MUSIC STRUCTURE ANALYSIS WITH A PROBABILISTIC FITNESS FUNCTION IN MIREX2009

Jouni Paulus and Anssi Klapuri
Department of Signal Processing
Tampere University of Technology
jouni.paulus@tut.fi, anssi.klapuri@tut.fi

ABSTRACT

This paper describes the method we submitted for the "Structural Segmentation" task at MIREX2009. The method defines a fitness function for structural descriptions based on the idea that all occurrences of a musical part should be acoustically similar and differ from the occurrences of other parts. The method creates a large set of potential segments, estimates the probability of each two segments to be occurrences of the same part, and uses these probabilities in a fitness function. The function optimisation is done with a greedy search algorithm.

1. INTRODUCTION

Music piece structure analysis refers to the task of providing a temporal segmentation of the piece into occurrences of musical parts, such as "chorus" and "verse", and grouping of the occurrences of a part. This kind of a analysis is meaningful on pieces having a sectional form. An occurrence of a musical part is often 20–30 s in length and may be repeated later in the piece.

Various methods for music structure analysis have been proposed in the literature, for an overview of the basic principles refer to [1]. The main method categorisation provided in [2] divides the methods into "state" and "sequence" approaches. The former considers the piece to be produced by a state machine, while the latter assumes that the piece contains repeated sequences of musical events. The method proposed in this paper belongs into the "state" category, or it can be considered to belong to a third category: "fitness" function based approaches.

The submitted method uses three acoustic features describing different aspects of the piece, creates several potential segmentations, matches each segment pair with two distance measures, and produces probabilities for the two segments to be occurrences of the same part. The probabilities are used in a fitness function for descriptions of the piece structure, and a greedy search algorithm is employed for the function optimisation. For more details, see [3].

2. METHOD DESCRIPTION

The method starts by estimating a musical beat grid with the method from [4]. The reliability of the estimation is improved by a two-pass scheme: first, only a 20 s excerpt is analysed. The produced period estimate is then used to sharpen the prior distribution of beat length by setting the Gaussian distribution mean to the estimated period value and halving the original variance parameter value. Then the entire signal is analysed. Still, the resulting beat grid may have π-phase errors. The effect of this is reduced by halving the period, producing a half-beat grid.

Raw acoustic feature extraction is done from 4096 sample frames with 50% overlap. 13 mel-frequency cepstral coefficients (MFCCs) from the output of a 42-band triangular mel-scaled filter bank are calculated and the lowest coefficient is discarded. The second acoustic feature used is chroma, which is calculated with the method described in [5]. It estimates the saliences of different fundamental frequencies in the range 80–640 Hz, resamples the frequency scale to a semitone scale by retaining only the maximum salience in each semitone range, and finally produces the chroma by octave folding. The features are then temporally resampled to the beat-synchronised grid.

The acoustic features are then focused on two time scales by Hanning window weighted median filtering. The finer time-scale features are obtained by skipping the filtering, while the coarser time-scale features are obtained with 33 and 65 frame filtering windows for MFCCs and chroma respectively. In addition to MFCCs and chroma, a third acoustic feature, rhythmogram [6], is calculated. The calculation uses the onset detection accent function produced by the beat estimation instead of the perceptual spectral flux proposed in the original publication. The feature itself is simply the autocorrelation of the accent function calculated in sliding windows of 33 half-beat frames in length. All the features are finally normalised to zero mean and unity variance over time.

From the five acoustic features (MFCC and chroma on two temporal scales, and rhythmogram), separate self-distance matrices (SDMs) are calculated using cosine distance measure. A set of candidate segmentation points is generated with novelty vector calculation [7]. A Gaussian tapered 40×40 checkerboard kernel matrix is correlated along the main diagonals of the SDMs and the resulting novelty vectors are summed. Maximum of 30 highest local maxima

This work was supported by the Academy of Finland, (application number 129657, Finnish Programme for Centres of Excellence in Research 2006–2011).

with the minimum distance of 12 frames are then located from the summed novelty vector and they are considered to be potential segmentation points.

All possible segments between two candidate segmentation points are created, and all non-overlapping segment pairs are matched with two distance measures: stripe and block. The stripe distance is calculated from the short time-scale SDMs, while the block distance is calculated from the coarser time-scale SDMs. The stripe distance between two segments is the dynamic time warping path cost through the SDM submatrix the two segments define, normalised by the length of the longer segment. The block distance is the average element value in the submatrix.

The distance values $d\left(s_i, s_j\right)$ between two segments s_i and s_j are transformed into probabilities that the two segments are occurrences of the same musical part by applying a sigmoidal warping function

$$p\left(g(s_i) = g(s_j)\right) = \left(1 + \exp(z_1 d\left(s_i, s_j\right) + z_0)\right)^{-1}. \tag{1}$$

The sigmoid parameters z_1 and z_0 are calculated from training material (the data set *TUTstructure07* was used here).. The probabilities are then combined with geometric mean, and a constraint prohibiting pairs with segments differring more than by factor 1.2 in length is applied. The final fitness function to be optimised is

$$P(\mathbb{E}) = \sum_{s_i \in \mathbb{S}} \sum_{s_j \in \mathbb{S}} W\left(s_i, s_j\right) l(s_i, s_j, g), \tag{2}$$

where

$$l(s_i,s_j,g) = \tag{3}$$
$$\begin{cases} \log\left(\hat{p}\left(g(s_i) = g(s_j)\right)\right), & \text{if } g(s_i) = g(s_j) \\ \log\left(1 - \hat{p}\left(g(s_i) = g(s_j)\right)\right), & \text{if } g(s_i) \neq g(s_j) \end{cases}.$$

In the equations above, $\mathbb{E}$ is the candidate structure description, $\mathbb{S}$ the set of segments s_i in the description, $g(s_i)$ is the group (musical part) in which the segment s_i is assigned into, and $W\left(s_i, s_j\right)$ is the number of elements in the submatrix defined by the two segments.

The optimisation of (2) is done with a greedy algorithm proposed in [3]. A directed acyclic graph (DAG) is formed by replicating each segment with a possible group assignment to create the nodes. There is an edge between two nodes if the two segments are temporally directly consecutive. The task is to find the optimal path through the DAG. It is fulfilled by a search which starts by inserting a single token to the start node. At each iteration, the β best tokens in each node are propagated to the following nodes and their partial fitnesses are updated accordingly. Then, the number of tokens in each node is reduced to α by discarding the less fit tokens. The tokens arriving to the final node contain a path through the DAG, and the path encodes a segmentation and a grouping of the segments. The search can be iterated until all tokens have reached the final node, or some convergence condition is met. The implementation uses the values $\beta = 10$ and $\alpha = 50$, and determines convergence if the best description has not changed in 10 iterations.

	ANO	ANO2	GP	MND	PK
S_O(%)	63.7	65.4	60.1	73.9	59.3
S_U(%)	63.7	57.5	67.7	61.8	79.0
F_{pair}(%)	58.2	57.7	53.3	60.0	54.0
P_{pair}(%)	59.7	54.3	62.7	56.1	74.1
R_{pair}(%)	61.4	67.0	50.5	71.0	46.2
Rand(%)	76.2	73.5	75.9	74.8	79.2
F_{B}@0.5s (%)	18.3	12.8	18.4	21.0	27.1
P_{B}@0.5s (%)	16.0	12.5	14.6	15.8	24.3
R_{B}@0.5s (%)	22.2	13.5	26.0	36.0	32.3
F_{B}@3s (%)	59.0	58.4	50.0	39.9	53.1
P_{B}@3s (%)	51.5	57.5	40.0	29.9	47.7
R_{B}@3s (%)	71.4	61.5	69.8	69.2	63.2
Δ_{T2C}(s)	1.63	2.47	2.11	2.23	2.44
Δ_{C2T}(s)	3.55	3.21	3.52	3.44	3.51

Table 1. Evaluation results for all methods. The described method is denoted with “PK”. See text for more details.

3. IMPLEMENTATION ISSUES

The results reported earlier in [3] were obtained with a mixed Matlab and C++ implementation. The submitted method is implemented as a command line executable completely in C++, and therefore the results may differ from the ones reported earlier. The analysis time corresponds approximately to half the length of the input signal when run on a single core of a 1.86 GHz Intel Core2 CPU. The time consumption inside the method is divided approximately to 35% for acoustic analysis, 50% for the segment generation and matching, and the remaining 15% for the search. The complexity of the search can be controlled with the parameters β and α, but they also affect memory consumption and extent of the search.

4. RESULTS

The evaluation results of all methods submitted to the task are summarised in Table 1. [1] The method described in this paper is denoted with “PK” in the results. The different evaluation measures are denoted as follows

- S_O, S_U: over- and under-segmentation scores proposed in [8].
- F_{pair}, P_{pair}, R_{pair}: frame pair clustering measure, used in [9].
- Rand: Rand clustering index described in [10].
- F_{B}, P_{B}, R_{B}: segment boundary retrieval score with different allowed deviations.
- Δ_{T2C}, Δ_{C2T}: median time difference from annotated segment boundaries to found ones, and vice versa.

[1] The results originate from `http://www.music-ir.org/mirex/2009/index.php/Music_Structure_Segmentation_Results`.

The results suggest that the proposed method has some problems with over segmenting the result. This may show either as providing the description on a finer time-scale than the ground-truth, or as not locating the repeats of a part. The former problem could be alleviated by increasing the length of the feature median filtering window for the "block" features, as well as increasing the relative weight of the "block" information compared to the "stripe" information. The latter problem, which is more likely cause for the imbalanced performance, could be alleviated by adjusting the sigmoidal mapping function to produce larger probabilities. These issues should be addressed in the future work on the system.

5. REFERENCES

[1] Roger B. Dannenberg and Masataka Goto. Music structure analysis from acoustic signals. In David Havelock, Sonoko Kuwano, and Michael Vorländer, editors, *Handbook of Signal Processing in Acoustics*, volume 1, pages 305–331. Springer, New York, N.Y., USA, 2008.

[2] Geoffroy Peeters. Deriving musical structure from signal analysis for music audio summary generation: "sequence" and "state" approach. In *Lecture Notes in Computer Science*, volume 2771, pages 143–166. Springer-Verlag, 2004.

[3] Jouni Paulus and Anssi Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1159–1170, August 2009.

[4] Anssi Klapuri, Antti Eronen, and Jaakko Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, January 2006.

[5] Matti P. Ryynänen and Anssi P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008.

[6] Kristoffer Jensen. Multiple scale music segmentation using rhythm, timbre, and harmony. *EURASIP Journal on Advances in Signal Processing*, 2007. Article ID 73205, 11 pages.

[7] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Proc. of IEEE International Conference on Multimedia and Expo*, pages 452–455, New York, N.Y., USA, August 2000.

[8] Hanna Lukashevich. Towards quantitative measures of evaluating song segmentation. In *Proc. of 9th International Conference on Music Information Retrieval*, pages 375–380, Philadelphia, Pa., USA, September 2008.

[9] Mark Levy and Mark Sandler. Structural segmentation of musical audio by constrained clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):318–326, February 2008.

[10] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985.