CHORUS DETECTION IN SONGS OF POP MUSIC

Volker Mildner, Peter Klenner und Karl-Dirk Kammeyer

University Bremen
mildner@ant.uni-bremen.de

Abstract:

This contribution addresses the problem of chorus detection foraudio thumbnailing. We utilise the similarity matrix in order to locate within a pop song the position of its chorus (to be the audio thumbnail). Our focus lies on the problem of analysing the similarity matrix. We do so by filtering its elements including methods known from image processing and using the resulting data. The algorithm is designed to locate the beginning of a chorus as exact as possible in order to be able to extract a passage of thirty seconds from a song containing the chorus. Our simulation results show that the algorithm performs well on a variety of pop songs.

1 Introduction

When searching for a certain song - within a database or an archive offered via the internet - but knowing neither the exact name of the song nor the artist (these two details are usually the only information provided in text format in a file along with the raw audio data), a user might have to skip through an enormous number of files by listening to each of them, even if a pre-selection has been made before. During this so called *pre-listening*, it would be quite helpful not to listen to each song from its beginning but to the chorus instead, since this is the most recognizable part. Thus, it is highly desirable to have algorithms at hand which provide information about the chorus' position within a song.

To choose an audio-thumbnail it is at the moment a common procedure in the recording industry to extract a passage with the length of thirty seconds from a song which reaches from second number 50 to second number 80, hoping that the chorus is contained within such a passage, although second number 50 is not necessarily the beginning of the chorus.

Our approach is to detect the chorus' beginning in order to use this as the starting point of a thirty second passage as the audio thumbnail. Usually the chorus of a pop song is not longer than 20 seconds (at most), so that in case of a reliable detection of the chorus' beginning our chosen audio thumbnail should contain the chorus.

The problem of audio thumbnailing has been addressed previously: Logan and Chu [1] tried the application of Hidden Markov Models as well as clustering for this purpose. The principle of the *similarity matrix* for visualizing the structure of audio-data has been introduced by Foote [2,3] and its application to audio thumbnailing was presented by Bartsch and Wakefield [4] and has also been investigated by Van Steelant *et al* [5].

This paper is organized as follows: the general idea of extracting features from an audio signal and visualizing structure of audio data by a similarity matrix is reviewed in Section 2. In Section 3 we give an overview of our algorithm and a closer description of certain algorithm steps. Results of chorus detection for different pop songs are presented in Section 4. Conclusions and outlook are given in Section 5.

2 Feature Extraction and Similarity Matrix

2.1 Principle of the Similarity-Matrix

The extraction of features from an audio signal and generating the similarity matrix is depicted in Figure 1.

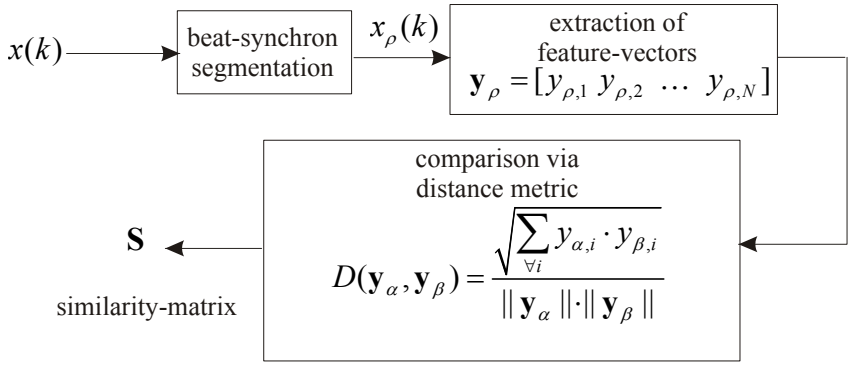

Figure 1 – Feature extraction and similarity-matrix

The discrete audio signal $x(k)$ is separated into beat-synchronous frames $x_\rho(k)$ (where ρ denotes the frame index) of a length smaller than 100 ms and for each frame a feature-vector $\mathbf{y}_\rho$ of size N as a reduced spectral representation is extracted (also see Section 2.2). The feature vectors of all frames are compared by the cosine distance D [2] and the results are placed in the upper triangular part of the *similarity matrix* $\mathbf{S}$. The cosine distance is 0 for identical feature-vectors and becomes greater for less similar feature-vectors. The general structure of such a matrix is shown in Figure 2.

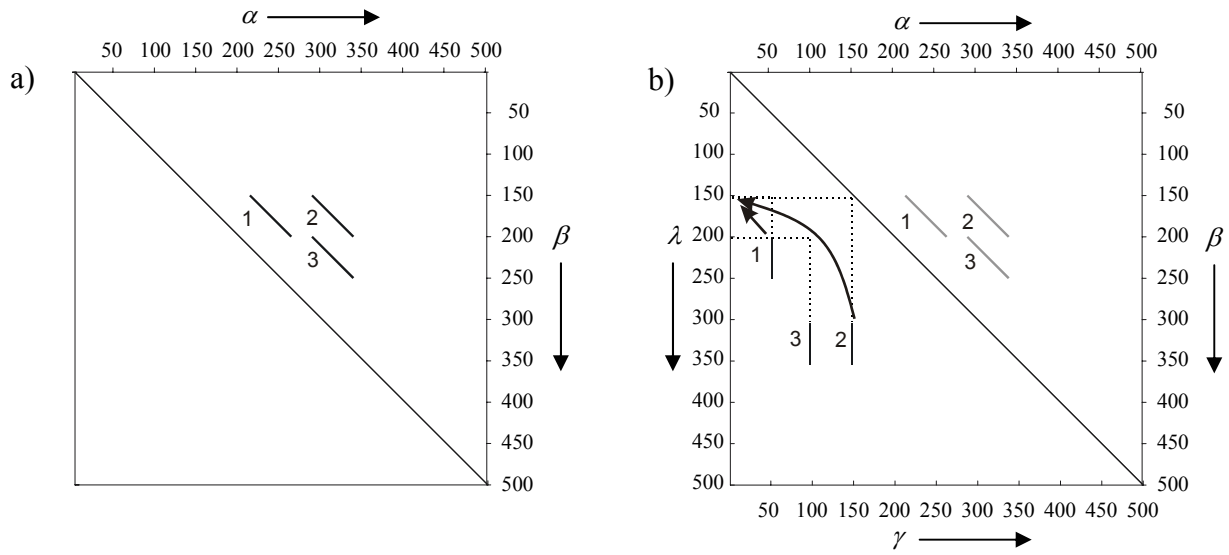

Figure 2 – a) Similarity matrix with diagonals indicating multiple repetition of a passage b) Matrix turned; matching of two lines for the same starting position of a preceding passage

First, consider Figure 2.a : small values of D are displayed as dark pixels, while α and β are the indices of the compared frames. Since the feature-vector of each frame is identical to itself we see a black line along the main diagonal. Assume that the exact same chorus is played three times throughout the song: From frame 150 to 199, frame 200 to 249 and from frame 300 to 349. The similarity of the first two passages results in the dark diagonal line with index '1'. The third appearance of the chorus from frame 300 to 349 results in two further lines: one line due to the similarity to frame 150 to 199 (index '2') and another line due to the

similarity to frame 200 to 249 (index '3'). The Matrix displayed in Figure 2.b also contains a lower triangular matrix: The upper triangular part has been 'rotated', such that the main diagonal forms the first column and higher diagonals the following columns. Now, the new frame indices λ and γ can be interpreted as follows: the perpendicular line with index '1' denotes that a passage played from frame λ=200 … 249 has already appeared γ=50 frames *earlier* in the song. Considering the lines with indices '1' and '2' we see that they both refer to the same preceding passage which started at frame λ=150 (as indicated by the dotted lines and the arrows). This idea of 'multiple indication' will be exploited later on.

2.2 The Feature

Although numerous authors [1,2,3,4] have suggested the use of Mel-Frequency-Cepstral-Coefficients for feature-extraction well known from speech recognition, we make use of another feature. It is based on the *critical band scale rate* [6], which has been successfully applied to the purposes of noise reduction [7] and audio coding [8].

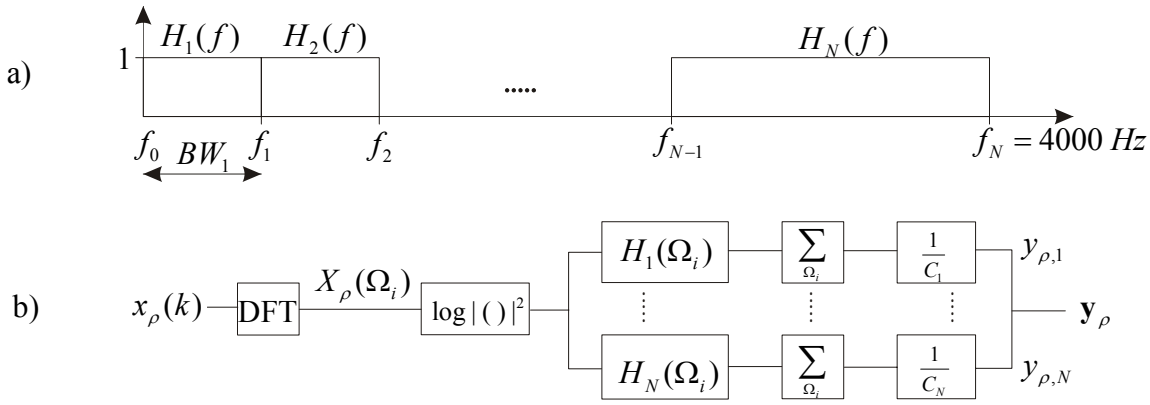

Figure 3 – a) Subbands defined by bark-band-width *BW* b) Block diagram for feature-extraction

The frequency axis is split into subbands (see Figure 3.a) of *bark-band-width BW* which is 100 Hz for the first subband BW_1 and gets wider for higher frequencies [6]. Using the subbands as bandpass-filters, we extract the feature as depicted in Figure 3.b : for each frame we compute the periodogramm as an approximation of the power spectral density, sum the periodogramm within each of the *N* subbands and divide this sum by the number of discrete frequency bins $C_{1\ldots N}$ belonging to that subband. The values from each subband $y_{\rho,1\ldots N}$ are concatenated to form the feature vector $\mathbf{y}_\rho = [y_{\rho,1} \ \ldots \ y_{\rho,N}]$.

3 The Algorithm

In this section we outline our algorithm and explain certain steps more detailed. As a general overview a block diagram along with results from simulation steps for one song is shown in Figure 4 (*'We will rock you'* by Queen).

3.1 Matrix generation

As already outline in Section 2.1 we segment the audio signal into frames, extract a feature-vector for each frame and place the result of comparing all feature-vectors by the cosine distances in the upper triangular part of the similarity-matrix (Figure 4.b). Since the values are within a range $[0 \ldots 1]$ it is obvious that we have to determine a threshold for display purposes in order to make expected diagonals visible.

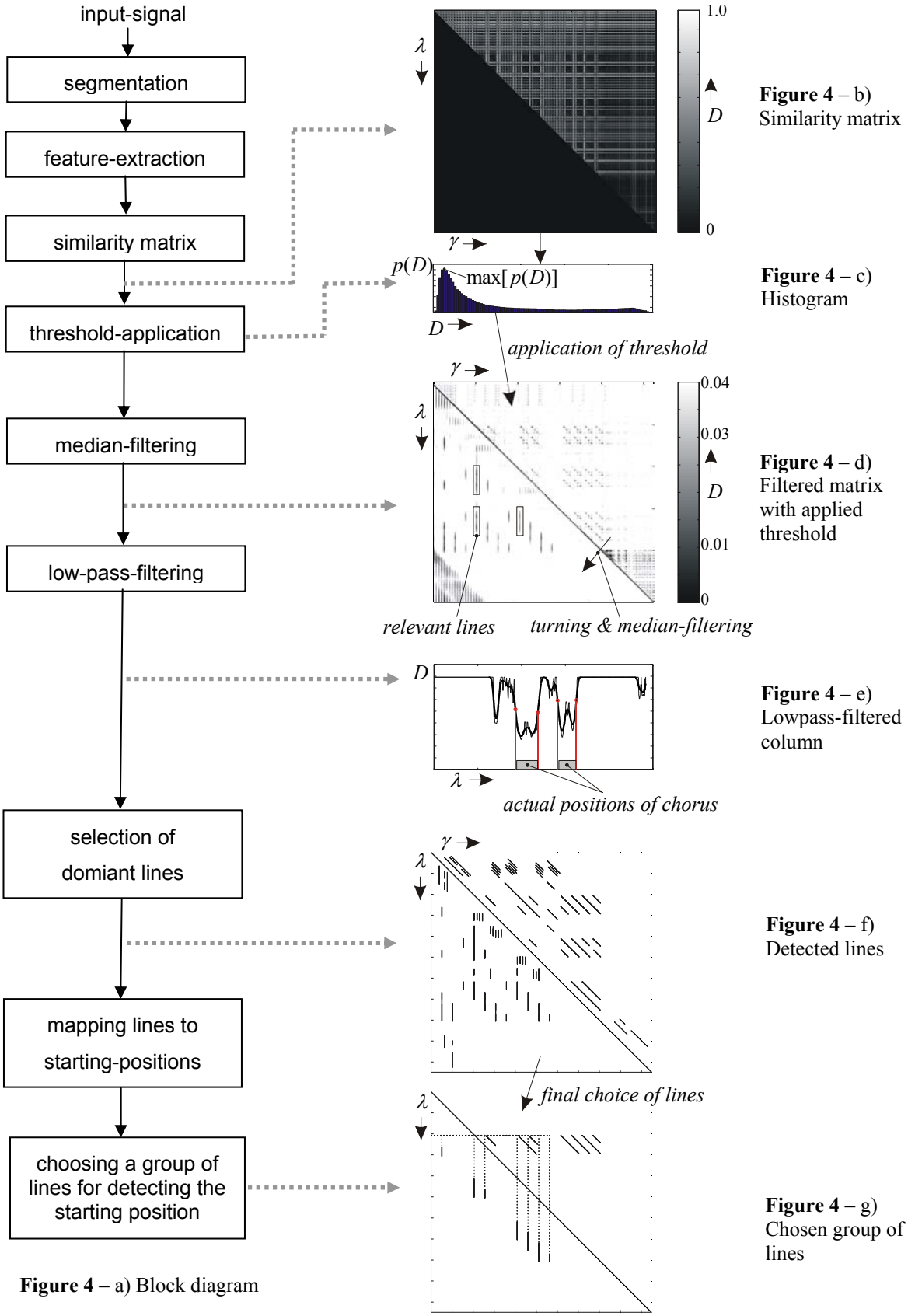

Figure 4 – a) Block diagram

Figure 4 – b) Similarity matrix

Figure 4 – c) Histogram

Figure 4 – d) Filtered matrix with applied threshold

Figure 4 – e) Lowpass-filtered column

Figure 4 – f) Detected lines

Figure 4 – g) Chosen group of lines

Figure 4 – Overview of algorithm: a) Block diagram b)-g) Algorithm steps and simulation examples

3.2 Threshold determination

For the values of the upper triangular matrix we generate the histogram and choose that value of *D* as the threshold for which the histogram reaches its maximums (Figure 4.c). All values of the matrix exceeding the value of the threshold are set to that threshold. This principle of so called 'histogram stretching' is known from image processing for the purpose of improving a picture's contrast.

The result can be seen in the upper triangular matrix depicted in Figure 4.d where the maximum value of the display range has been set to the threshold.

3.3 Turning and median-filtering

Although the application of the threshold has revealed some slight diagonal lines they are still not easy to distinguish from the rest of the matrix. Thus, we filter each diagonal with a median filter and turn the matrix as described in Section 2.1. As an example the result is given in Figure 5.

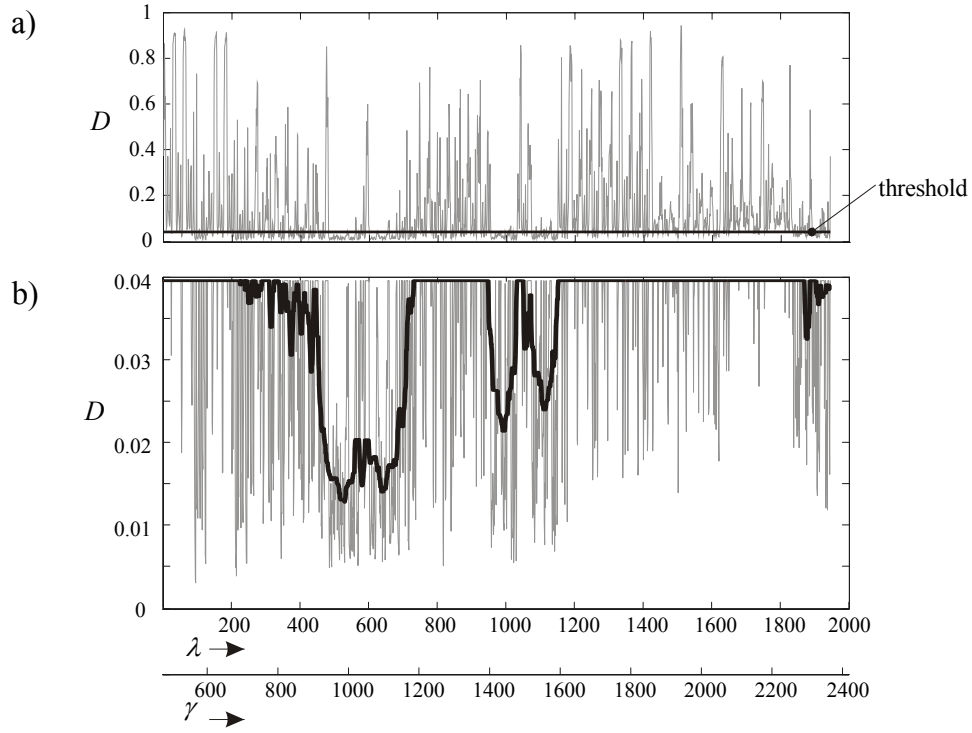

Figure 5 – a) Data values of a diagonal (grey); black line indicating the threshold
b) Diagonal after thresholding (grey) and median filtering (black)

This filter is often used in image processing to remove impulse-like disturbances. As the length of the filter we choose a size equivalent to 4 seconds and now we are able to see some clear lines in the lower triangular matrix in Figure 4.d . The chosen length of the filter turned out to work well during simulations. Shorter filters of 1 second would tend not to average strong enough over short variations of the data values, thus leaving a lot of short lines 'in the picture'. Much longer filters of 10 seconds or more would almost remove the lines we want to detect. Examples of such relevant lines are highlighted in Figure 4.d by grey boxes.

The authors of the first approach of audio thumbnailing decided to choose that position of time in the song which is indicated by the point of greatest similarity (here: smallest value) in the matrix hoping that this was within a line caused by a chorus' repetition [4]. They would extract a sequence of fixed length around this position of time, where the start of the chorus would not necessarily be the start of the audio-thumbnail. In contrary to this, we try to detect

the apparent lines and their beginnings in order to yield the position of time of the chorus' beginning. We do so to use this as the starting point of a thirty second passage as the audio thumbnail. We furthermore exploit the fact, that numerous lines indicate towards the same time position (as already described in Section 2.1).

3.4 Lowpass-filtering

The data values along a single column are lowpass-filtered since their characteristics are not yet suitable for a reliable detection of lines. For the column indicated in Figure 4.d the values before and after lowpass-filtering are plotted in Figure 4.e . From the lowpass-filtered curve its points of inflection having at least a minimal distance to each other are now used as the beginning- and end-points of relevant lines. They are indicated by the perpendicular dotted lines. The actual position of the chorus is plotted as grey bars on the bottom for comparison.

3.5 Selection of dominant lines and final choice of starting position for the chorus

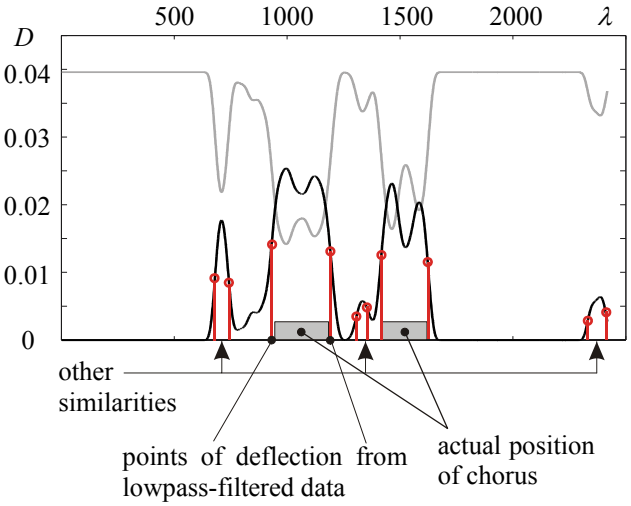

Figure 6 – Data values of the lowpass-filtered curve (grey); curve after inverting and integration (black); points of deflection for beginning and end positions of lines (red)

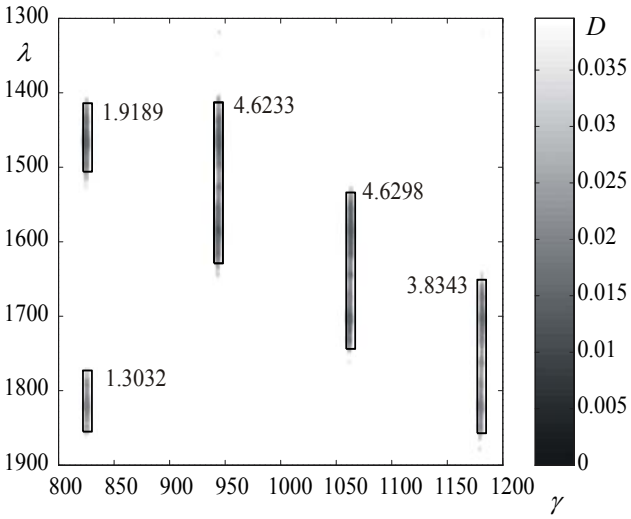

Figure 7 – Sector of the similarity matrix: detected lines along with the value indicating their strength

It is now necessary to distinguish between the most dominant lines. To obtain a value indicating the 'strength' of a line, we set the values of 0 to the threshold and vice versa (,inverting') and integrate them within a window of a length equivalent to 5 seconds. The values given by the integrated curve are summed between the points of deflection given from the lowpass-filtered curve. This sum is used as a value to indicate a lines strength.

In Figure 6 exemplary curves can be seen.

For a sector of the similarity selected lines along with their values indicating their strength are shown in Figure 7.

We choose that frame λ as the starting position of our audio thumbnail for which the sum of the lines' strength values pointing towards it is the highest.

We have to consider that using the points of deflection for starting positions of the lines is not too reliable as a method. Thus, some lines might point to a position just a few frames λ away from another line. Therefore, we have to average the sum for each frame within a window of two seconds before making our final decision

In Figure 4.g the group of lines indicating towards the same frame λ are depicted.

4 Results

Our algorithm was tested on 30 different pop songs, from Beatles to Billy Idol (an exact list of the titles can be seen on [9], '*We will rock you'* is number 22). Pop songs were chosen since they usually have a clear and simple structure of elements such as verse and chorus, where the chorus is repeated several times within the song and the verses are slightly different to each other because of the different lyrics sung by the vocalist.

The deviation of the chorus' beginning from the positions detected by the different methods are displayed in Figure 8.

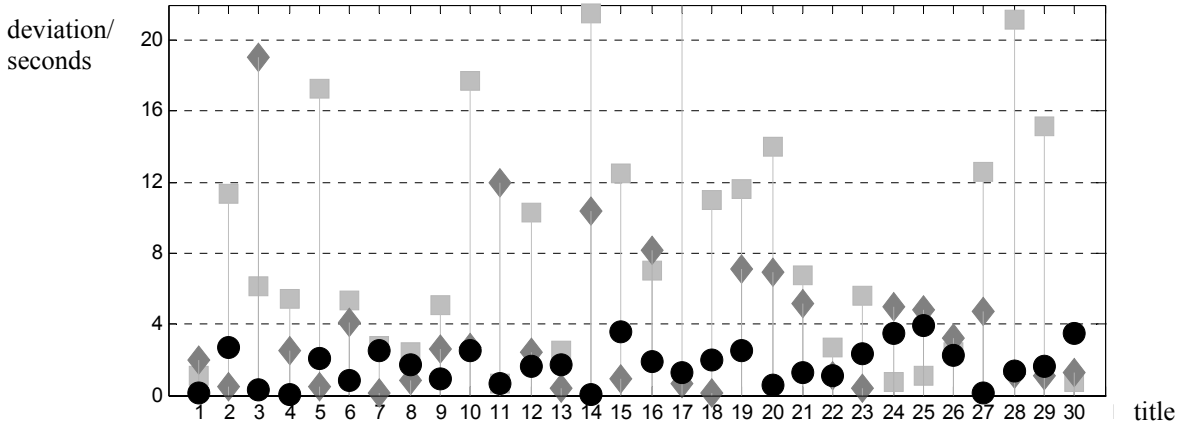

Figure 8 – Deviations of detection results from different methods: recording industry's method (light grey squares), method of greatest similarity (dark grey diamonds), our algorithm (black circles)

The light grey squares show the results for the method currently used in the recording industry. As expected there is a great deviation for numerous titles, although for 10 titles (one third of all) the deviation is smaller than 4 seconds.

Depicted by the dark grey diamonds are the deviations if we would use the point of the greatest similarity (smallest value) within the similarity matrix as the beginning of the chorus. For 20 titles this leads to reasonable results with deviations smaller than 4 seconds, but for 10 titles there are larger deviations up to as much as 19 seconds, far away from the actual chorus. The reason for such large deviations is that throughout a song two rather short passages are more similar to each other than longer passages such as the chorus. Such shorter passages are often instrumental parts with a length of just one or two bars played in verses.

Indicated by the black circles are the deviations obtained by our algorithm. For all songs we yield a deviation smaller than 4 seconds. The deviations are mostly caused by the fact that not only the chorus and its repetitions are extremely similar to each other but also the last bars of each verse preceding the chorus (sometimes also called 'the bridge' by musicians).

If we were to choose the actual starting position of the audio-thumbnail 5 seconds ahead of the detected position in order to allow for a fade-in sequence, we would have a reasonable chance to include the whole chorus in an audio-thumbnail of 30 seconds since the length of a chorus is usually smaller than 20 seconds.

5 Conclusions

This paper addressed the problem of chorus detection within pop songs for the purpose of audio thumbnailing.

In section 1 we have given a motivation for audio thumbnailing which is the application of pre-listening for a user that is searching for a song in a database.

The principle of extracting features from an audio signal and visualizing the structure of music via the similarity matrix was reviewed in section 2. It was explained why reliable detection of lines in the matrix helps to detect the beginning of a song's chorus.

Our algorithm for analysing the similarity matrix was presented in section 3. A general overview was given along with exemplary simulation results. We explained certain simulation steps more in detail and showed that our approach of detecting lines within the similarity matrix gives valuable information.

Results for 30 different pop songs were given in Section 4 to compare the presented algorithm with the methods from the recording industry and the first approach of audio thumbnailing. The results show that our algorithm performs well, but still gives some deviations when trying to detect the beginning of a chorus. A higher accuracy is still desirable at this point.

Literature

[1] B. Logan and S. Chu, "Music Summarization Using Key Phrases", *ICASSP '00 Proceeding*, 2000 , Volume: 2 , 2000, Page(s): II749 -II752 vol.2

[2] J. Foote, "Automatic Audio Segmentation Using a Measure of Audio Novelty", *IEEE Proceedings*, *ICME*, 1999, vol. I, pp. 452-455.

[3] J. Foote, "Visualizing Music and Audio Using Self-Similarity",*Proceedings of ACM Multimedia '99*, Orlando, Florida, November 1999, pp. 77-80.

[4] M.A. Bartsch and G.H. Wakefield, "To Catch a Chorus: Using Chroma-Based Representations for Audio Thumbnailing", *IEEE WAPAA 2001*, New Paltz, NewYork, 2001.

[5] D. Van Steelant *et al*., "Discovering Structure and Repetition in Musical Audio", *Proceedings of Eurofuse Workshop,* Varenna, Italy, 2002.

[6] H. Fastl and E. Zwicker, "Psychoacoustics: Facts and Models", Springer, New York, 1990.

[7] S. Gustafsson, "Enhancement of Audio Signals by Combined Acoustic Echo Cancellation and Noise Reduction", PhD thesis, IND, University of Technology Aachen, June 1999.

[8] ISO/IEC, "International Standard 11172-3:1993, Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to 1.5 Mbit/s – Part3, Audio", 1993

[9] Web-page of the Project: www.ant.uni-bremen.de/research/audio/index.html