CHORUS DETECTION WITH COMBINED USE OF MFCC AND CHROMA FEATURES AND IMAGE PROCESSING FILTERS

Antti Eronen
Nokia Research Center
Tampere, Finland
Antti.Eronen@nokia.com

ABSTRACT

A computationally efficient method for detecting a chorus section in popular and rock music is presented. The method utilizes a distance matrix representation that is obtained by summing two separate distance matrices calculated using the mel-frequency cepstral coefficient and pitch chroma features. The benefit of computing two separate distance matrices is that different enhancement operations can be applied on each. An enhancement operation is found beneficial only for the chroma distance matrix. This is followed by detection of the off-diagonal segments of small distance from the distance matrix. From the detected segments, an initial chorus section is selected using a scoring mechanism utilizing several heuristics, and subjected to further processing. This further processing involves using image processing filters in a neighborhood of the distance matrix surrounding the initial chorus section. The final position and length of the chorus is selected based on the filtering results. On a database of 206 popular & rock music pieces an average F-measure of 86% is obtained. It takes about ten seconds to process a song with an average duration of three to four minutes on a Windows XP computer with a 2.8 GHz Intel Xeon processor.

1. INTRODUCTION

Music thumbnailing refers to the extraction of a characteristic, representative excerpt from a music file. Often the chorus or refrain is the most representative and "catchiest" part of a song. A basic application is to use this excerpt for previewing a music track. This is very useful if the user wishes to quickly get an impression of the content of a playlist, for example, or quickly browse the songs in an unknown album. In addition, the chorus part of a song would often make a good ring tone for a mobile phone, and automatic analysis of the chorus section would thus facilitate extraction of ring tone sections from music files.

Western popular music is well suited for automatic thumbnailing as it often consists of distinguishable sections, such as intro, verse, chorus, bridge, and outro. For example, the structure of a song may be intro, verse, chorus, verse, chorus, chorus. Some songs do not have as clear verse-chorus structure but there still are separate sections, such as section A and section B that repeat. In this case the most often repeating and energetic section is likely to contain the most recognizable part of the song.

Peeters et al. ([1]) divide the methods for chorus detection and music structure analysis into two main categories: the "state approach" which is based on clustering feature vectors to states having distinctive statistics, and the "sequence approach" which is based on computing a self-similarity matrix for the signal. One of the first examples of the state approach was that of Logan and Chu [2]. Recently, e.g. Levy et al. [3] and Rhodes et al. [4] have studied this approach. Similarity-matrix based approaches include the ones by Wellhausen & Crysandt [5] and Cooper & Foote [6]. Bartsch & Wakefield [7] and Goto [8] operated on an equivalent time-lag triangle representation. There are also methods utilizing many different cues, including e.g. segmentation into vocal / nonvocal sections, such as [9], or methods that iteratively try to find an optimal segmentation [10].

Here we present a method for detecting the chorus or some other often repeating and representative section from music files. The method is based on the self-similarity (distance) representation. The goal was to device a computationally efficient method that still would produce high quality music thumbnails for practical applications. Thus, iterative methods based on feature clustering or computationally intensive optimization steps could not be used. The following summarizes the novel aspects of the proposed method:

The self-distance matrix (SDM) used in the system is obtained by summing two distance matrices calculated using MFCC and chroma features. This improves the performance compared to the case when either of the features would be used alone. Although the MFCC features are sensitive to changing instrumentation between the occurrences of the chorus, the fact that the instrumentation and expression during the chorus is often different than in other parts of the song seems to overweigh this, at least with our pop & rock dominated data. The benefit of the proposed distance-matrix summing approach instead of merely concatenating the features into one, longer vector is that different enhancement operations can be applied for each matrix.

An initial chorus section is obtained from the repetitions detected from the SDM by utilizing a novel heuristic scoring scheme. The heuristics consider aspects such as the position of a repetition in the self-distance matrix (SDM), the adjustment of a repetition in relation to other repetitions in the SDM, average energy and average distance in the SDM during the repetition, and number of times the repetition occurs in the musical data.

The system performs the chorus determination in two steps: first a preliminary candidate is found for the chorus section, and then its final location and duration is determined by filtering with a set of image processing filters, selecting the final chorus position and duration according to the filter which gives the best fit.

Evaluations are presented on a database of 206 popular and rock music pieces. The method is demonstrated to provide sufficient accuracy for practical applications while being computationally efficient.

2. METHOD

2.1. Overview

Figure 1 shows an overview of the proposed method, which consists of the following steps. First the beats of the music signal are detected. Then, beat synchronous mel-frequency cepstral coefficient (MFCC) and pitch chroma features are calculated. This results in a sequence of MFCC and chroma feature vectors. Next, two self-distance matrices (SDM) are calculated, one for the MFCC features and one for the chroma features. Each item in the SDM represents the distance of feature vector at beat *i* to a feature vector at beat *j*. In the distance matrix representation, choruses or other repeating sections are shown as diagonal lines of small distance. The diagonal lines of the chroma distance matrix are then enhanced. Next we obtain a summed distance matrix by summing the chroma and MFCC distance matrices. This is followed by binarization of the summed distance matrix, which attempts to detect the diagonal regions of small distance (or high similarity). From the detected diagonal segments, the most likely chorus section candidate (diagonal stripe) is selected, and subjected to further processing. This further processing involves using image processing filters in a neighbourhood of the similarity matrix which surrounds the most likely chorus candidate. The final position and length of the chorus is selected based on the image processing results.

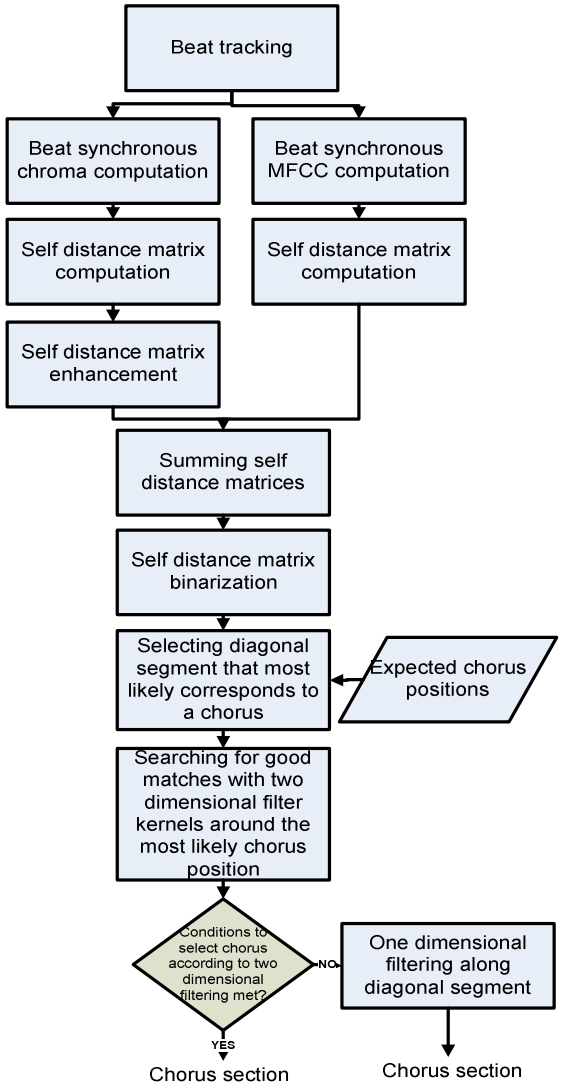

Figure 1: *Overview of the proposed method.*

2.2. Beat tracking

The feature extraction step begins by finding the beats in the acoustic music signal. We utilize the efficient beat tracking method described in [11] to produce an initial set of beat times and an accent signal $v(n)$. The accent signal measures the change in the spectrum of the signal and exhibits peaks at onset locations. An additional, non-causal postprocessing step was implemented to prevent the beat interval from changing significantly from one frame to another, which might cause problems with the beat synchronous self-distance matrices. The postprocessing is performed with a dynamic programming method described by Ellis [12]. The dynamic programming step takes as input the accent signal and median beat period produced by the method described in [11], performs smoothing of the accent signal with a Gaussian window, and then finds the optimal sequence of beats through the smoothed accent signal. The method iterates through each sample of the smoothed accent signal, and finds the best previous beat time for each time sample. The selection is affected by the strength of the accent signal at the previous beat position, and the difference to the ideal beat interval. The indices of best previous beats are stored for each time sample, and in the end the single best sequence is obtained by backtracking through the previous beat records. For more details see [12].

2.3. Feature calculation

Next, beat synchronous MFCC and chroma features are calculated. Analysis frames are synchronized to start at a beat time and end before the next beat time, and one feature vector for each beat is obtained as the average of feature values during that beat. Beat synchronous frame segmentation has been used earlier e.g. in [7]. It has two main advantages: it makes the system insensitive to tempo changes between different chorus performances, and significantly reduces the size of the self-distance matrices and thus computational load. Prior to the analysis, the input signal is downsampled to 22050 kHz sampling rate.

The MFCC features are calculated in 30 ms hamming windowed frames during each beat, and the average of 12 MFCC features (ignoring the zeroth coefficient) for each beat is stored. We use 36 frequency bands spaced evenly on the mel-frequency scale, and the filters span the frequency range from 30Hz to the nyquist frequency. Chroma features are calculated in longer, 186 ms frames to get a sufficient frequency resolution for the lower notes. In our implementation, each bin of the discrete Fourier transform is mapped to exactly one of the twelve pitch classes C, C#, D, D#, E, F, F#, G, G#, A, A#, B, with no overlap. The energy is calculated from a range of six octaves from C3 to B8 and summed to the corresponding pitch classes. The chroma vectors are normalized by dividing each vector by its maximum value.

After the analysis, each inter-beat interval is represented with a MFCC vector and chroma vector, both of which are 12-dimensional.

2.4. Distance matrix calculation

The next step is to calculate the self-distance matrix (SDM) for the signal. Each entry $D(i, j)$ in the distance matrix indicates the distance of the music signal at time i to itself at time j. As we are using beat synchronous features, time is measured in beat units. Two distance matrices are used, one for the MFCC features and one for the chroma features. The entry $D_{mfcc}(i, j)$ of the MFCC distance matrix is calculated as the Euclidean distance of MFCC vectors of beats i and j. Correspondingly, in the chroma distance matrix $D_{chroma}(i, j)$ each entry corresponds to the Euclidean distance of the chroma vectors of beats i and j. Figures 2 and 3 show examples of a chroma and MFCC distance matrices, respectively. As the Euclidean distance is symmetric, the distance matrix will also be symmetric. Thus, the following operations consider only the lower triangular part of the distance matrix.

Alternatives to calculating two different distance matrices would be to concatenate the features before calculating the distances, or combine the features in the distance calculation step. The benefit of keeping the distance matrices separate is that different enhancement operations can be applied to the chroma and MFCC matrices. Based on our experiments, it seems beneficial to apply an enhancement only for the chroma distance matrix and not for the MFCC distance matrix. When long chords or notes are played during several adjacent beats, the chroma distance matrix will exhibit a square area of small distance values. An enhancement operation similar to the one described in [8] was found to be beneficial in removing these. The MFCC distance matrix does not exhibit similar areas as the MFCC features are insensitive to pitch information, so this would explain the MFCC distance matrix does not benefit from the enhancement. Moreover, summing the distance matrices first and then enhancing the summed matrix did not perform as well as enhancing the chroma matrix only and then summing with the MFCC matrix. The next section describes the used enhancement and SDM summing steps.

2.5. Enhancing and summing the distance matrices

Ideally, the distance matrix should contain diagonal stripes of small distance values at positions corresponding to repetitions of the chorus or refrain section. However, due to variations in the performance of the chorus at different times (articulation, improvisation, changing instrumentation), the diagonal stripes are often not very well pronounced. In addition, there may be additional small distance regions which do not correspond to chorus sections. To make diagonal segments of small distance values more pronounced in the distance matrix, an enhancement method similar to the one presented in [8] is utilized.

The chroma distance matrix $D_{chroma}(i, j)$ is processed with a 5 by 5 kernel. For each point (i, j) in the chroma distance matrix, the kernel is centred to the point (i, j). Six directional local mean values are calculated along the upper-left, lower-right, right, left, upper, and lower dimensions of the kernel. If either of the means along the diagonal is the minimum of the local mean values, the point (i, j) in the distance matrix is emphasized by adding the minimum value. If some of the mean values along the horizontal or vertical directions is the smallest, it is assumed that the value at (i, j) is noisy and it is suppressed by adding the largest of the local mean values. After the enhancement the diagonal lines corresponding to repeating sections are more pronounced.

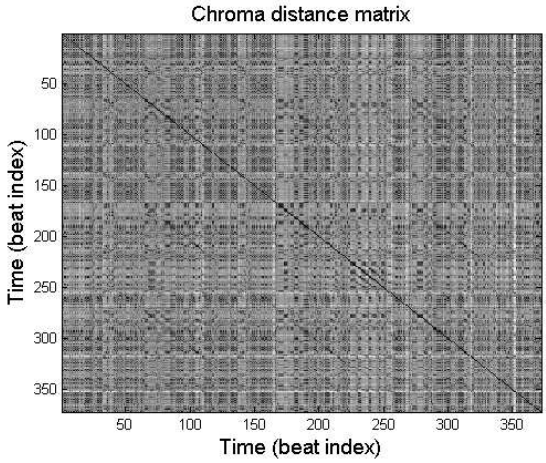

Figure 2: *The chroma distance matrix $D_{chroma}(i, j)$ of the song "Like a virgin" by Madonna.*

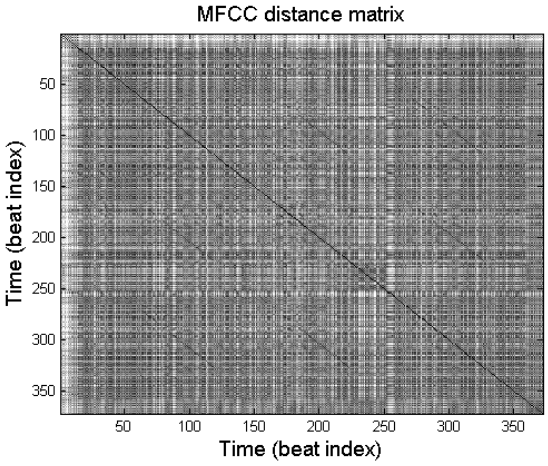

Figure 3: *The MFCC (timbre) distance matrix $D_{mfcc}(i, j)$ of the song "Like a virgin" by Madonna.*

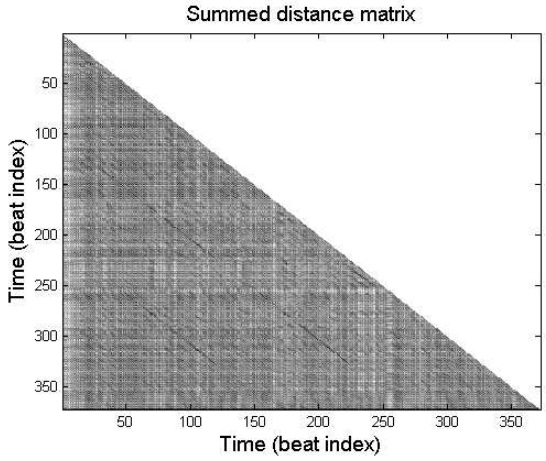

Figure 4. *The final distance matrix $D(i, j)$ of the song "Like a virgin" by Madonna obtained after summing the enhanced chroma distance matrix and MFCC distance matrix.*

After the enhancement step the chroma and MFCC distance matrices are summed. This gives the final distance matrix D, where the entries $D(i,j) = \widetilde{D}_{chroma}(i,j) + D_{mfcc}(i,j)$, where $\widetilde{D}_{chroma}$ is the chroma distance matrix after the above described enhancement operation. Figure 4 shows the summed distance matrix for Madonna's "Like a virgin". Weighted summation was also attempted for the different matrices with certain weight combinations, but equal weights (i.e. no weighting) seem to perform well. A slightly related approach to our distance matrix summing was presented by Marolt [13]. He constructed several beat synchronous melodic representations by comparing excerpts of different length, and then combined the representations by pointwise multiplication. This was reported to help in reducing noise in the self-similarity representation.

2.6. Detecting repetitions from the self-distance matrix

The following step attempts to detect which parts of the distance matrix correspond to a repetitive segment and which do not. The binarization method used here is similar to the one presented by Goto in [8], except that we operate on the low-triangular part of a distance matrix whereas Goto operated on the time-lag triangle. In addition, the filtering operations are simplified here and the threshold selection operations differ slightly.

When a sum is calculated along a diagonal segment of the distance matrix, a smaller value indicates a larger likelihood that the particular diagonal contains one or more line segments with small similarity values. A sum is calculated along the low-left diagonals k of the distance matrix, giving the values

$$F(k) = \frac{1}{M-k}\sum_{c=1}^{M-k} D(c+k,c)\ ,\ k = 1,...,M-1 \qquad (1)$$

where M is the number of beats in the song. Thus, $F(1)$ corresponds to the first diagonal below the main, $F(2)$ to the second below the main diagonal, and so on. The values of k corresponding to the smallest values of $F(k)$ indicate diagonals which are likely to have repetitions in them. With Eq. 1 there exists a possibility that some small-distance values are masked by high distance values that happen to locate at the same diagonal. Thus, it might be worth studying whether special methods to remove the effect of high-distance values would improve the performance. However, this was left for future research as the simple summing seems to work well.

A certain number of diagonals corresponding to minima in $F(k)$ are then selected. Before looking for minima in $F(k)$, it is "detrended" to remove cumulative noise from it. This is done by calculating a lowpass filtered version of $F(k)$, using a FIR lowpass filter with 50 taps, the value of each coefficient being 1/50. The lowpass filtered version of $F(k)$ is subtracted from $F(k)$.

The minima correspond to zero-crossings in the differential of $F(k)$. The smoothed differential of $F(k)$ is calculated by filtering $F(k)$ with an FIR filter having the coefficients $b_1(i) = K\text{-}i$, $i = 0, \ldots, 2K$, with $K = 1$. The minima candidates are obtained by finding the points where the smoothed differential of $F(k)$ changes its sign from negative to positive. The values of the minima are dichotomized into two classes with the Otsu method presented in [14], and the values smaller than the threshold are selected. We observed that sometimes it may happen that only a few negative peaks are selected using this threshold. This would mean that the following binarization would examine only a few diagonals of the distance matrix, increasing the possibility that some essential diagonal stripes are left unnoticed. To overcome this, we raise the threshold gradually until at least 10 minima (and thus diagonals) are selected. The subset of indices selected from all the diagonal indices $k \in [1, M-1]$ to search for line segments is denoted by Y.

The diagonals of the SDM selected for the line segment search are denoted by

$$g_y(c) = D(c+y,c)\ ,\ c = 1,...,M-y \qquad (2)$$

where $y \in Y$. The diagonals $g_y(c)$ of the distance matrix are smoothed by filtering with a FIR with coefficients $b_2(i) = ¼$, $i = 1, \ldots, 4$. Goto ([8]) performed another threshold selection with the Otsu method ([14]) to select a threshold to be used for detecting the line segments from the diagonals. However, we found it better to define a threshold such that 20% of the values of the smoothed diagonals $\widetilde{g}_y(c)$ are left below it, and thus 20% of values are set to correspond to diagonal repetitive segments. This threshold is obtained in a straightforward manner by concatenating all the values of $\widetilde{g}_y(c)$, $c = 1,...,M-y$ and $y \in Y$ into a long vector, sorting the vector, and selecting the value such that 20 % of the values are smaller. Points where $\widetilde{g}_y(c)$ exceeds the threshold are then set to one, others are set to zero. This gives the binarized distance matrix.

Next the binarized matrix is enhanced, such that diagonal segments where most values are ones (i.e. detected small distance segments) are enhanced to be all ones under certain conditions. This is done in order to remove gaps of few beats in such diagonal segments that are long enough. These kinds of gaps occur if there is a point of high distance within a diagonal segment (due to e.g. a variation in the musicians' performance). The enhancement process processes the binarized distance matrix with a kernel of length 25 (beats). Thus, at the position (i, j) of the binarized distance matrix B(i, j), the kernel analyzes the diagonal segment from $B(i, j)$ to $B(i$+25-1, j+25-1). If at least 65 % of the values of the diagonal segment are ones, $B(i, j) = 1$ and either $B(i$+25-2, j+25-2) = 1 or $B(i$+25-1, j+25-1) = 1, all the values in the segment are set to one. This removes short gaps in the diagonal segments. The length of the kernel is a parameter to the system, the value 25 was found to work well. Goto ([8]) did not report a need for such an enhancement process but we found it necessary.

2.7. Locating interesting segments

The result of the previous steps is an enhanced binarized matrix $B_e(i, j)$ where the value one indicates that that point corresponds to a repetitive section and zero corresponds that there is no repetition at that point. The next step is to find diagonal segments that are interesting, i.e. likely correspond to a chorus.

There may be repetitions that are too short to correspond to a chorus, such as those that occur e.g. when the same pattern of notes are repeatedly played with some instrument. By default, segments longer than four seconds are searched and used for further processing. In the case no segments longer than four seconds are found, the system tries to extend the segments until at least some segments longer than four seconds are detected. If this does not help, the length limit is relaxed and all segments are used.

With some songs there may be a very large number of repetitive diagonal segments at this point. Therefore, some of the segments are removed. For each diagonal segment found in the binarized matrix, the method looks for diagonal segments which are located close to it. Let us denote a diagonal segment which starts at (i, j) and ends at (i', j') with $\underline{x}_p = [i, j, i', j']$. Furthermore, the length $\Delta(\underline{x}_p) = j' - j + 1$ is the duration of the segment in beats. Given two segments $\underline{x}_1$ and $\underline{x}_2$, the segment $\underline{x}_2$ is defined to be close to $\underline{x}_1$ iff

$\underline{x}_2(1) \geq (\underline{x}_1(1) - 5)$ and $\underline{x}_2(3) \leq (\underline{x}_1(3) + 20)$ and

$|\underline{x}_2(2) - \underline{x}_1(2)| \leq 20$ and $\underline{x}_2(4) \leq (\underline{x}_1(4) + 5)$

where |.| denotes absolute value. The parameters were obtained by experimentation and may be changed.

For each segment, the method then lists its close segments fulfilling the conditions above, finds the segments that have more than three close segments, and removes the extra segments. If some segment with more than three close segments is in the removal list of some other segment, then it is not removed. The result of this step is a collection of the diagonal segments $\underline{x}_p$, $p = 1,...,P$ in the binarized matrix.

2.8. Selecting the diagonal segment most likely corresponding to a chorus

Next the method selects the segment most likely corresponding to a chorus. This is done by utilizing a novel heuristic scoring scheme which considers aspects such as the position of a repetition in the self distance matrix, the position of a repetition in relation to other repetitions in the SDM, average energy and average distance in the SDM during the repetition, and number of times the repetition occurs in the musical data.

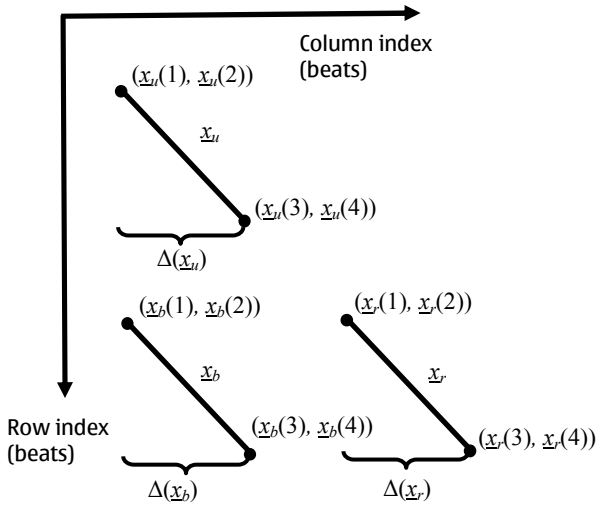

Figure 5: *Notations when giving scores to a group of three diagonal segments (detected stripes of small distance of the distance matrix). The units are measured in beats.*

2.8.1. Position of a repetition in the distance matrix

The first criterion used in making the decision is how close a diagonal segment is to an expected chorus position in the song. This is based on the observation that often in pop music there is a chorus at time corresponding to approximately one quarter of the song length. A partial score $s_1(\underline{x}_p)$ measures the difference of the middle column of segment $\underline{x}_p = [i, j, i', j']$ to one quarter of the song length:

$$s_1(\underline{x}_p) = 1 - \frac{\left|(j + \Delta(\underline{x}_p)/2) - round(M/4)\right|}{round(M/4)}, \quad (3)$$

where *M* is the length of the song in beats. The partial score $s_2(\underline{x}_p)$ measures the difference of the middle row of segment $\underline{x}_p$ to three quarters of the song length:

$$s_2(\underline{x}_p) = 1 - \frac{\left|(i + \Delta(\underline{x}_p)/2) - round(3 \cdot M/4)\right|}{round(M/4)}. \quad (4)$$

With $s_1(\underline{x}_p)$ and $s_2(\underline{x}_p)$ we give more weight to such segments that are close to the position of the diagonal stripe on the low left hand corner of Figure 4, which corresponds to the first occurrence of a chorus (and match to the third occurrence) and is often the most prototypically performed chorus, i.e. no articulation or expression.

2.8.2. Adjustment in relation to other repetitions

The second criterion relates to the adjustment of a segment within the distance matrix in relation to other repetitions. Motivated by the approach presented in [5], we look for possible groups of three diagonal stripes that might correspond to three repetitions of the chorus. See Figure 5 for an example of an ideal case. The search for possible groups of three stripes is done as follows: the method goes through each found diagonal segment $\underline{x}_u$, and looks for possible diagonal segments below it. If a segment below $\underline{x}_b$, $b \neq u$, is found, it looks for a segment $\underline{x}_r$ $r \neq u$, $r \neq b$, on the right from the segment $\underline{x}_b$. In order to qualify as a below segment, we require that $\underline{x}_b(1) > \underline{x}_u(3)$, and that there must be some overlap between the column indices of $\underline{x}_u$ and $\underline{x}_b$. To qualify as a right segment $\underline{x}_r$, there must be some overlap between the row indices of segments $\underline{x}_b$ and $\underline{x}_r$. The groups of three segments fulfilling the above criteria are denoted with $\underline{m}_z = [u, b, r]$, $z = 1,...,Z$. In theory there could be at maximum of $P(P-1)(P-2)$ such groups of three segments, in practice the number is much less. An arbitrary segment may belong to zero or several groups.

The groups of three stripes are then scored based on how close to ideal the group of three stripes is. This scoring affects the scores of some of the segments belonging to these groups. Four partial scores are calculated to measure the quality of each group of three stripes $\underline{m}_z = [u, b, r]$. The first partial score measures how close is the end point of the above segment $\underline{x}_u$ and below segment $\underline{x}_b$:

$$\sigma_1(z) = 1 - 2\left|\underline{x}_u(4) - \underline{x}_b(4)\right| / (\Delta(\underline{x}_b) + \Delta(\underline{x}_u)), \quad (5)$$

where $\underline{x}_u(4)$ and $\underline{x}_b(4)$ are the column indices of the end points of upper and below segments, respectively. The second partial score depends on the vertical alignment of upper and below segments:

$$\sigma_2(z) = \begin{cases} 1-(\underline{x}_u(2)-\underline{x}_b(2))/\Delta(\underline{x}_b) & \text{if } \underline{x}_b(2) < \underline{x}_u(2) \\ 1-(\underline{x}_b(2)-\underline{x}_u(4))/\Delta(\underline{x}_b) & \text{if } \underline{x}_b(2) > \underline{x}_u(4) \\ 1 & \text{otherwise} \end{cases} \quad (6)$$

The next score measures whether the segments $\underline{x}_b$ and $\underline{x}_r$ are of equal length:

$$\sigma_3(z) = 1 - \left|\Delta(\underline{x}_r) - \Delta(\underline{x}_b)\right| / \Delta(\underline{x}_b) . \quad (7)$$

The final partial score depends on the difference in the position of left and right segments:

$$\sigma_4(z) = 1 - \frac{2 \cdot \min(\left|\underline{x}_b(1)-\underline{x}_r(1)\right|, \left|\underline{x}_b(3)-\underline{x}_r(3)\right|)}{\Delta(\underline{x}_b)+\Delta(\underline{x}_r)}, \quad (8)$$

where 'min' denotes minimum operator.

The final score for the group of three segments $\underline{m}_z = [u,b,r]$ is the average of $\sigma_1(z)$, $\sigma_2(z)$, $\sigma_3(z)$, and $\sigma_4(z)$ denoted $\hat{\sigma}(z)$. Since this score considers a segment group, we need to decide whether all the segments in the group receive a score, or whether only certain segments. It was found beneficial to give the score to segment $\underline{x}_b$. The score could also be given to segment $\underline{x}_u$ as it may also correspond to the first instance of the chorus. However, the diagonal stripe corresponding to $\underline{x}_u$ is often longer than the actual chorus, it often consist e.g. of the repeating verse and chorus. It was observed that it gives better results to score the segment $\underline{x}_b$ as its length often more closely corresponds to the correct chorus length. Thus, depending on whether each found segment belong to at least one group of three segments, it will receive a score $s_3(\underline{x}_p) = \max \hat{\sigma}(y)$, $\{y \mid \underline{m}_y(2) = p\}$. The maximum is taken as each segment may belong to more than one group. If a segment $\underline{x}_p$ does not belong to any group of three segments, $s_3(\underline{x}_p) = 0$.

2.8.3. Average energy and distance of a segment

The next criterion $s_4(\underline{x}_p)$ is the average logarithmic energy of the portion of the music signal defined by the column indices of segment $\underline{x}_p$ normalized with the average energy over the whole signal. Using the energy as one criterion gives more weight to such segments that have high average energy, which is often a characteristic of chorus sections. The partial score $s_5(\underline{x}_p)$ takes into account the average distance value during the segment: the smaller the distance during the whole segment the more likely it is that the segment corresponds to a chorus:

$$s_5(\underline{x}_p) = 1 - \phi(\underline{x}_p)/\Phi , \quad (9)$$

where $\phi(\underline{x}_p)$ is the median distance value of the diagonal segment $\underline{x}_p$ in the distance matrix, and Φ is the average distance value over the whole distance matrix.

2.8.4. Number of times the repetition occurs

The last partial score $s_6(\underline{x}_p)$ considers the number of times the repetition occurs. Other diagonal segments locating on top of or below segment $\underline{x}_p$ are indications that the segment defined by the column indices of $\underline{x}_p$ is repeating more than once. To get a score for this, a search is done for all segment candidates $\underline{x}_p$, and a count is made of all those other segments $\underline{x}_q$ which fulfill the condition $\left|\underline{x}_p(2)-\underline{x}_q(2)\right| \le 0.2 \cdot \Delta(\underline{x}_q)$ and $\left|\underline{x}_p(4)-\underline{x}_q(4)\right| \le 0.2 \cdot \Delta(\underline{x}_q)$. The count of other segments $\underline{x}_q$ fulfilling the above criterion is stored as the score for all segment candidates $\underline{x}_p$. When these counts for all segment candidates have been obtained, the values are normalized by dividing with the maximum count, giving the final values for a score $s_6(\underline{x}_p)$ for each segment.

2.8.5. Selecting the most likely chorus segment

The remaining task is to select the most likely chorus segment based on the various criteria. For each segment $\underline{x}_p$, a score is given as

$$\begin{aligned} S(\underline{x}_p) &= 0.5 \cdot s_1(\underline{x}_p) + 0.5 \cdot s_2(\underline{x}_p) \\ &+ s_3(\underline{x}_p) + 0.5 \cdot s_4(\underline{x}_p) + s_5(\underline{x}_p) + 0.5 \cdot s_6(\underline{x}_p). \end{aligned} \quad (10)$$

There is a possibility to optimize the weights in Eq. 10, which we did not fully explore in the fear of over fitting data but just manually selected weights that gave good performance on a small set of music files. The segment $\underline{x}_p$ maximizing the score *S* is selected as the most likely chorus segment. If at least one group of three diagonal stripes fulfilling the criteria of section 2.8.2 has been found, the choice is made among such segments $\underline{x}_o$ for which $s_3(\underline{x}_o) \neq 0$, i.e. those that have been at an appropriate position in at least one group of three diagonal stripes. If no sets of three stripes is found, the selection is made among all the segments by maximizing *S*. In this case the group score $s_3(\underline{x}_p) = 0$ for all segment candidates. The result of this step is an initial chorus segment $\underline{x}_c$.

2.9. Finding the exact location of the chorus

Next the exact location and length of the chorus section is refined using filtering in two or one dimensions. 2D kernels have earlier been used by Shiu et al. to analyze local similarity of the signal by detecting repeated chord successions from a measure-level self-similarity matrix [15]. Here, we use 2D filters to get the exact position for a chorus segment. Often, the time signature in western pop and rock music has a 4/4 time signature, and the length of a chorus section is 8 or 16 measures (32 or 64 beats, respectively) [9]. In addition, the chorus may consist of two repeating subsections of equal length. Two dimensional filter kernels are constructed to model the pattern of ideal small-distance stripes that would be caused by a chorus of 8 or 16 measures long, with two repeating subsections. Figure 6 shows the filter of dimension 32 by 32 beats, with two 16 by 16 beats long repeating subsections. This is the idealized shape of the small-distance stripes occurring in the distance matrix if the song has this kind of chorus. The second filter is

similar but of dimension 64 by 64, and with diagonals modeling the 32 beat long subsections.

The area of the distance matrix surrounding the chorus candidate is filtered with these two kernels. The chorus candidate start column is denoted with $\underline{x}_c(2)$ and the end column $\underline{x}_c(4)$. The columns of the low triangular distance matrix starting from $\max(1, \underline{x}_c(2) - N_f/2)$ to $\min(\underline{x}_c(4) + N_f/2, M)$ are selected as the area from which to search for the chorus. N_f is the dimension of the filter kernel, either 32 or 64, and M is the length of the song in beats. min and max are applied to prevent over indexing. If the length of the area above in the column dimension is shorter than the filter dimension, this step is omitted. The area is limited to lessen the computational load and to prevent the refined chorus section from departing too much from the initial chorus candidate.

When the upper left-hand side corner of the filter with dimension N_f is positioned in (i, j) at the distance matrix, the following values are calculated: mean distance $\alpha(i, j, N_f)$ along the diagonals (marked with black color in Figure 6), mean distance $\beta(i, j, N_f)$ along the main diagonal and mean distance $\lambda(i, j, N_f)$ of the surrounding area (white color in Figure 6). The ratio $\rho_\alpha(i, j, N_f) = \alpha(i, j, N_f) / \lambda(i, j, N_f)$ indicates how well the position matches with a chorus with two identical repeating subsections, and the ratio $\rho_\beta(i, j, N_f) = \beta(i, j, N_f) / \lambda(i, j, N_f)$ how well the position matches a strong repeating section of length N_f with no subsections. The smaller the ratio, the smaller the values on the diagonal compared to the surrounding area. The smallest value of $\rho_\alpha(i, j, N_f)$ and $\rho_\beta(i, j, N_f)$ and the corresponding indices are stored for both filters, i.e. with N_f=32 and N_f =64. These smallest values are denoted by $\rho'_\alpha(N_f)$ and $\rho'_\beta(N_f)$.

Several heuristics are then used to select the final chorus position and length based on the filtering results, or if the conditions are not met then another filtering in one dimension along the initial chorus segment is performed. The final chorus section is selected according to the two dimensional filtering, if the smallest ratios are small enough. The following heuristics are used, although many other alternatives would be possible. These rules below have been obtained via trial and error by experimenting with a subset of 50 songs from our music collection.

If $\rho'_\alpha(64) < \rho'_\alpha(32)$, it indicates a good match with the 64 beat long chorus with two 32 beat long subsections. The chorus starting point is selected according to the column index of the point which minimized $\rho'_\alpha(64)$, and its length is taken as 64 beats. Else, if the length of the initial chorus section is less than 32, the chorus section is adjusted according to the point minimizing $\rho'_\alpha(32)$ only if the chorus beginning would change at maximum one beat from the initial location. Finally, if the length of the initial chorus section is closer to 48 than 32 or 64 and $\rho'_\alpha(32) < \rho'_\alpha(64)$ and $\rho'_\beta(32) < \rho'_\beta(64)$ and the column index of the point minimizing $\rho'_\alpha(32)$ is the same as the point minimizing $\rho'_\beta(32)$, the chorus is

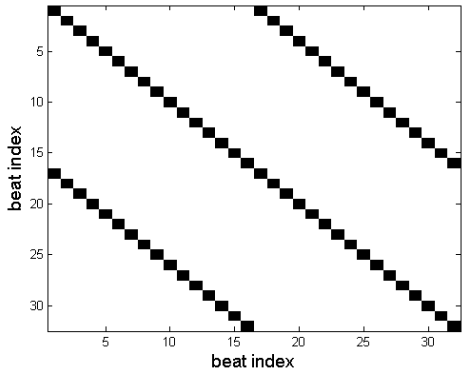

Figure 6. *Two dimensional filter kernel modelling the stripes occurring if the song has a chorus of 32 beats in length with two 16 beat repeating subsections. The average distance is calculated along the entries marked with black colour, and compared to the average distance of locations corresponding to rest of the kernel (white entries).*

set to start at the point minimizing both $\rho'_\alpha(32)$ and $\rho'_\beta(32)$ and its length is set to 32. Thus, these are special rules to adjust the chorus section if it seems likely that there song has either a 32 or 64 beats long chorus with identical subsections half its size.

In many cases, the above conditions are not met, and the chorus section is adjusted by performing filtering along the diagonal values of the initial chorus section and a small offset of five beats before and after its beginning and end. Thus, if the row and column indices of the initial chorus section are denoted with $(\underline{x}_c(1), \underline{x}_c(2))$ (the beginning) and $(\underline{x}_c(3), \underline{x}_c(4))$ (the end), the values to be filtered are found along the line from $(\underline{x}_c(1) - 5, \underline{x}_c(2) - 5)$ to $(\underline{x}_c(3) + 5, \underline{x}_c(4) + 5)$.

The filtering is done with two kernels of length 32 and 64, but now on one dimension along the diagonal distance values of the initial chorus section and its immediate surroundings. The ratio $r(32)$ is the smallest ratio of mean of distance values on the 32 point kernel to the values outside the kernel. If $r(32) < 0.7$ and the length of the initial chorus section is closer to 32 than 64, the chorus starting point is set according to the location minimizing $r(32)$ and its length is set to 32. If the length of the initial chorus section is larger than 48, the final chorus start location and length is selected according to the one giving the smaller score. This step in our method looks for the best position of the chorus section e.g. in the case the diagonal stripe selected as the chorus section consists of a longer repetition of a verse and chorus, for example. Note that the method is not limited to 4/4 time signature and chorus lengths of 32 or 64: if the conditions above are not met, the chorus section is kept as the one returned from the binarization process. In these cases its length does not have to be 32 or 64.

3. EVALUATION

The method was evaluated on database consisting of 206 popular and rock music pieces. Most of the pieces have a clear verse-chorus structure, although there are some instances where the structure is

less obvious. The chorus sections were annotated manually from the pieces. The annotations were made with a dedicated tool, which showed the beat synchronized SDM of the signal aligned with the signal itself. The self-distance matrix visualization significantly speeded up the annotation work as the different sections were more easily found.

Performance of the system is measured with the F-measure, defined as the harmonic mean of the recall rate (R) and precision rate (P): $F = (2RP) / (R + P)$. To calculate R and P, we find the annotated chorus section with maximum overlap with the detected chorus section, and calculate the length l_{corr} of the section where the detected chorus section overlaps with the annotated section. R is calculated as the ratio l_{corr} to the length of the annotated chorus section, and P is the ratio of l_{corr} to the length of the detected chorus section. The F-measure is calculated for each track, and the reported overall F-measure is the average of the F-measures over all tracks.

Table 1 shows the chorus detection results. Baseline is the normal system. The most common error is small offsets in the beginning and/or end locations of the chorus section that reduce the score. The second row represents the results when the output chorus section length is fixed to 30 seconds. Being able to output a fixed length segment may be desirable in some applications, such as music preview. If the initial chorus section is shorter than 30 seconds, expanding is done by following the diagonal chorus segment into the direction of minimum distance in the SDM. Correspondingly, shortening is done by dropping in turn the point with larger distance value from either end. As the recall rate increases when the 30 s limit is applied, the method has not always captured the whole chorus section. If it is desirable that the thumbnail section captures the chorus and it's acceptable if the section extends beyond the chorus, the 30s option can be used. The method is efficient; it takes about ten seconds to process a song with an average duration of three to four minutes on a Windows XP computer with a 2.8 GHz Intel Xeon processor.

Method	P	R	F
Baseline	89%	83%	86%
30s length	70%	92%	79%

Table 1: *Chorus detection results.*

4. CONCLUSIONS

A method for chorus detection from popular and rock music was presented. The method utilizes a novel feature analysis front-end where the MFCC and chroma distance matrices are summed and a two step procedure of initial chorus selection and section refinement. A novel heuristic scoring scheme was proposed to select the initial chorus candidate from the binarized distance matrix, and a novel approach utilizing image processing filters is used to refine the final position and length of the chorus candidate. Evaluations on a manually annotated database of 206 songs demonstrate that the performance of the method is sufficient for practical applications, such as previewing playlists of popular and rock music. Moreover, the method is computationally efficient.

5. REFERENCES

[1] G. Peeters, A. La Burthe, X. Rodet, "Toward Automatic Music Audio Summary Generation from Signal Analysis", in *Proc. of the 3rd International Conference on Music Information Retrieval, ISMIR 2002*, Paris (France), October 2002.

[2] B. Logan, S. Chu, "Music summarization using key phrases," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, *ICASSP 2000*, vol. 2, pp. 749-752, Istanbul, Turkey, May 2000.

[3] M. Levy, M. Sandler, M. Casey, "Extraction of High-Level Musical Structure From Audio Data and Its Application to Thumbnail Generation," in *Proc. IEEE ICASSP 2006*, vol. V, pp. 13-16.

[4] C. Rhodes, Casey, S. Abdallah, M. Sandler, "A Markov-chain monte-carlo approach to musical audio segmentation," in *Proc. IEEE ICASSP 2006*, vol. V, pp. 797-800.

[5] J. Wellhausen and H. Crysandt, "Temporal Audio Segmentation Using MPEG-7 Descriptors," in *Proc. of the SPIE International Symposium on ITCom 2003 - Internet Multimedia Management Systems IV,* Orlando (FL), USA, September 2003.

[6] M. Cooper, J. Foote, "Summarizing Popular Music Via Structural Similarity Analysis," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2003*, October 19-22, 2003, New Paltz, NY.

[7] M. A. Bartsch, G. H. Wakefield, "Audio Thumbnailing of Popular Music Using Chroma-Based Representation," *IEEE Trans. on Multimedia*, vol. 7, no. 1, Feb. 2005, pp. 96-104.

[8] M. Goto: "A Chorus Section Detection Method for Musical Audio Signals and Its Application to a Music Listening Station," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 5, Sept. 2006 pp. 1783 – 1794.

[9] N. Maddage, "Automatic Structure Detection for Popular Music," *IEEE Multimedia*, Jan.-March 2006, vol. 13, no. 1, pp. 65-77.

[10] J. Paulus, A. Klapuri, "Music Structure Analysis by Finding Repeated Parts", in *Proc. of the 1st Audio and Music Computing for Multimedia Workshop (AMCMM2006)*, Santa Barbara, California, USA, October 27, 2006, pp. 59-68.

[11] J. Seppänen, A. Eronen, and J. Hiipakka, "Joint Beat & Tatum Tracking from Music Signals", In *Proc. of the 7th International Conference on Music Information Retrieval, ISMIR 2006,* Victoria, Canada, 8 - 12 October 2006.

[12] D. Ellis, "Beat Tracking with Dynamic Programming", MIREX 2006 Audio Beat Tracking Contest system description, Sep 2006, available at http://www.ee.columbia.edu/~dpwe/pubs/Ellis06-beattrack.pdf

[13] M. Marolt, A Mid-level Melody-based Representation for Calculating Audio Similarity, In *Proc. of the 7th International Conference on Music Information Retrieval, ISMIR 2006,* Victoria, Canada, 8 - 12 October 2006.

[14] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62-66, Jan. 1979.

[15] Y. Shiu, H. Jeong, C.-C. Jay Kuo, "Similarity Matrix Processing for Music Structure Analysis", In *Proc. of the 1st Audio and Music Computing for Multimedia Workshop (AMCMM2006)*, October 27, 2006, Santa Barbara, California, USA.