PROJECT PROTOCOL "SONG STRUCTURE ANALYSIS"

Sebastian Herrmann, Schokri Ben Mustapha, Philipp Sehling (Group 1)

Technische Universität Berlin

ABSTRACT

Audio or music segmentation embodies one of the typical music information retrieval (MIR) topics. It is already a well-researched field, analysed and tested through diverse approaches: Different features can be used, several parameter sets and optimizations are proposed for the single processing steps and various segmentation and clustering approaches coexist. The challenge of this task lies in the intelligent combination of existing analysis methods and statistical procedures to achieve results that are as independent from the source material as possible to guarantee a flexible chain of algorithms and robust results.

In order to familiarize ourselves with the methods and possibilities involved in this task, we implemented a set of algorithms that emerged as good or best practices during the course of our research. The latter include transforming the given audio signal into Mel-frequency cepstral coefficients (MFCCs), calculating a feature similarity matrix, its novelty curve and peak picking to result in segment borders. A clustering procedure will then group and label the found segments and compare our results with a dataset.

Index Terms— Music, Structure, Segmentation

1. INTRODUCTION

Historically, the analysis of music form or structure is a key part of music theory and musicology[1]. Although not always clearly recognizable most music has an underlying structure and hierarchy. Beats become rhythms, notes become melodies which become phrases and figures. Through the use of repetition and variation these establish the form of a piece of music. Next to the traditional method of manual music analysis exists the automatic approach using tools from the science of music information retrieval (MIR). MIR is a very interdisciplinary task with participants from the fields of computer science, cognitive science, audio engineering, digital sound processing and, of course, musicology and music theory amongst others[2]. The simplest type of structure representation is based on alphabetically naming and indexing parts. A typical appearance example would be "$A_1B_1A_2CA_3B_2A_4$" for the well-known rondo form used for instance by Bach, Schubert or Beethoven[3]. The Letters represent repeating parts while the indices explain the number of repetition. Therefore A_2 represents the second repetition of the first part, which is labelled A.

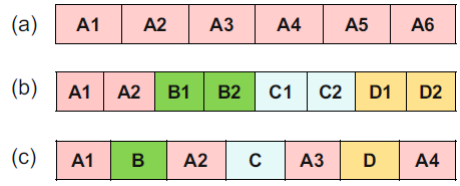

Fig. 1. Examples for musical structures in Western music[4, p. 173]. a) Strophic form. b) Chain form with repetition. c) Rondo form.

Knowing song structure is especially useful in Manoeuvring within songs or music databases. It can improve music recommendation systems and for example enable skipping directly to a chorus for previewing situations. From an educational point of view it can give quick and understandable insights into the structural organization of a piece. Noteworthy are also creative possibilities considering re-mixes based on structural information. From an educational point of view it can give quick insights into the structural organization of a piece[5].

The remainder of the paper will show several related works, explain in detail the method we used to gather relevant audio features and the approach we used to gain structure information out of the self similarity matrix. Following that we will evaluate the results and discuss problems and possible solutions for future work. A brief conclusion will then be the end of this paper.

2. RELATED WORK

Since the beginning of research on automatic structure analysis different ways have been developed to process the audio signal. According to Paulus et al [6] it is possible to sort them in three different categories organized by their clustering approach:

1. Repetition-based Identification of recurring parts.
2. Novelty-based Detection of transitions between contrasting parts.

3. Homogeneity-based Identification of consistent parts.

In finding musical structure and underlying boundary's the human brain reacts on key features as change in timbre and repetition as well as changes in dynamics[7]. To work in a similar way the raw audio data is first being analysed for several audio features as seen in figure 2. Well established is the extraction of Mel-Frequency-Cepstral-Coefficents (MFCCs), Pitch Chroma and Tempogram. Also several other scalar features might be used, to varying success. As this process is usually done in a block wise way, the amount of audio information gets reduced which allows for a better overview over the structure and more resourceful calculations. Following is always the generation of self similarity matrix (SSM) or self distance matrix(fig.3) which gives good visual representation of structural information[8].

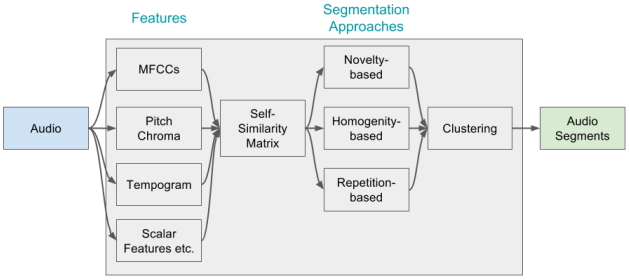

Fig. 2. Main methods to automatically extract a music structure

Several clustering propositions have been developed and proven their functionality. Foote[9] uses a novelty score to extract boundaries from music sources applying a checkerboard kernel to a self similarity matrix. An example for chroma based work is found in [10] using note information for the analysis. Aided by constant Q transformation accurate melody similarity could be achieved. Constant Q transformation was also used by Wang et al.[11] to extract musical note information. In this case an adaptive threshold method extracts significant repeating patterns from the self similarity matrix. In contrast to the former works Serrà[12] describes a way of generating structural analysis from time series features by emulating the short-time human memory using temporal lag information and bivariate probability density with Gaussian kernels.

3. ALGORITHM OVERVIEW & DESCRIPTION

After receiving an audio file as an initial input for our algorithm, the first processing step involves simple feature retrieval. We decided to focus on the MFCCs as our initial and only audio feature which is a widely recommended procedure for audio segmentation purposes according to Turnbull et al [13, p. 4] and Foote [8, p. 2]. We also run tests with chroma-based features like pitch chroma, Chroma Energy Normalized Statistics (CENS) and Chroma DCT-Reduced log Pitch (CRP).[14] However, those tests resulted in less precise results compared to MFCCs. To extract the MFCCs we rely on the MatLab function *ComputeFeature* provided by Lerch [15].

After retrieving a feature vector for the specified audio material we calculate a similarity matrix (SM) via vector correlation, as proposed by Foote [8, p. 3] and illustrated in the equation shown below.

$$S_w(i,j) = \frac{1}{w}\sum_{k=0}^{w-1}(v_{i+k} \cdot v_{j+k}) \tag{1}$$

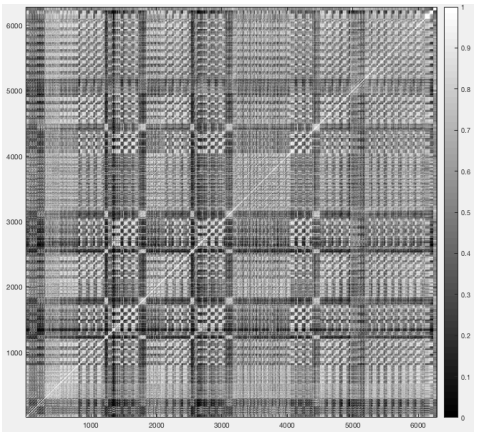

Fig. 3. Similarity matrix produced for the song "Suds & Soda" by Deus that visualizes repeating segments as chessboard-like patterns.

In order to retrieve the segment boundaries from the SM we first produce a novelty curve through correlating a checkerboard kernel over the main diagonal of our SM. A kernel size of around 32 proved to be most successful during our tests. The algorithm and checkerboard structure we used were both introduced by Foote [9, p. 2f.]. By finding the peaks of our novelty curve we can now determine a first set of segment boundaries. MatLab offers an adaptable function *findpeaks* that can be used to achieve this. [16] Its visual output is shown in comparison to the segment boundaries in our dataset in figure 4.

We generally decided to favor the state approach over the sequence approach for our analysis because the former is less computationally intensive and more robust for a variety of signals. [5, p. 21f.]

The retrieved segments from the noveltypeak approach will be used to compute a segment-indexed similarity matrix based on statistics of the spectral data in each segment [17]. To cluster the segments we compute the Singular Value Decomposition (SVD) of the segment-indexed similarity matrix.

Therefore, we calculate the mean μ_i and the covariance matrix Σ_i in each segment and obtain a statistical represen-

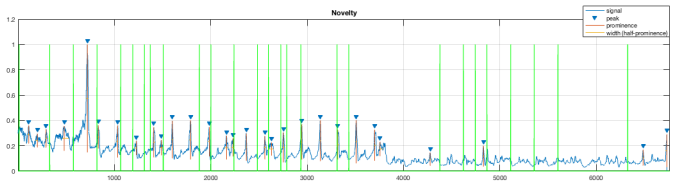

Fig. 4. Novelty curve produced for the song "Suds & Soda" by Deus. The highlighted peaks indicate a first assumption of segment borders, the green lines show the dataset segment borders.

tation by its Gaussian densities $\mathbb{G}_i$. A symmetric variation of the Kullback-Leibler (KL) distance is chosen to measure similarity:

$$\begin{aligned}\hat{d}_{KL}(\mathbb{G}_i \| \mathbb{G}_j) = &\frac{1}{2}[Tr(\Sigma_i \Sigma_j^{-1}) + Tr(\Sigma_j \Sigma_i^{-1}) \\ &(\mu_i - \mu_j)^T (\Sigma_i^{-1} + \Sigma_j^{-1}) \\ &(\mu_i - \mu_j)] - B\end{aligned} \tag{2}$$

Thus, inter-segment similarity between segments p_i and p_j is:

$$d_{seg}(p_i, p_j) = \exp(-\hat{d}_{KL}(\mathbb{G}(\mu_i, \Sigma_i) \| \mathbb{G}(\mu_i, \Sigma_i))) \tag{3}$$

The pairwise distance measure is embedded in the segment-indexed similarity matrix S_S:

$$S_S(i,j) = d_{seg}(p_i, p_j), \qquad i,j = 1, ..., P \tag{4}$$

We then decompose S_S into a matrix-sum using the SVD:

$$S_S(i,j) = \sum_{p=1}^{P} \lambda_p U(i,p) V(j,p) \tag{5}$$

$$= \sum_{p=1}^{P} B_p(i,j) \tag{6}$$

The sum terms B_p are unit norm matrices scaled by decreasing singular values λ_p and ordered by the amount of structure in S_S for which they account. We calculate the sum of the rows of B_p to get a set of vectors $b_p(j)$ which indicate the similarity of segment j to all the segments in the p^{th} segment cluster.

$$b_p(j) = \sum_{p=1}^{P} B_p(i,j), \qquad i,j = 1, ..., P. \tag{7}$$

We then associate the jth segment with the pth cluster such that

$$c = ArgMax(b_p(j)), \qquad p = 1, ..., P \tag{8}$$

The final result of our algorithm mainly consists of three outputs. A list of all processed songs contains details for each of them, such as audio information, the matched dataset, the detected peaks, the retrieved clusters and final evaluation results (variable *songs*). This way the procedure can be related to each song and allows comparisons and benchmarks depending on the initial audio input. The variable *statistics* delivers an overview of the six evaluation criteria for each song, once again enabling the user to compare the results on a song basis. Finally, the variable *overall* summarizes the evaluation results of each song by averaging their values individually.

The more we optimized the algorithm and its many parameters for one type of music, the less generically usable its naturally becomes. Especially the initial segment boundary detection is highly dependent on the prior processing steps and the underlying configuration (e.g. options for the *findpeaks* function). So one challenge was to find a compromise between having a precise boundary detection and keeping up this precision among the whole (relatively diverse) dataset.

4. EVALUATION

Ground truth and methodology

Our evaluation will follow the proposal of Lukashevich at the ISMIR conference 2008(Zitat). To compare the results, we chose the dataset by Levy et al [18] as our ground truth focusing on songs from *The Beatles*, expecting that clearer segment boundaries such as in pop music will be easier to match. In a first step we will evaluate the boundary retrieval of our algorithm. Based on the correcty detected segment boundaries we will evaluate the frame clustering.

Boundary retrieval

Of two possible time constraints for a successful boundary declaration we decided in favor of Levy & Sandler[1] offering a 3 second tolerance from a border in the ground truth which is also used in[19]. At this point it is also noteworthy to mention that a ground truth of music structure is difficult to determine due to individual interpretations of musical events (repetitions and variations) [5, p. 22], [6, p. 634]. Based on the matching boundary segments, precision rate , recall rate and F-measure are calculated.

Frame clustering

Like previous works on that matter, our evaluation won't care about true labels such as *Chorus,Verse* etc., meaning that we will evaluate clustering by measuring the quality of detecting repeated segments. Therefor we will calculate precision rate , recall rate and F-measure.

Results

Table 1 shows the results we retrieved after analysing all songs to be found in the dataset. Compared to similar research results by Jun et al and Levy and Sandler, our final average precision, recall and f-measure values appear very comparable. However, it must be noted that especially the evaluation metrics regarding the segment labeling are still on a low level. This can be related to a relatively high amount of missing values or values that are invalid due to faulty calculations. A lot of potential can be seen in improving the stability of the final evaluation procedure in our algorithm.

Table 1. Precision and recall for several algorithms.

Method	Precision	Recall	F-Measure
Jun et al[19]	72.12%	39.36%	–
Levy and Sandler[1]	56.00%	54.10%	54.00%
Our results (boundaries)	40.53%	74.07%	52.40%
Our results (labeling)	15.32%	11.87%	13.38%

The calculations for precision (9), recall (10) and f-values (11) follow the explanations in Levy et al [1].

$$P = \frac{|P_E \cap P_A|}{P_E} \tag{9}$$

$$R = \frac{|P_E \cap P_A|}{P_A} \tag{10}$$

$$F = \frac{2PR}{P+R} \tag{11}$$

5. CONCLUSION

Equipped only with basic MatLab knowledge and some background on statistical methods we approached the challenge of automated audio segmentation. Although this task is common among MIR applications there is no apparent and established methodology but instead we learned about a big variety of possible procedures and various suggestions from different researchers. After weeks of experimenting with algorithms, third-party libraries and parameter sets we built a relatively stable application that is capable of analysing an array of songs and retrieving their top-level structure in a simple manner. During the execution phase we decided that an even more sophisticated clustering and labeling would exceed the scope of this project and we therefore limited our end result to audio segments that are being differentiated but not clearly labeled in a classical way, e.g. as *A* and *B* instead of *verse* and *chorus*. The comparison of our results with the dataset shows that our results are only partially on a par with comparable research projects like the one from Levy[1]. While the boundary detection at least produces analysable results for all songs, the labeling evaluation proved to be less stable.

Therefore the complexity and difficulty of the task at hand has clearly been experienced, as well as a deeper understanding of the many underlying challenges.

Many ideas surfaced regarding the future potential of our current project state. Especially the topic of adaptability of input parameters appears very interesting since the manual tests showed that little changes to parameters like the kernel size have big impact on the quality of the outcome. Considering songs with highly varying tempi the algorithm's frame and hop size could also adapt to the onset times of the audio material and would thus improve the precision of the segment border detection. Another way of increasing the robustness of the analysis would be to choose different audio features depending on the musical style (which could be identified in a pre-processing step or using the meta data of the sound file).

6. 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, Feb 2008.

[2] Frans Wiering, *Can Humans Benefit from Music Information Retrieval?*, pp. 82–94, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[3] Tom Service, "Stuck on repeat: why we love repetition in music," apr 2016.

[4] Meinard Müller, *Fundamentals of Music Processing*, Springer International Publishing, 2015.

[5] Geoffroy Peeters, "Deriving musical structures from signal analysis for music audio summary generation: "sequence" and "state" approach," 2004.

[6] Jouni Paulus, Meinard Müller, and Anssi Klapuri, "Audio-based music structure analysis," in *11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[7] Michael J. Bruderer, Martin McKinney, and Armin Kohlrausch, "Structural boundary perception in popular music," in *7th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006.

[8] Jonathan Foote, "Visualizing music and audio using self-similarity," in *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*. 1999, MULTIMEDIA '99, pp. 77–80, ACM.

[9] Jonathan Foote, "Automatic audio segmentation using a measure of audio novelty," 2009.

[10] Hong-Jiang Zhang Lie Lu, Muyuan Wang, "Repeating pattern discovery and structure analysis from acoustic

music data," October 2004, Association for Computing Machinery, Inc.

[11] Muyuan Wang, Lie Lu, and Hong-Jiang Zhang, "Repeating pattern discovery from acoustic musical signals," in *Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on*, June 2004, vol. 3, pp. 2019–2022 Vol.3.

[12] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Lluis Arcos, "Unsupervised detection of music boundaries by time series structure features," 2012.

[13] Douglas Turnbull, Gert Lanckriet, Elias Pampalk, and Masataka Goto, "A Supervised Approach For Detecting Boundaries In Music Using Difference Features And Boosting," in *Austrian Computer Society*, 2007.

[14] Meinard Müller and Sebastian Ewert, "Chroma toolbox: MATLAB implementations for extracting variants of chroma-based audio features," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, 2011.

[15] Alexander Lerch, *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*, Wiley-IEEE Press, 2012.

[16] MathWorks, Inc., "findpeaks - find local maxima," 2016.

[17] Jonathan T. Foote and Matthew L. Cooper, "Media segmentation using self-similarity decomposition," 2002.

[18] M. Levy, K. Noland, and G. Peeters, "Reference structural segmentations," aug 2016.

[19] Sanghoon Jun, Seungmin Rho, and Eenjun Hwang, "Music structure analysis using self-similarity matrix and two-stage categorization," *Multimedia Tools and Applications*, vol. 74, no. 1, pp. 287–302, 2015.