MUSIC STRUCTURE DISCOVERY BASED ON NORMALIZED CROSS CORRELATION

Alexander Wankhammer,

Institute of Electronic Music and Acoustics,
University of Music and Performing Arts
Graz, Austria
wankhammer@iem.at

I. Vaughan L. Clarkson, Andrew P. Bradley,

School of ITEE,
University of Queensland
Brisbane, Australia
v.clarkson@itee.uq.edu.au,
a.bradley@itee.uq.edu.au

ABSTRACT

Music Structure Discovery (MSD) for popular music is a well known task in Music Information Retrieval (MIR). The proposed approach tries to find the basic musical structure of a piece of music, by applying a template matching algorithm on a modified, bar level Self Distance Matrix (SDM). Mel frequency cepstral coefficients (MFCC) are used to represent timbral qualities of the audio material while chroma vectors are selected to incorporate pitch and harmonic content. The new idea of template matching instead of trying to find explicit blocks or off-diagonal lines is independent of any specific characteristics of the underlying SDM and can therefore be used on a wide range of different songs.

1. INTRODUCTION

If we listen to music, we are literally surrounded by repetitive structures and varying patterns. We will hear different combinations of melodic and harmonic progressions, ongoing rhythmic movements and on a wider time scale changes in timbre and the dynamics of the song. Despite this variety of patterns, most MSD algorithms aim to detect a specific high-level musical structure often referred to, in music theory, as the musical form.

The musical form can be seen as the decomposition of a song into its major building blocks. Every building block has its own label and can occur at various times throughout the song. Typical labels in popular music are for example intro, verse, chorus, bridge and outro. Although the exact determination of the musical form is not always unambiguous, most people will unconsciously split songs into closely related blocks when listening to music. Therefore, detecting the musical form is a reasonable and natural objective for MSD [1].

Knowing the structure of a piece of music is useful in various fields. For examples, it can help to support other MIR applications like the detection of different versions of the same song [2] or audio thumbnailing [3], [4], [5]. Alternatively, the results can be used to improve the usability of a wide range of existing audio applications, such as allowing more intuitive navigation within pieces when using audio players or Digital Audio Workstations (DAW) [6], [7].

Since the creativity of composers is the only limit for the variety of differences and similarities among song segments, several approaches to solve the problem of musical structure detection have been developed. Typically, a set of appropriate features is initially derived from a short time spectral representation of the audio file. Most of these features have been found to be adequate descriptors for either one or several different aspects of human cognition of music. Therefore, the most commonly used features in MSD are often based on timbre [8], [9], pitch and harmony [4], rhythm or a set of multiple descriptors [1], [10], [11], [12].

Once the feature sequences are extracted, the search for repetitive parts and related sections can mainly be focused on two different temporal qualities: sequences and states [10]. Sequence-based approaches try to find clear repetitions of consecutive feature sequences [4], [10], [13], whereas state-based approaches handle the feature sequence as a succession of different states and try to find relations by applying clustering algorithms [14] or hidden Markov models (HMM) [8], [15]. These two basic types of repetitions become apparent, when a Self Distance Matrix (SDM) is used to visualize the temporal structure of a song. Given a feature vector sequence $V[n]$ consisting of single feature vectors $\vec{v_i}, i = 1, 2, \ldots, N$, the SDM $S(i, j)$ represents the distance between each pair of feature vectors over time. Therefore, an off-diagonal line inside $S(i, j)$ corresponds to the repetition of a certain sequence of consecutive feature vectors, whereas a rectangular block represents a group of overall similar feature vectors, potentially belonging to the same state. [1]

In this paper we use the well known mel frequency cepstral coefficients (MFCC) to represent timbral qualities of the audio material and the chroma or pitch class profile (PCP) as an abstract descriptor for pitch and harmonic progression. The search for repetitions is then based on a combined, modified SDM using Normalized Cross Correlation (NCC). The main contribution of this paper is to search for similarities between vertical slices within a similarity matrix, instead of trying to find explicit blocks or off-diagonal line segments. This makes the detection of repetitions widely independent of any distinct structures in underlying SDM, as will be explained in more detail in Section 2.2.

The remainder of this paper is organized as follows: Section 2 gives an overview of the implemented system, starting with the basic feature extraction (2.1) before focusing on the structural analysis based on the new idea of template matching using NCC (2.2). In Section 2.3, we outline the segment detection algorithm leading to the final song structure. Section 3 provides an evaluation of the system on a small testing corpus and gives a short analysis of the overall performance while Section 4 offers a brief outlook of possible future work.

[1] A good example for this kind of structures can be found in Figure 1b (right). Part A (verse) shows a quite distinct line structure, whereas part B (bridge) reveals a strong block like structure.

2. PROPOSED METHOD

2.1. Feature Extraction

As our selected features both rely on a short time spectral representation of the input signal, a Hanning windowed Short Time Fourier Transform (STFT) of length $N_{STFT} = 4096$ is computed. To assure a constant temporal resolution of about 45ms for each processed frame, all songs are resampled to $f_s = 11025Hz$ and the hop-size between adjacent frames is set to $k_{STFT} = 512$.

Chroma has been demonstrated to be a successful basic indicator of the harmonic and melodic progressions of music as it measures the spectral energy related to the 12 semitones of the well-tempered scale. We use a constant-Q filterbank [16], in which every single frequency band k_{cq} represents one semitone and we receive the chroma vector $\vec{c}$ by summing the energy over all the bins belonging to one tone $\vec{c}(k_{cq}) = mod(k_{cq}, 12)$. To avoid misinterpretation of songs not played according to the standard tuning frequency, we perform a tuning of the filterbank by detecting the center of spectral energy within ± 1 quarter-tone around 440Hz.

In addition to the chroma, the MFCCs of every frame are calculated using a 42 band Mel-filterbank. MFCCs have been utilized in audio and speech applications for many years as they are a powerful method for describing timbral properties, incorporating the non-linear frequency and energy reception of the human auditory system. For our method, we chose 10 coefficients, including the zeroth coefficient.

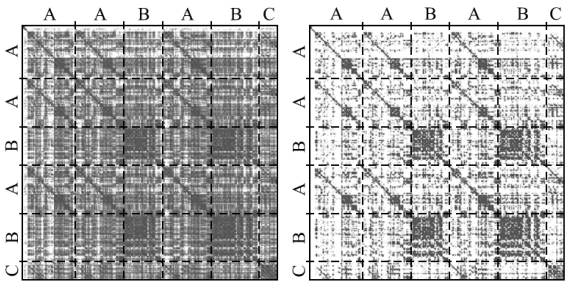
After feature calculation, all features are averaged over the period of one beat. Beat-averaging offers a tempo-invariant time base for further computations as well as a more stable representation of the extracted features. To avoid blurring of beat averaged results by transient events, we define an offset of about 45ms at the beginning and end of every beat. For beat detection we use the method proposed by Ellis [17] that has been shown to perform well and in a stable manner.

2.2. Structural Analysis

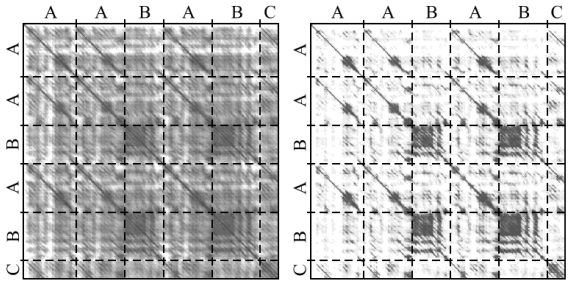
2.2.1. Embedding and mapping

SDMs have been widely used for musical structure detection [4], [13]. In [18] it has been proposed to use measure level similarity matrices instead of frame or beat based matrices for MSD, as bars are the smallest natural building blocks of a higher-scale musical structure. In our approach we follow a very similar idea inspired by the concept of the embedding dimension known from recurrence plots (RP) [19].

RPs have been developed as a tool for nonlinear data analysis helping to visualize and understand the recurrent behavior of dynamical systems. In RP analysis, the embedding dimension ρ defines how many time instances of a feature sequence are combined to calculate the RP. Therefore, $S(i, j)$ is constructed by computing distances between all pairs of embedded vectors $\vec{e_i} = (\vec{v_i}^T, \vec{v}^T_{i+1}, \ldots, \vec{v}^T_{i+(\rho-1)})$ and $\vec{e_j} = (\vec{v_j}^T, \vec{v}^T_{j+1}, \ldots, \vec{v}^T_{j+(\rho-1)})$. The simplified idea behind the embedding dimension of recurrence plots states that each single observed parameter of a dynamic system (e.g. air pressure) contains important information about the dynamics of the whole system (e.g. weather). By defining an "adequate" embedding dimension, the behavior of the overall system can be reconstructed by only using one embedded parameter. Although music is not a natural dynamic system, it reveals a clear dependency on multiples of beats and bars. When comparing Figure 1a and Figure 1b, one can see that setting the embedding dimension to values $\rho > 1$ clearly enhances the clarity of off-diagonal lines in the SDM.



(a) *beat level SDMs (no embedding)*



(b) *bar level SDMs (embedding dimension $\rho = 4$)*

Figure 1: *SDMs with different embedding dimensions before (left) and after (right) mapping operation [Beatles - All'I've Got To Do]*

Based on the aim to find bar level relations, we use an embedding dimension of $\rho = 4$ corresponding to the four beats forming one bar, when analyzing songs written in common time ($^4/_4$ bar). Euclidean Distance is then used for computing two bar level distance matrices.

The two resulting SDMs are finally normalized to $[0, 1]$ and combined into one matrix $S_{cmb}(i, j)$ by pointwise multiplication. To further reduce noisy information before applying the template matching algorithm, the values of the combined bar level SDM are mapped by a continuous function (1), enforcing areas of high similarity while suppressing areas of low similarity (see Figure 2).

$$S_{map}(i, j) = 0.5 - 0.5 * \tanh[\pi * \lambda * (S_{cmb}(i, j) - \gamma)] \quad (1)$$

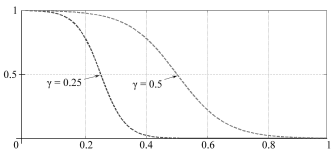


Figure 2: *Mapping function with $\gamma = 0.25$ and $\gamma = 0.5$. (In our case, γ is always smaller than 0.4 due to the used thresholding method.)*

To find an adequate threshold γ for the mapping operation, we apply Otsu's method on $S_{cmb}(i, j)$, neglecting values larger than 0.4. Otsu's method is a widely used algorithm for histogram based

image thresholding which tries to minimize the intraclass variance of two classes of variables. For more detailed information, please refer to [20]. The additional constraint to use values smaller than 0.4 focuses the thresholding operation on the relevant parts of the matrix, helping to keep important structures while adequately suppressing noisy areas. Examples for mapped SDMs can be found in Figure 1 (right). Depending on the resulting threshold γ, the parameter λ is set automatically to map 0 to 1. Although the resulting matrix looks similar to a binary recurrence plot, the continuous mapping preserves the fine structure of highly repetitive areas, which is important for template matching.

As mentioned in Section 1, different MDS systems often focus on different aspects of repetitions, namely states (rectangular blocks showing areas of high similarity) or sequences (off-diagonal lines). Approaches only relying on specific characteristics of the SDM sometimes fail, as songs do not always show clear off-diagonal lines or distinct blocks of high similarity. Therefore, we propose to use a template matching algorithm based on Normalized Cross Correlation to find repetitive parts. Exploiting the inherent symmetry of SDMs, it directly compares different similarity profiles (SP) (vertical SDM slices) and is therefore independent of any specific structure.

2.2.2. Template Matching

Normalized Cross Correlation is a standard method for image registration (template matching) in various fields of digital image processing [21]. Using NCC for finding a template image T within a search image I results in a cross correlation matrix $C(i,j)$, showing maxima at positions of high correlation. This can be computationally expensive, as the NCC has to be computed at all possible positions of the template with respect to the search image.

$$C(i,j) = \ldots \\ \frac{\sum_{x,y}[I(x,y) - \overline{I}_{i,j}][T(x-i,y-j) - \overline{T}]}{\{\sum_{x,y}[I(x,y) - \overline{I}_{i,j}]^2 \sum_{x,y}[T(x-i,y-j) - \overline{T}]^2\}^{0.5}} \quad (2)$$

Equation 2 shows the general form of the NCC. The sums run over x, y in the region under the template positioned at i, j, $\overline{I}$ is the mean of the search image in the same region and $\overline{T}$ is the mean of the template. Fortunately, the number of necessary computations is drastically reduced when we are trying to find repetitions in the SDM. The template T is always simply a vertical or horizontal slice of the search image I (due to the inherent symmetry of the SDM, vertical or horizontal slices represent the same 90° shifted similarity profile) and the template only has to be shifted into one direction. In our approach, we use vertical slices and evaluate all horizontal shifts of the templates across the search image. Therefore, i is always 1.

To further reduce computational cost, the means $\overline{T}$ and $\overline{I}$ for all possible template positions can easily be calculated in advance. As two bars are basically the smallest sequential building block of a song, we chose a template width of $w = 8$. The hop size between adjacent templates is set to 1.

Considering an SDM of dimension NxN and columns $\vec{s}_{1,2,\ldots,N}$, the initial template $T_1 = [\vec{s_1}, \vec{s_2}, \ldots, \vec{s_w}]$ is evaluated over the related search image $I_1 = [\vec{s}_{w+1}, \vec{s}_{w+2}, \ldots, \vec{s}_N]$ (see Figure 3). Since the resulting cross correlation vectors $\vec{c}_{1,2,\ldots,N}$ become shorter with each hop, all vectors are zero padded to length N and stored into a matching matrix $M(i,j)$. The relevant part of $M(i,j)$ is presented as the upper triangular matrix in Figure 4.

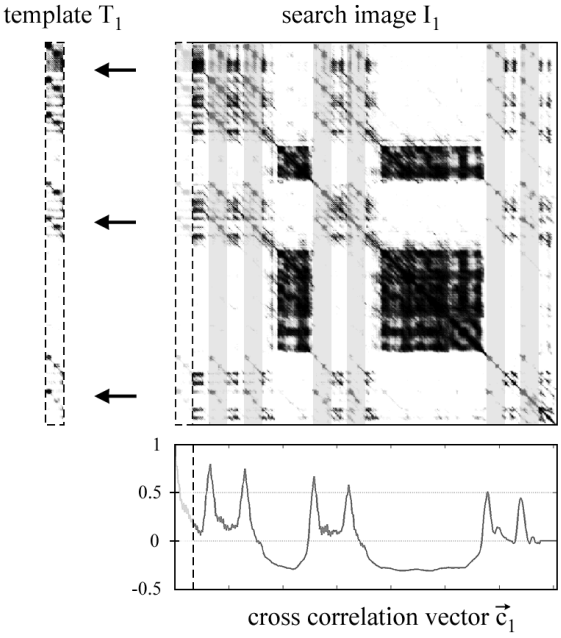


Figure 3: *As indicated with light gray areas, maxima in the cross correlation vector $\vec{c}_1$ mark possible repetitions of the initial template T_1 in the related search image I_1.[Radiohead - Creep]*

To find all valid repetitions, the correlation maxima in the columns $\vec{m}_{1,2,\ldots,N}$ of $M(i,j)$ have to be detected in a first step. To decide whether a peak in $\vec{m}_j$ represents a valid detection or not, an individual threshold for each bar of the song is computed. Similar as for the mapping threshold γ, we use Otsu's method on each row of $M(i,j)$, neglecting all values smaller than 0.5, to find a good threshold for valid peaks. In Figure 4 (right) it can be seen that the resulting vector $\vec{t}(j)$ shows high thresholds on time instances with high overall correlation values while offering moderate thresholds in other areas.

After finding a list of repetitions for each template using $\vec{t(j)}$, the results are marked as valid detections within binarized vectors. The vectors are then stored into a matrix $M_{bin}(i,j)$, showing repetitive sequences as continuous, diagonal lines (see Figure 4, lower triangular matrix). Since $M_{bin}(i,j)$ is based on template matching, it will even show off-line diagonal lines if the underlying SDM did not expose such a structure.

2.2.3. Audio novelty function

Template matching will consider groups of consecutive segments, always occurring in the exact same context throughout the song, as one large segment. Therefore, we use the audio novelty function $\vec{\eta}(j)$ [9] with slight modifications as simple additional indicator for potential segment borders. Typically $\vec{\eta}(j)$ results from correlating a checkerboard-like kernel matrix K along the main diagonal of a full SDM. Segment borders are then found by detecting peaks inside $\vec{\eta}(j)$.

We basically followed this standard approach with two minor changes. First, we correlate the kernel K with the mapped matrix $S_{map}(i,j)$ instead of using the full SDM $S_{cmb}(i,j)$, to obtain clearer peaks in $\vec{\eta}(j)$. Second, we set the main diagonal

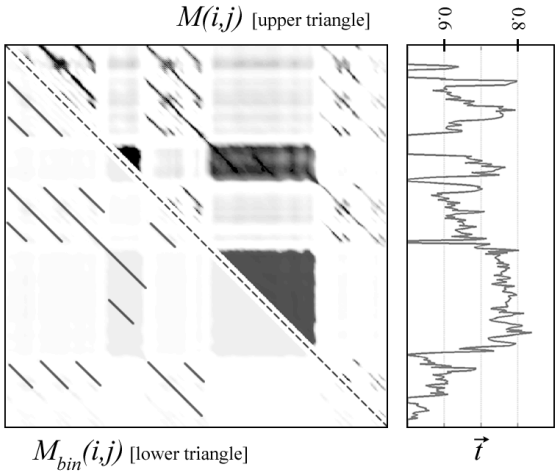


Figure 4: *The upper triangular matrix shows the full matching matrix $M(i,j)$, while the lower triangular matrix represents the binarized matching matrix $M_{bin}(i,j)$. The threshold vector $\vec{t}$ (right) is needed to perform this binarization. [Radiohead - Creep]*

and respectively the first four off-line diagonals (corresponding to one bar) of K to zero, to avoid high correlation values for areas showing only very strong bar level dependencies, but no real long-term similarities. The final border candidates are then selected by a strict peak picking algorithm and stored into a vector $\vec{s}$. The peak detection algorithm is based on an adaptive sliding-median threshold and a very high lower-bound, related to the largest possible correlation value regarding the kernel K, to avoid adding any wrong detection in this step.

The following relatively straight forward segment detection steps could be used on any other kind of binarized similarity matrix and has mainly been designed to show that the concept of template matching could be an interesting alternative to pure line-detection methods.

2.3. Segment Detection

To simplify the detection of meaningful segments, $M_{bin}(i,j)$ is cleaned up by removing discontinuities within longer segments and deleting very short line segments (both thresholds are set to $5s$). Further, a linked list connecting all repetitions belonging to every single bar is created to exploit the dependencies of repetitions targeting the same time interval.

To illustrate the idea, let $\vec{r}_a$, $\vec{r}_b$ and $\vec{r}_c$ be lists of repetitions belonging to bars b_a, b_b and b_c, where b_a is repeated at b_b which again is repeated at b_c. Since all repetitions found for b_b and b_c are implicitly related to b_a, a linked list $\vec{r}_a \rightarrow \vec{r}_b \rightarrow \vec{r}_c$ revealing these higher-order relations is constructed to find all repetitions of b_a, even those not included in the original list $\vec{r}_a$. This linked list therefore creates a repetition profile of the whole song.

After incorporating this new information into $M_{bin}(i,j)$, an initial set of U repetitions $\Phi_{all} = \{\vec{\varphi}_u | u = 1, 2, \ldots, U\}$ can be extracted. Each repetition $\vec{\varphi}_u$ is characterized by the starting index of the related song segment, the corresponding time lag and its length. Φ_{all} is then extended by the border candidates $\vec{s}$ found by the audio novelty function. They are added with a time lag of zero, as they already represent non shifted starting indices. Further, stand-alone segments which are neither repeated nor the target of a repetition are extracted from the repetition profile and are likewise added to Φ_{all} with a time lag of zero.

In a next step, repetitions related to song segments starting within a time frame of $\pm 4s$ are grouped into G_{all} subsets $\Phi_{all\ 1,2,\ldots,G_{all}}$. After this grouping operation, the time lags belonging to each repetition of a subset $\Phi_{all\ j}$ are added to their corresponding starting indices and stored into a vector $\vec{v}_j$. This vector shows all actual occurrences of the song segment related to $\Phi_{all\ j}$. As some subsets will only represent different repetitions of the same song segment, the entries of all vectors $\vec{v}_{1,2,\ldots,G_{all}}$ are compared and subsets belonging to vectors with a significant number of overlaps (allowing a deviation of $\pm 2s$) are merged. Based on these final subsets $\Phi_{fin\ 1,2,\ldots,G_{fin}}$, a two-columned list L_{start} containing the occurrences of each song segment and their corresponding subset IDs is constructed.

It has to be mentioned that we only focus on the starting positions of each segment and, for the moment, ignore any information about their estimated durations. We directly transform the list of starts L_{start} into a list of segments L_{seg}, where each segment is defined by a corresponding ID and runs from its starting index to the closest starting index in L_{start}. Too avoid segments which are too short, occurring if two starting indices in L_{start} are very close together and have different IDs, we remove all segments $< 8s$. As we always want to keep the starting index related to the song segment that occurred earlier, the starting index with the higher ID is ignored.

If each segment of the investigated song is defined by a start in L_{start}, the resulting segmentation will already represent the final structure of the song. The ignored information of segment durations is only critical, if two or more segments are repeated as identical groups. For example, when trying to detect the song structure $A - A - B - A - B - C$, the starts of segment B will not be detected, as B is a subsegment of the sequence $A - B$. Since such "double" segments tend to be unusually long, we define a temporal threshold of $30s$ and check all longer segments for an eventually missed segment transition.

As truncating segments which are potentially too long can not be based on existing starting indices, we use the so far neglected information of segment lengths (or stops) to find alternative segment borders. We check all segment lengths of the related segments in $\Phi_{fin\ j}$ and compute the resulting potential stops. If any of these lengths approximately matches the length of another segment with the same ID, the segment is truncated to this length. Otherwise, the segment is accepted as too long. If the segment is truncated, the position of the new stop is added as a potential start to the segment list. Further, all segment information is updated starting from the grouping operation to incorporate the new start.

3. EVALUATION

As already mentioned in Section 1, the musical form of a song is rarely unambiguous and basically only the composers could ultimately define the *true* musical form of their compositions. Still, every evaluation of musical structure has to be based on a predefined ground truth.

To evaluate our system, we use a body labeled by either professional musicians and/or musicologists for the MPEG-7 working group. The testing corpus, similar to the one used in [1], consists of 32 songs and includes the full Beatles album "With the Beatles" as well as a list of more recent pop songs by artists like Alanis

Morisette, Björk, Madonna or The Spice Girls. Since the labeling is done in seconds, all segment borders have to be quantized to beat-level, granting a common time base for the automatically detected segmentation and the annotated ground truth.

3.1. Evaluation metrics

We use two different evaluation metrics which have widely been used in musical structure detection before [22]: the pairwise F-measure and the directional Hamming distance.

The pairwise F-measure F is a standard evaluation metric for clustering algorithms and represents the harmonic mean of the pairwise precision P_r and recall rate R_r. Let Γ_r be a set of identically labeled pairs of beats in the reference segmentation and Γ_d in the automatically detected segmentation. With $|\cdot|$ denoting the cardinality of the respective set, the measures are defined as:

$$P_r = \frac{|\Gamma_d \cap \Gamma_r|}{|\Gamma_d|} \tag{3}$$

$$R_r = \frac{|\Gamma_d \cap \Gamma_r|}{|\Gamma_r|} \tag{4}$$

$$F = \frac{2 \cdot P_r \cdot R_r}{P_r + R_r} \tag{5}$$

A low pairwise precision rate is an indicator for under segmentation, while a low pairwise recall rate indicates over segmentation. The pairwise F-measure therefore describes an overall quality of the found segmentation.

The second metric is the directional Hamming distance [14]. Given the reference segmentation as a sequence of n segments $R = \{S_R^1, S_R^2, \ldots, S_R^n\}$ and the automatically detected segmentation as a sequence of m segments $D = \{S_D^1, S_D^2, \ldots, S_D^m\}$, the directional Hamming distance is denoted by $D_H(R \Rightarrow D)$. For each segment S_D^i from the detected segmentation a segment S_R^j from the reference segmentation is associated so that the overlap of the segments $S_D^i \cap S_R^j$ is maximal. The directional Hamming distance is then defined as:

$$D_H(R \Rightarrow D) = \sum_{S_D^i} \sum_{S_R^k \neq S_R^j} |S_D^i \cap S_R^k| \tag{6}$$

Similarly, the *inverse* directional Hamming distance can be symmetrically computed for $D_H(D \Rightarrow R)$. Normalizing the resulting distances by the number of beats N of the underlying track allows us to derive two error rates: the missed boundaries (or miss rate) $m = D_H(R \Rightarrow D)/N$ and the segment fragmentation (or false alarm rate) $f = D_H(D \Rightarrow R)/N$. Low values of m indicate under segmentation, while low values of f indicate an over segmentation.

3.2. Results

Table 1 shows the overall performance of the system based on the pairwise F-measure while Figure 5 shows a scatter plot of the *missed boundaries* and the *segment fragmentation* for all analyzed songs. Even if the overall evaluation results for both metrics may not be fully comparable to other methods due to our relatively small testing corpus, they indicate that segment detection based on NCC could be a promising new approach in the field.

The overall performance of the algorithm on the *Beatles* corpus is slightly better than on the selection of modern songs, as most of the *Beatles* songs have a quite simple structure and long repetitive sequences. Since our algorithm is based on chroma vectors and MFCCs, strong timbral variations in segments of the same formal part (e.g. changes in the overall instrumentation) sometimes lead to over-segmentation. Although such songs may exhibit poor evaluation results compared to a reference segmentation, the over-segmentation is mainly caused by the used features and not the template matching algorithm itself.

A real problem for the template matching algorithm are songs with a lot of very short repeated areas (e.g. *REM-Drive*), as too many repetitions are found to be "valid". This problem could either be solved by a more sophisticated post-processing step or a multi-scale approach as proposed in Section 4.

Two typical segmentations detected by our algorithm are illustrated in Figure 6. Although both segmentations show slight temporal shifts of the detected segments (D) compared to the underlying reference segmentation (R) (sometimes the transition between two adjacent segments is assigned to the wrong segment), they still offer a good representation of the song structure.

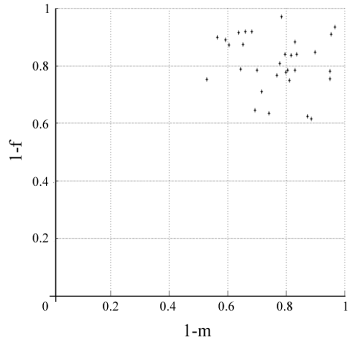


Figure 5: *Scatterplot of "segment fragmentation" against "missed boundaries" for all songs in the testing corpus.*

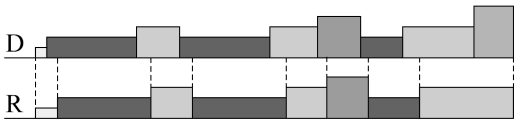
Corpus	R_r	P_r	F
Beatles	70	78	72
Recent	64	71	66
Overall	**67**	**74**	**69**

Table 1: *Pairwise F-measure (%)*

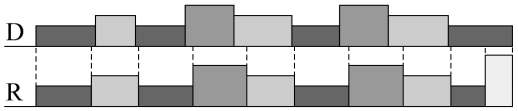
4. CONCLUSION AND FUTURE WORK

A system for automatic music structure analysis based on NCC and measure-level SDMs has been presented. We introduced the embedding dimension known from RP analysis as a tunable parameter to enhance the clarity of certain temporal dependencies inside SDMs on different time scales (in our case measures). Besides, we developed a mapping operation based on Otsu's method as a pre-processing step for the template matching.

The thresholding method for detections found by NCC is the most critical part for the whole system, as it can lead to wrong segmentations despite clear results after the template matching. Sometimes, small changes in the investigated SDM can have a



(a) *Madonna - Like a Virgin*



(b) *The Beatles - It Won't Be Long*

Figure 6: *Comparison of automatically detected segmentations (D) and their corresponding reference segmentations (R)*

strong influence on the threshold and the resulting detection. Although the proposed thresholding vector $\vec{t}(j)$ performs satisfactorily for our data set, methods incorporating other system parameters (e.g. embedding dimension, template width) could help to improve the overall stability of the system.

Future work could also be focused on a multi scale approach of the presented algorithm, allowing segmentations on different time scales. Namely, the combination of different embedding dimensions as well as different template widths could help to avoid typical cases of over-segmentation as mentioned in 3.2.

5. REFERENCES

[1] M. Levy and M. Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 318–326, 2008.

[2] E. Gómez, B.S. Ong, and P. Herrera, "Automatic tonal analysis from music summaries for version identification," *Proceedings of the 121st AES Convention*, 2006.

[3] M. Levy, M. Sandler and M. Casey, "Extraction of high-level musical structure from audio data and its application to thumbnail generation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006, vol. 5, pp. 13–16.

[4] M.A. Bartsch and G.H. Wakefield, "Audio thumbnailing of popular music using chroma-based representations," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 96–104, 2005.

[5] T. Zhang and R. Samadani, "Automatic generation of music thumbnails," in *IEEE International Conference on Multimedia and Expo*, 2007, pp. 228–231.

[6] M. Goto, "A chorus section detection method for musical audio signals and its application to a music listening station," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1783–1794, 2006.

[7] G. Boutard, S. Goldszmidt, and G. Peeters, "Browsing inside a music track, the experimentation case study," *Learning the Semantics of Audio Signals*, p. 87, 2006.

[8] G. Peeters, A. La Burthe, and X. Rodet, "Toward automatic music audio summary generation from signal analysis," in *Proceedings of the International Conference on Music Information Retrieval*, 2002, pp. 94–100.

[9] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2000, vol. 1, pp. 452–455.

[10] G. Peeters, "Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR). Vienna, Austria*, 2007.

[11] J. Paulus and A. Klapuri, "Music structure analysis using a probabilistic fitness measure and a greedy search algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1159–1170, 2009.

[12] Alexander Wankhammer, Peter Sciri, and Alois Sontacchi, "Chroma and MFCC based pattern recognition in audio files utilizing hidden Markov models and dynamic programming," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, 2009, pp. 401–407.

[13] A. Eronen and F. Tampere, "Chorus detection with combined use of MFCC and chroma features and image processing filters," in *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-07)*, 2007, pp. 229–236.

[14] S. Abdallah, K. Noland, M. Sandler, M. Casey, and C. Rhodes, "Theory and evaluation of a Bayesian music structure extractor," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2005, pp. 420–425.

[15] J.J. Aucouturier and M. Sandler, "Segmentation of musical signals using hidden Markov models," in *Proceedings of the 110th AES Convention*, 2001.

[16] J.C. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[17] Daniel P.W. Ellis and Graham E. Poliner, "Identifying cover songs with chroma features and dynamic programming beat tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP 2007)*, 2007, vol. 4.

[18] J. Paulus and A. Klapuri, "Music structure analysis by finding repeated parts," in *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*. ACM, 2006, pp. 59–68.

[19] C.L. Webber Jr, "Recurrence quantifications: Feature extractions from recurrence plots," in *Recurrence Plot Workshop*, 2005, p. 42.

[20] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, pp. 285–296, 1975.

[21] J.P. Lewis, "Fast normalized cross-correlation," in *Vision Interface*, 1995, vol. 10, pp. 120–123.

[22] H. Lukashevich, "Towards quantitative measures of evaluating song segmentation," in *Proceedings of the 9th International Conference of Music Information Retrieval (ISMIR 2008)*, 2008, pp. 375–380.