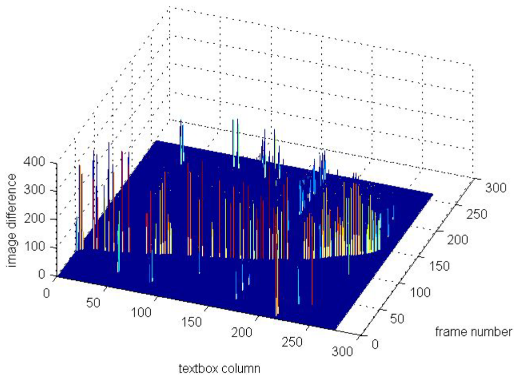
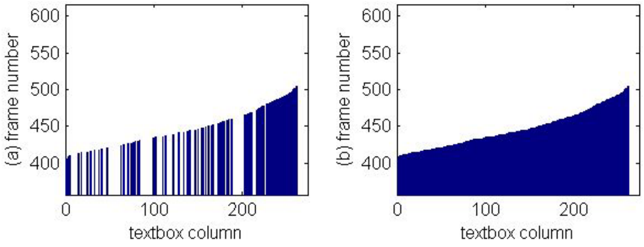


**Figure 10: Color change matrix of a text object**

Now we would like to pick up the color changes due to the lyrics text change from the matrix and filter out those due to the background change. This is done by a dynamic programming technique, which exploits the temporal spatial constraints of the lyrics text color changes, which is described as follows.

A table $S(t,x)$ of the same dimensions of matrix $M$ is constructed to find a trace in the table that has maximal accumulated change value from $M(1,1)$ to $M(N_f,N_c)$. In this dynamic programming setup, the previous cells $S(t_p,x_p)$ for the current cell $S(t,x)$ need to meet the the following conditions: (1) $t_p<t$, $x_p<x$, (2) both $M(t_p,x_p)$ and $M(t,x)$ are non zero, and (3) $(x-x_p)/(t-t_p)<threshold_3$. The condition (1) ensures the trace is formed according to the temporal spatial constraints. The condition (2) is imposed since only non zero cells should be considered to form the trace. And the condition (3) is imposed since the color changes of a text line cannot be faster than a certain speed (e.g. 15 pixels per frame for MPEG1 NTSC video). Among the previous cells, the one with maximal accumulated change value is chosen for the current cell, and the accumulated change value for the current cell is obtained by adding up the local change value $M(t,x)$. Starting from cell $S(1,1)$ that is initialized to 0, the cells of the table $S$ are computed in the temporal spatial sequential order. Then by finding the maximal accumulated difference value in $S$ and tracing the previous cells, the trace of lyrics text color change is obtained. This tracing technique effectively eliminates the isolated sparks in the color change matrix, which are due to the dynamics of the background.

The tracing result of the above example is shown in Figure 11(a). It can be seen that some color changes of lyrics text are missed in the trace. However the missed values can be easily recovered by interpolation based on the detected values, which is shown in Figure 11(b). The trace basically characterizes the spatial temporal position of a vertical bar that controls the color changes of a lyrics text line.



(a) (b)

**Figure 11: Text line color change tracking result**

As mentioned in the above, a text object may be temporally over segmented due to misclassifying a background scene change to text content change. Such cases can be detected and rectified by inspecting the relations of color change traces of the consecutive text objects. If the traces of color changes of two consecutive text objects are spatially and temporally connected, then they should be combined.

## 3.4 Text Bitmap Extraction

The bitmaps of the lyrics text are extracted based on the dynamical characteristics of the text: the pixel value keeps static either before or after the color change occurs. So for each pixel $(x,y)$ in the textbox, four features are computed: mean and standard deviation of the pixel value before and after the color change, which are denoted by $m_b(x,y)$, $m_a(x,y)$, $d_b(x,y)$, $d_a(x,y)$. A temporal margin of three frames at the beginning and the ending and around the color change is used in the computation to avoid precision errors of the boundaries. A pixel is classified to be part of lyrics text, if $|m_b(x,y)-m_a(x,y)|> d_b(x,y)+d_a(x,y)$. The result of the example is shown in Figure 12(a), in which the white color corresponds to the area of lyrics text. The isolated holes can be eliminated by image dilution process, which is shown in Figure 12(b). The bitmap of the text line after color change is shown in Figure 12(c).

After the text bitmap extraction, the color change trace can be finalized by taking only the columns that are covered by the text bitmap, as illustrated in Figure 12(d).

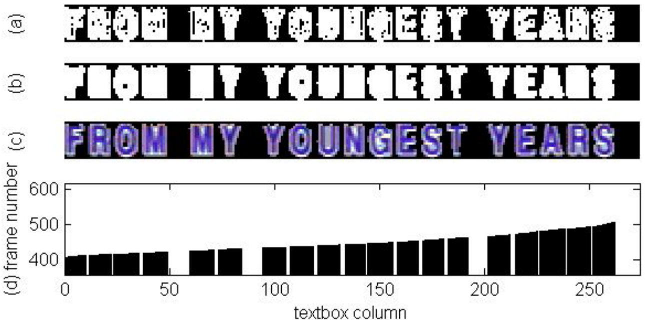


**Figure 12: Lyrics text bitmap extraction and trace finalizing**

Based on the extracted text bitmaps, the individual characters can be separated by estimating the character size and space size, which are typically uniform for the whole title. By employing the text color change trace, a spatial sequence of characters can be converted to a temporal sequence of characters. Figure 13 illustrates the time sequence of the segmented text characters, in which the time is the mean time of the pixel columns of a character. We use this temporal sequence of characters in detecting lyrics repetitions, as presented in the next section.

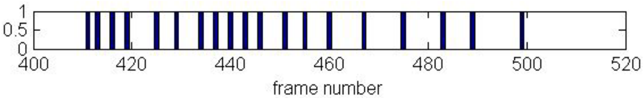


**Figure 13: Time sequence of the segmented text characters**

# 4. MUSIC STRUCTURE ANALYSIS

We propose a novel technique for music structure analysis based on multimodal contents of karaoke music. The verse and chorus are the main sections of songs. Locating and identify these section automatically is very useful for music semantic understanding and music retrieval. The verses and choruses are detected by analyzing the repetitions of lyrics and melody of the songs. The repetition of lyrics is detected based on similarity matching of the text bitmaps. The repetition of melody is computed based on melodic and harmonic similarity matching of the original singing audio. The time alignment of lyrics and singing voice is explored to segment the song lyrics into singing phrases.

## 4.1 Beat Detection

Beat is the regular time interval, on which the articulations of musical events are based. Beat detection is important for music audio processing, since the acoustic signal can be considered quasi-stationary in each beat. We used the beat detection technique proposed by Goto [13] to detect beat in the accompaniment audio channel. Detection error rate in accompaniment channel is lower than in the original singing channel, because the singing voice is a source of noise for note on-set detection. After beat detection, the audio signals are framed into windows of beat length, and the analysis is performed at each beat window.

## 4.2 Voice Detection

We detect the presence of vocal in the original singing audio channel by analyzing the signals in the two channels. Based on the observation that the audio contents in the two channels are almost identical for the non-vocal part, except for a volume balance difference, we proposed a straightforward method for voice detection in karaoke music.

The analysis is performed in each beat window. We first locate a number of the beats that contains no vocal, by using the lyrics text extraction result. These beats are found in either the time before the color change of the first text line taking place (the intro of the song) or the time gaps between the end of color change of a previous text line and the start of the color change of the following text line. A time margin of one beat is used for the end of color change of a text line to avoid the time inaccuracy for sustained long vocal note. Such beats are usually located at the intro, instrumental part, and verse sections.

The volume balance is then estimated from these located beat windows. The signals in the two channels are transformed to frequency domain by Constant Q Transform (CQT) [14], with 27.5 Hz initial frequency, 7-octave pitch range and 12-note-per-octave precision. A pitch profile with 84 components (7x12), similar to power spectrum, is then produced, which characterize the energy of each frequency band (note). The energy ratio between the original singing and the accompaniment for each band is computed for each beat window. The median value and standard deviation of the ratio among the selected beat window is computed. Figure 14 shows an example of the median and standard deviation for each band among 20 beat windows. From the values of the standard deviation, we derive that the ratio values for band 36 (220Hz) and above are reliable. And these bands also cover most of energies of the singing voice. So we obtain the channel volume balance by the ratio values from band 36 to 84, as shown in Figure 14 (a). And only these bands are used in the singing voice analysis. As the figure shows, the original singing channel has higher volume than the accompaniment channel for the high bands (as the ratios are larger than one).

After obtaining the channel volume balance, the accompaniment audio signals are compensated by being divided by the ratio values. Then the compensated accompaniment audio are subtracted from the original singing audio in the pitch profiles for each beat window. The original singing voice is characterized by the differences of the pitch profiles of the two audio channels. Figure 15 (a) and (b) illustrates the original singing channel and the compensated accompaniment channel. Figure 15 (c) shows the pitch profile difference in the interesting bands (36 to 84). The volume of the singing voice in each beat is then characterized by the sum of the energy of the interesting bands, which is shown in Figure 15 (d). By setting a threshold for the voice energy value, a beat window can be classified to vocal or non-vocal. Figure 15 (d) shows the first verse of “paint my love”, in which the singing voice starts from beat 17 and lasts for 4 beats. The detected singing voice and the volume in each beat are used in singing phrase segmentation, which is presented later.

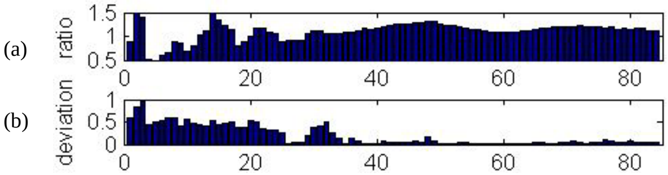


**Figure 14: Audio channel balance ratio and the standard deviation**

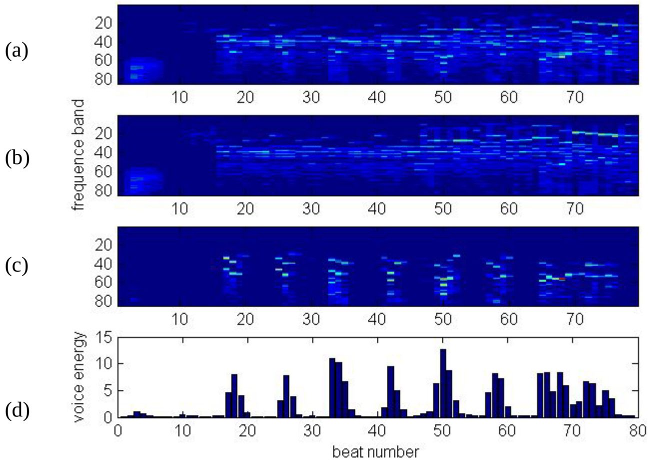


**Figure 15: Voice detection by pitch profile difference of the original singing channel and compensated accompaniment channel**

## 4.3 Lyrics Repetition Detection

Lyrics repetition provides important clues of the structure of songs, for example the lyrics of the choruses are usually same for the whole song and sometimes lyrics of the verses may also repeat, but at least two verses are different from each other. We propose a technique to detect the lyrics repetition based on the extracted lyrics text bitmaps. We do not do character recognition in this approach, thus this technique is independent of the language of the lyrics.

This technique is based on similarity matching of the lyrics text bitmaps, which are converted to a temporal sequence of characters as presented in section 2. The matching between two characters is done by computing the image difference of the two bitmaps. If the difference between a pair of pixel is less than a threshold (0.05), then the pair of pixels is marked as matched pixel. The pixel in one character bitmap without a corresponding pixel in the other bitmap is marked as unmatched pixel. Horizontal and vertical shifting of up to 2 pixels is used in the bitmap matching, and the one with maximal number of matched pixels is adopted. After computing the bitmap difference, the two characters is considered a match if the number of matched pixels takes a percentage higher than a threshold (75%) for both of the two bitmaps. The final bitmap matching result is one if there is match, and otherwise it is zero.

After matching all the character bitmaps in the character bitmap sequence, a self-similarity matrix can be derived. The repetition of lyrics is detected by looking for continuous lines of diagonal direction in the matrix. This can also be done by converting the matrix to character sequence-lag space, in which the repetition corresponds to horizontal lines. An example is shown in Figure 16. Figure (a) illustrates the self-similarity matrix of the lyrics bitmap sequence, where the x-axis and y-axis are the sequence index of the segmented characters. Figure 16 (b) shows the matrix in character sequence-lag space. And Figure 16 (c) illustrates the detected lines corresponding to repeated section of the lyrics text. This line detection technique is similar to [1].

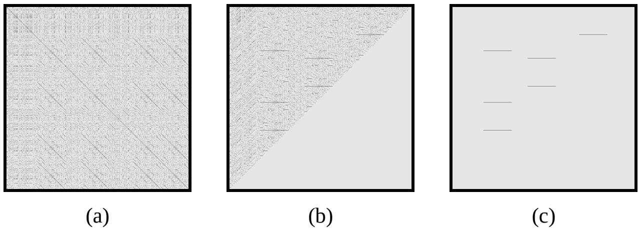


**Figure 16: Lyrics repetition detection by self-similarity matrix of character bitmap sequence**

## 4.4 Melody Repetition Detection

Melodies of a song repeat for both chorus and verse, and the verse and chorus normally have difference melodies. We adopted an approach that is similar to [1] for melody repetition detection. We computed the 12-dimention pitch-class-profile (PCP) vector for each beat interval of the audio signal in the original singing channel. This vector is obtained by summing the pitch profile energy of the same pitch classes in the pitch profile. The pitch class profile vector characterizes the energies of the 12 pitch classes in the audio. Then we compute the similarities between the PCP vectors by the correlation, which derives a self-similarity matrix for the whole music. The repeated sections are listed by locating continuous line segments with high similarity values in the diagonal directions in the matrix, which can also be done in the time-lag space. The detected line segments correspond to the repeated verses or choruses. An example is shown in Figure 17. The x-axis and y-axis in (a) are the time index of the beat windows. Figure 17(b) shows the matrix in time-lag space, and (c) shows the detected lines corresponding to sections of repeated melody.

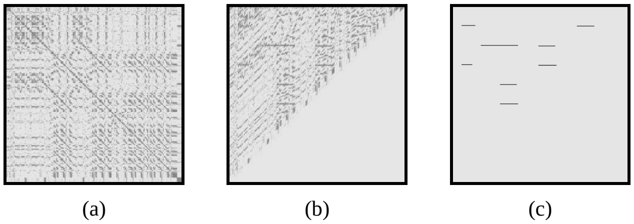


**Figure 17: Melody repetition detection by computing the pitch class profile self- similarity.**

## 4.5 Verse and Chorus Detection

We have obtained the lyrics repetition sections and the melody repetition sections. The repetition results are then combined by synchronizing the character sequence with the beat windows to detect and identify the verses and choruses. The sections with repeated lyrics and melody are detected as choruses, and the sections with repeated melody but not the lyrics are detected as verses. There are cases, in which the lyrics of verses repeat. Such cases can be distinguished by identifying at least one section with same melody but different lyrics.

There are also the cases that the lyrics repeat, but melody does not repeat, which is mainly due to the key changes of the subsequent choruses. These cases can be verified by doing a pitch shifted pitch-class-profile vector matching.

## 4.6 Singing Phrase Segmentation

A singer sings the lyrics of a song phrase by phrase, where each phrase may last several bars. We define a phrase as the words separated by breath inhaling in the singing. A phrase may correspond to a line of lyrics text, but it may not always be the case. Segmenting the singing phrases can help doing music retrieval by lyrics or melody.

We proposed a technique for singing phrase segmentation based on the lyrics text extraction and singing voice detection. From the lyrics text extraction and color change tracking, the time of lyrics text changes correspond to the time of singing voice, and a continuous period of time of no text color change corresponds to no singing voice. These time intervals can be accurately detected based on the text color change traces. The end of a phrase is identified if the non-vocal last longer than a threshold (2 beat time or 1.5 seconds). If the vocal interval lasts for longer than a threshold (8 beats), then it should be segmented according to the voice detection result: if a note last longer than 2 beat and the volume is getting lower compared with the previous beats, it can be considered a boundary between 2 phrases.

# 5. EXPERIMENTAL RESULTS

We have conducted experiments to evaluate the proposed techniques using a dataset of 20 karaoke titles of popular songs, which consists of 15 English songs, 3 Chinese songs and 2 Japanese songs. All the titles are encoded in MPEG1 NTSC format, i.e. video has 352x240 frame size and 29.97 frame rate and audio has 2 channels with 44.1 kHz sampling rate and 16 bits sample size. We have used the MPEG Developing Class Toolbox [15] to assist on the video analysis and manual annotation. The audio stream are extracted from the MPEG system stream and decoded to waveform for separate analysis. Time synchronization of video and audio is done by using the video frame rate and audio sampling rate.

We manually annotated the positions of the textboxes by the coordinates of the vertical and horizontal bounds and annotated the durations that lyrics text lines are presented in each textbox by the starting and ending frame numbers.

We used Adobe Audition software to help annotate the singing phrase boundaries. By listening to the original singing channel, the time of the end of each singing phrase is annotated by inserting a pulse click sound in a spare channel that are synchronized with the original sing channel. The singing phrase boundaries groundtruth is then obtained by locating the pulse in the spare channel. We annotated the singing phrases of 10 English songs in this experiment.

## 5.1 Lyrics Extraction

### *5.1.1 Textbox localization and text presence detection*

Our proposed technique can localized the textboxes with 100% accuracy for the 20 titles, where the location is considered correct if the differences between the detected coordinates and annotated coordinates are within 3 pixels.

Text presence detection in textboxes is evaluated by the precision and recall of the detection. The recall is 99.4% and the precision is 97.1%. The 0.6% missed text presence is due to the fade-in and fade-out effects of the lyrics, and such errors have no significant impact on the lyrics extraction results. The 2.9% false positives are due to the complex background scene in the background. However, these false positives cannot pass through the bitmap extraction process at later stages.

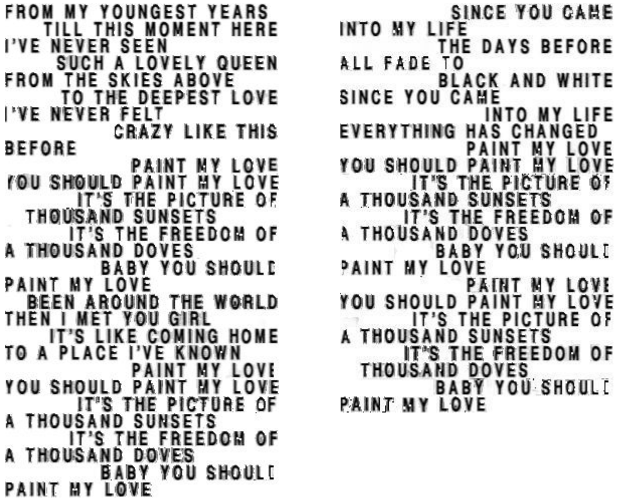
### *5.1.2 Text object segmentation*

We do not annotate the text object boundary groundtruth for evaluation. Instead, the segmentation result can be easily verified by seeing the contents of the textbox at the beginning and ending frame of a segmented text object and the beginning of the next immediate text object. It is considered a false negative if the beginning and ending frames of a text object contains different lyrics text lines. And it is considered a false positive if the ending frame of the current text object and the beginning frame of the next text object contain a same text line. The recall is 100%, i.e. all the text content changes are correctly detected. The precision of the segmentation is 93.5%, and the errors are all due to the significant scene changes of the background scene.

We also evaluated the performance of content change type (text or background) classification. This is also done by manually verify the contents before and after the detected change. No text changes have been misclassified as background change, and 91.2% of the background scene changes are correctly classified. 8.8% of the background scene changes are misclassified as text change. Thus there some over segmented text objects.

### *5.1.3 Text bitmap extraction and color change tracking*

As mentioned before, the over segmented text objects can be recovered by the text color tracking process. A case of over segmentation can be detected, if the color change trace of the current object last to the ending frame, the color change trace of the next text object starts at the beginning frame, and the two traces are temporally spatially continuous. In the 21 cases of text object over segmentation, 19 cases are successfully recovered. The remaining two cases are of short time intervals, where there is no text color change, so they can be eliminated without affecting the text extraction result.



FROM MY YOUNGEST YEARS
TILL THIS MOMENT HERE
I'VE NEVER SEEN
SUCH A LOVELY QUEEN
FROM THE SKIES ABOVE
TO THE DEEPEST LOVE
I'VE NEVER FELT
CRAZY LIKE THIS
BEFORE
PAINT MY LOVE
YOU SHOULD PAINT MY LOVE
IT'S THE PICTURE OF
THOUSAND SUNSETS
IT'S THE FREEDOM OF
A THOUSAND DOVES
BABY YOU SHOULD
PAINT MY LOVE
BEEN AROUND THE WORLD
THEN I MET YOU GIRL
IT'S LIKE COMING HOME
TO A PLACE I'VE KNOWN
PAINT MY LOVE
YOU SHOULD PAINT MY LOVE
IT'S THE PICTURE OF
A THOUSAND SUNSETS
IT'S THE FREEDOM OF
A THOUSAND DOVES
BABY YOU SHOULD
PAINT MY LOVE

SINCE YOU CAME
INTO MY LIFE
THE DAYS BEFORE
ALL FADE TO
BLACK AND WHITE
SINCE YOU CAME
INTO MY LIFE
EVERYTHING HAS CHANGED
PAINT MY LOVE
YOU SHOULD PAINT MY LOVE
IT'S THE PICTURE OF
A THOUSAND SUNSETS
IT'S THE FREEDOM OF
A THOUSAND DOVES
BABY YOU SHOULD
PAINT MY LOVE
PAINT MY LOVE
YOU SHOULD PAINT MY LOVE
IT'S THE PICTURE OF
A THOUSAND SUNSETS
IT'S THE FREEDOM OF
THOUSAND DOVES
BABY YOU SHOULD
PAINT MY LOVE

**Figure 18: Lyrics text extraction result for "paint my love"**

The text bitmap extraction is evaluated by the accuracy of the text characters that are successfully extracted. The extracted text bitmap is manually compared with the beginning frame of the text object. An error is considered for the whole text object, if at least one character is missed in the text line. Out of the 850 text objects, 22 text objects contain errors. So the accuracy is 97.4%. Some of the errors are caused by certain characters that cover small areas (like the Chinese character "one'), thus they are occasionally missed out in the spatial temporal filtering process. These isolated missed characters can hardly affect the result of music structure analysis, since the lyrics repetition is detected by comparing character sequences. Figure 18 shows the lyrics text bitmap images of the karaoke title "paint my love".

## 5.2 Verse and Chorus Detection

Out of the 20 songs, all the choruses are correctly located and identified, and 18 verses are correctly detected and identified. The two errors on verse detection are caused by the melody variation between the verses, and the detected verse is only the beginning part of the actual verse section. Table 1 illustrates the performance comparison of the proposed method with the previous method [6], using detection precision and recall. The result shows that the performance of our multimodal approach is better than [6], which is based only on audio content analysis.

**Table 1: Verse and chorus detection performance**

| | Verse | | Chorus | |
|---|---|---|---|---|
| | Precision | Recall | Precision | Recall |
| The proposed method | 100% | 93% | 100% | 100% |
| The previous method [6] | 86.85% | 79.58% | 88.68% | 81.36% |

## 5.3 Phrase Segmentation

The phrase segmentation result is evaluated by comparing with the annotated phrase boundaries. Out of the 356 annotated phrase boundaries of the 10 songs, the detection precision is 87%, and the recall is 92%. Many of the misses are due to very short break of the singing voice, and the duration of the gaps either from lyrics text or voice detection is too short to be captured. Some false positives are due to errors in singing voice detection.

## 6. CONCLUSION AND DISCUSSION

In this paper, we have presented a multimodality approach for content-based structure analysis of karaoke music. We proposed a novel technique for tracking and extracting lyrics text from the video frames. The extracted lyrics text bitmaps are used in detecting sections of repeated lyrics in the song. The choruses and verses of songs are located and identified based on the results of lyrics repetition and melody repetition detection. We also presented a technique to characterize the original singing voice based on the signals in the two audio channels, and proposed a method for segmenting the lyrics phrases based on the results of voice detection and lyrics text tracking. The presented work is a part of our efforts toward the development of a multimedia music retrieval system, which allows a user to retrieve and browse karaoke music by lyrics, melody, and summaries.

The lyrics text extraction method presented in this paper assumes the usual textbox configuration, i.e. two textboxes with text lines aligned to the left and the right respectively. However, the technique can be easily modified to automatically work for other configurations, such as one or three textboxes, and/or center aligned texts. In addition, the application of the lyrics text analysis technique is not limited to music structure analysis. It can also be used for automatic generation of visual summary of the music video, and lyrics text based retrieval.

In addition, some techniques presented in this paper are not limited to the analysis of Karaoke music. The text-box detection and text object segmentation methods can be used for text detection in other types of video, like caption or subtitles in TV programs. Furthermore, the overall multimodality approach for music structure analysis can also be applied to the analysis of generic music video, where the synchronization of text to music is at a coarser level and the lyrics texts do not have color changes.

Our future work consists of discovering the pitch structures of the music, such as key and key change detection of difference sections of a song, vocal melody pitch extraction, and etc. We would also increase the data set of system for experiments and evaluation.

## 7. REFERENCES

[1] Goto, M. A Chorus-Section Detecting Method for Musical Audio Signals. In *Proc. IEEE ICASSP*. 2003

[2] Bartsch, M. A., and Wakefield, G.H. To Catch a Chorus: Using Chroma-based Representations for Audio Thumbnailing. In *Proc. WASPA*. 2001.

[3] Chai, W., and Vercoe, B. Music Thumbnailing via Structural Analysis. In *Proc. ACM Multimedia 2003*, 223-226.

[4] Cooper, M., and Foote, J. Automatic Music Summarization via Similarity Analysis, In *Proc. ISMIR*, 2002.

[5] Lu, L., and Zhang, H. Automated Extraction of Music Snippets, In *Proc. ACM Multimedia. 2003*, 140-147.

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

[7] Shao, X., Xu, C., and Kankanhalli M. A New Approach to Automatic Music Video Summarization. *Proc. ICIP 2004.*

[8] Wang, Y., Kan, M.Y., New, T.L., Shenoy A., and Yin, J. lyrically: Automatic Synchronization of Acoustic Musical Signals and Textural Lyrics. In *Proc. ACM Multimedia 2004*.

[9] Lienhart, R. and Wernicke A., Localizing and Segmenting Text in Images and Videos. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 12, No. 4, April 2004.

[10] Lienhart, R. and Effelsberg, W., Automatic text segmentation and text recognition for video indexing, *Multimedia Syst.*, vol. 8, pp. 69–81, Jan. 2000

[11] Li, H, Doermann D., and Kia, O., Automatic text detection and tracking in digital video, *IEEE Trans. Image Processing*, vol. 9, pp. 147–156, Jan. 2000.

[12] Sato, T., Kanade, T., Hughes, E., Smith, M., and Satoh, S.-i Video OCR: Indexing digital news libraries by recognition of

superimposed caption, *Multimedia Syst.*, vol. 7, no. 5, pp. 385–395, 1999.

[13] Goto, M. An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds. *Journal of New Music Research*. June 2001, Vol.30, 159-171.

[14] Brown J.C. 1991. Calculation of a constant Q spectral transform. In *J. Acoust. Soc. Am.*, 89(1):425-434, 1991.

[15] Li, D. and Sethi, I.K., MDC: a software tool for developing MPEG applications, *IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 445-450, 1999.