MUSIC BOUNDARY DETECTION BASED ON A HYBRID DEEP MODEL OF NOVELTY, HOMOGENEITY, REPETITION AND DURATION

Akira Maezawa

Yamaha Corporation

ABSTRACT

Current state-of-the-art music boundary detection methods use local features for boundary detection, but such an approach fails to explicitly incorporate the statistical properties of the detected segments. This paper presents a music boundary detection method that simultaneously considers a fitness measure based on the boundary posterior probability, the likelihood of the segmentation duration sequence, and the acoustic consistency within a segment. Evaluation shows that our method improves segmentation $F_{0.58}$-measure by about 10 points compared to DNN with peak-picking, a popular scheme used in the state-of-the-art music boundary detectors.

***Index Terms*—** music information retrieval, music boundary detection, deep learning, duration model

1. INTRODUCTION

Music boundary detection is the task of detecting, in an audio signal, edges of music structure such as "verse" and "chorus." It is an important problem in music information retrieval, such as for visualization or summarization of music [1].

Current methods that are competitive with the state-of-the-art music boundary detectors [2] are potentially prone to emitting inconsistent segmentations. Such inconsistencies occur because they detect each boundary inside an audio signal, *independently* of the properties of the segmentations. Thus, for example, in the first verse of a song, the verse and chorus may be segmented whereas in the second verse, the verse and chorus may not be segmented.

This kind of failure mode is easily detected by taking into account long-term segment durations and homogeneity within a structure segment. For example, suppose an estimated sequence of segment duration measure lengths is given as $(16, 16, 8, 16, 4, 4, 4, 4, 8)$. Since music structure has a unique "language" of segment durations, one would suspect that either the four successive 4's is one segment of 16 measures or segments of length 16 is actually four successive segments of 4 measures. Furthermore, the choice can be narrowed down by assessing whether a segment remains more homogeneous by merging successive 4's into one segment, or splitting 16's into four successive 4's.

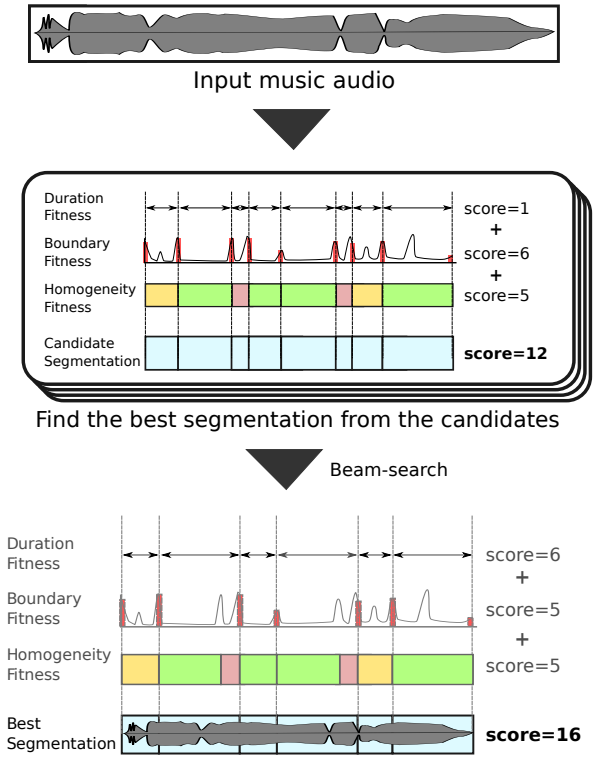

Fig. 1: Overview of our method.

In this paper, we propose a music boundary detection based on this kind of hybrid sources of hypotheses, by extracting a segmentation that simultaneously optimizes a fitness measure of the segmentation inside a song, comprising of (1) structure segment boundaries, (2) sequence of segment durations, and (3) homogeneity of timbre within a segment.

Our contributions are as follows: (1) we present a way to combine deep neural network (DNN)-based boundary detector, various segment duration models and segment homogeneity model using beam-search, (2) we propose various higher-order duration models for modeling segment durations (in beats) to show that more expressive model does contribute to improved performance, and (3) we show that incorporating fitness measure of duration and timbre homogeneity contributes to improved segmentation quality, primarily by preventing over-segmentation.

2. RELATED WORK

Music boundary detection is tackled by exploiting (1) novelty of acoustic features on boundaries, (2) homogeneity of timbral features inside a segment, and (3) repetition of features [3]. Most methods tackle one of these characteristics, or a hybrid [4]. Recent methods explicitly find the repetitive structure using matrix factorization [5], or employ deep neural networks [6, 2] to directly estimate the boundaries based on local acoustic features and long-term similarity matrix.

Music structure is not only homogeneous, repetitive, and has novel boundaries, but also has a predictable segment duration sequence. Indeed, a unigram or bigram duration model is often used in music boundary detection [7].

3. METHOD

In this paper, we develop a hybrid scheme like [4] based on novelty, repetition, homogeneity, and duration. The former three are attained implicitly using a deep neural network boundary detector, inspired by the recent state-of-the-art [2]. We also explicitly express homogeneity through a dedicated model that checks for timbre variance within each segment. Furthermore we incorporate various forms of duration models and compare the segmentation performance with different model expressiveness.

Let us formulate the boundary segmentation problem. We denote a possible segmentation by an ordered set $\mathbf{B} = (b_1, b_2 \cdots b_k)$, which means that boundaries occur at beats $b_1 < b_2 \cdots < b_k$. There are two special beat positions: beat 0 is the beginning of the audio data and beat b_M is the end of the audio data. We formulate boundary detection as an estimation of a segmentation $\mathbf{B}'$ that maximizes a fitness function $f(\mathbf{B})$ over the space of all possible segmentation, *i.e.*, $\mathbf{B}' = \arg\max_{\mathbf{B}} f(\mathbf{B})$. As will be discussed in Section 3.4, we will find $\mathbf{B}'$ by successively appending segment boundaries to a set of W candidate segmentations $\{\mathbf{B}_w\}_{w=1}^{W}$ until beat b_M is added.

The fitness $f(\mathbf{B})$ incorporates three fitness measures f_B, f_D and f_H, and is given as follows:

$$f(\mathbf{B}) = f_B(\mathbf{B}) + \lambda f_D(\mathbf{B}) + \nu f_H(\mathbf{B}), \tag{1}$$

where $\lambda > 0$ and $\nu > 0$ are the weights of each fitness. The fitness $f_B(\mathbf{B})$ denotes the log-probability of each beat in $\mathbf{B}$ to be the boundary given the audio signal, obtained from a novelty curve derived from repetition and local spectral features. It encourages boundaries to be placed where spectral feature repeats and changes drastically. The fitness $f_D(\mathbf{B})$ denotes the log-probability of the segment beat duration sequence. It encourages the segment beat durations to be consistent with respect to each other. The fitness $f_H(\mathbf{B})$ denotes the log-probability of the segmentation based on homogeneity of timbre within each segment. It encourages splitting of segments whose timbre vary significantly.

For computing the fitness functions, we first extract the beat times using [8]. For each beat b and its neighboring half-beats, we extract 128-dimensional mel-scale log spectrum (MSLS) from 0 to 8 kHz, and tatum-level self-similarity matrix (SSM), where tatum is defined as half-beat. SSM is computed by taking the inner product of the MSLS at the beats in concern, and at beat b, the similarity is computed for $b \pm 200$ beats, and the rest set to 0. The SSM is rotated so that at every beat, the center element is the similarity to itself. We denote the tuple of MSLS and SSM at beat b concatenated with that at one tatum after as $X(b)$. Each frequency bin of MSLS is normalized to zero mean and unit variance, and the SSM is normalized to zero mean and unit variance.

3.1. Criteria 1. Boundary fitness

For a candidate segmentation $\mathbf{B} = \{b_1, b_2 \cdots b_N\}$, f_B is computed as follows:

$$f_B(\mathbf{B}) = \sum_n \log(p_B(X(b_n))) - \log(1 - p_B(X(b_n))). \tag{2}$$

$p_B(X)$ is the posterior probability that MSLS/SSM pair X corresponds to a music structure boundary.

This fitness function denotes the increment of the log probability for a given segmentation compared to having no boundary at all, assuming $p_B(X(b_n))$ is independent. To see why, since the log probability of having no boundary at all is $\sum_{b=0}^{b_M} \log(1 - p_B(X(b)))$, the first term on the r.h.s. of Eq. 2 adds the log-probability of having a label, and the second term undoes the log probability of having no label.

Inspired by state-of-the-art music boundary detector [2], $p(X)$ is modeled by a DNN, where MSLS and SSM over radius of 16 beats each goes through a convolutional layer of 16 channels with kernel size of (3×6) with the first axis being the temporal axis, batch-norm and max-pooling of (1×6) and another convolutional layer of 32 channels with kernel size (3×3), with batch-norm and leaky ReLU activation. Outputs of the CNN for MSLS and SSM are then concatenated and connected to a fully-connected layer of 1024 output units with leaky ReLU activation, followed by a fully-connected layer with 1 output unit with sigmoid activation. In contrast to the existing work which used uniformly sampled features, we use beat-synchronous features.

The network is trained using ADAM [9] to minimize the cross-entropy between the output and the ground-truth.

3.2. Criteria 2. Segment duration fitness

Let $L_n = b_{n+1} - b_n$ be the number of beats used by segment n. Then, we model the fitness function f_D as the following:

$$f_D(\mathbf{B}) = \sum_n^{N-1} \log(p_D(L_n | L_{1 \cdots n-1})). \tag{3}$$

$p_D(L_n|L_{1\cdots n-1})$ is the probability of observing a segment duration of L_n beats, given the previous segment durations. We call this the duration language model (LM).

There are many possibilities for the LM in addition to existing models like unigram and bigram. This paper specifically explores two more variants: LSTM and N-gram LM (N>2). *LSTM LM* is a long-short term memory (LSTM [10]) with 1024 output units, which is trained to predict the segment beat duration L_n given past segment durations, $p_D(L_n|L_{1\cdots n-1})$. Because of the recurrent connection, it remembers the previous segment durations. The beat duration is expressed as a one-hot vector. The input duration, before being fed to the LSTM is reduced to 64 dimensions by a fully-connected network with tanh activation. The LSTM output is converted to one-hot representation of beat duration by a fully-connected network, with leaky ReLU activation. The model is trained to minimize the cross-entropy loss. *N-gram LM* predicts the next beat duration using a categorical distribution defined for all permutations of $N-1$ previous beat durations, with Katz backoff [11]. The LM is trained using maximum likelihood estimation and smoothed by assigning log-likelihood of -100 for an unseen sequence.

In either case, we treat two edge cases differently: the beginning and the end. The first structural segment could occur at any beat b because arbitrary silence may precede the beginning of the song. Thus, we assume the beat position of the initial segment boundary is exponentially distributed with scale parameter α:

$$f_d((0,b)) = -\alpha b. \tag{4}$$

Similarly, the last beat b_M does not necessarily have a meaningful structural boundary. Thus, for the end we marginalize over possible ways of segmenting past the end and let:

$$\begin{aligned} &f_D((b_1 \cdots b_k, b_M)) \\ &= f_D((b_1 \cdots b_k)) + \log\Big(\sum_{b'>b_M} p_D(b'-b_k|L_{1\cdots k})\Big), \end{aligned} \tag{5}$$

where b' in the summation is evaluated up to some reasonable value (we evaluate up to $b_M + 64$ beats).

3.3. Criteria 3. Timbre homogeneity fitness

To encourage segments with similar timbre to be clustered in one segment, we evaluate the inter-segment variance of the MSLS. Let $S(b) \in \mathbb{R}^N$ be the MSLS at beat b smoothed over the frequency axis by a Hanning window. We assume that MSLS inside each segment is Normally distributed with a spherical variance. Then, we set the fitness function proportional to the maximum log-likelihood[1]:

$$f_H(\mathbf{B} = (b_1 \cdots b_k)) = -\sum_{i=1}^{k-1}\sum_{n=1}^{N} \mathrm{Var}(\{S_n(b)\}_{b=b_i}^{b_{i+1}-1}). \tag{6}$$

[1] Strictly speaking, the variance term needs to be weighted by the segment beat duration, but we found that the method performs better without it.

This fitness decreases when there is a high inter-segment variance, so it encourages more segments to be generated.

3.4. Segmentation estimation using beam-search

Finding the best $\mathbf{B}'$ that maximizes the fitness function is infeasible since it requires evaluating the fitness over all possible segmentation. Therefore, we use beam-search to greedily find the boundaries. Specifically, we start with some W candidate segmentations $\{\mathbf{B}_w\}_{w=1}^{W}$, each initialized to an ordered list containing the first beat as the segment boundary, *i.e.*, $\mathbf{B}_w = (0)$. Then, we iterate the following until every candidate segmentation ends with b_M, the end of the song:

1. For each candidate segmentation $\mathbf{B}_w = (b_{1,w}, \cdots b_{k,w})$, append all possible beat segments past $b_{k,w}$. In other words, for each w, generate $\mathbf{B}_{w,b'} = (b_{1,w}, \cdots b_{k,w}, b')$ for $b' \in [b_{k,w}+1, b_M]$.
2. For each $\mathbf{B}_{w,b'}$, evaluate $f(\mathbf{B}_{w,b'})$. For computing f_D term, if $\mathbf{B}_{w,b'}$ is of form $(0, b')$, then use Eq. (4); if $b' = b_M$, then use Eq. (5); otherwise, use Eq. (3).
3. Retain W segmentations with the best score $f(\mathbf{B}_{w,b'})$, and set each to $\mathbf{B}_w$.

Finally, the best segmentation result is used as the estimate.

4. EVALUATION

For the training dataset, the duration LM was trained on 7700 MIDI files with music structure annotation. The DNN boundary detector was trained on synthesized versions of the 7700 MIDI files and another dataset of 410 Japanese popular songs.

For the validation dataset, we used the first album of the Beatles with the Isophonics ground-truth label [12].

For the test dataset, we used three different datasets. First, we used 164 songs from the Beatles and corresponding Isophonics ground-truth label, *excluding* the first album which is used for validation (the result with the first album was not very different). Second, we used the 100 songs from the RWC Popular Music Database [13], and its ground-truth structural labels [14]. Third, we used 468 songs from the Internet Archive data from SALAMI 2.0 dataset [15]. We did not use other songs from SALAMI that are based on commercial songs. The internet archive dataset contains many degraded-quality audio signals, making segmentation and beat detection more difficult compared to the other datasets and other songs in the full SALAMI.

To evaluate the methods, mir_eval [16] was used to compute the boundary segmentation performance, including precision, recall, F_1-score and the more perceptually-relevant $F_{0.58}$-score [17]. The estimated boundary was treated as correct if it was within 0.5 seconds of a ground-truth boundary label. We omitted the ends of the song from the structural boundary, as they artificially improve the performance.

Table 1: F-measure for different duration LM.

Language Model	F_1	λ	ν
None	0.37	0.00	0.01
1-gram	0.37	0.03	0.12
2-gram	0.37	0.02	0.02
3-gram	0.47	0.20	0.40
4-gram	0.46	0.26	0.08
5-gram	0.44	0.50	0.77
LSTM	0.49	0.40	0.62

In the following experiments, the beam width W was set to 20. λ and ν were optimized with Bayesian optimization [18], as to maximize the F_1-score on the validation data.

4.1. Experiment 1 - choice of the duration LM

We compared the performance of different boundary duration LMs. To this end, we prepared and trained boundary detectors with (1) no duration LM, (2) N-gram LM ($N = 1$ to 5), (3) LSTM LM. Each boundary detector was tested on the Beatles dataset, minus the validation dataset.

Table 1 shows the F_1-score of boundary detection, evaluated over different language models, along with the best λ and ν found for each LM. The improvement of F_1-score with an expressive LM demonstrates its effectiveness for improving the segmentation quality. For N-gram LM, the performance boosts after 3-gram. This shows that the benefit of LM comes only with more expressive LMs. LSTM outperforms N-gram because it is capable of handling longer contexts.

The general trend of the weights λ and ν shows that (1) expressive LM is informative, since λ, the importance of duration fitness, increases with an expressive LM, and (2) the two fitnesses encourage different kinds of segmentation and are both important for performance, since ν and λ increase together, instead of one dominating the other.

4.2. Experiment 2 - boundary detection

We compared the performance of different boundary detectors, and tested them on Beatles, RWC-Popular and SALAMI dataset. For the baseline methods, we prepared (1) the music structure analysis framework [19] with its default setting (method SF [20] with PCP features, denoted "MSAF(PCP/CF)") [2], (2) spectral clustering [21] (denoted "Spec. Cluster"[3]), (3) the peak-picking scheme from [2] using the posterior probability of our DNN boundary detector, with the peak-picking threshold optimized on the validation dataset (denoted "DNN Peak-pick").

[2]Obtained from `https://github.com/urinieto/msaf` commit 9dbb57d

[3]Obtained from `https://github.com/bmcfee/laplacian_segmentation` commit 94a2c34

Table 2: Comparison of different detection methods.

Method	Precision	Recall	F_1	$F_{0.58}$
MSAF(PCP/SF)	0.206	0.145	0.165	0.181
Spec. Cluster	0.240	0.336	0.243	0.235
DNN Peak-pick	0.429	**0.511**	0.458	0.441
Proposed	**0.608**	0.447	**0.503**	**0.545**

(a) RWC Popular Music Database.

Method	Precision	Recall	F_1	$F_{0.58}$
MSAF(PCP/SF)	0.235	0.207	0.211	0.220
Spec. Cluster	0.154	0.424	0.203	0.173
DNN Peak-pick	0.368	**0.542**	0.427	0.394
Proposed	**0.507**	0.520	**0.492**	**0.495**

(b) Beatles dataset (w/o the validation data).

Method	Precision	Recall	F_1	$F_{0.58}$
MSAF(PCP/SF)	0.162	0.168	0.157	0.158
Spec. Cluster	0.154	0.232	0.170	0.159
DNN Peak-pick	0.217	**0.386**	0.267	0.239
Proposed	**0.301**	0.347	**0.306**	**0.300**

(c) SALAMI dataset (Internet Archives).

Table 2 shows the results. It shows that incorporating the consistency of segmentation improves the segmentation quality significantly. Compared to the DNN peak-picking baseline, our method boosts by about 5 points the F_1 score, 10 points in the perceptually-relevant $F_{0.58}$-score, both thanks to the significantly improved precision. The performance of the DNN peak-picking baseline is comparable but slightly behind the values reported in the literature; we speculate the discrepancy comes from fewer audio training data.

The results also show that (1) the incorporation of duration constraint serves not so much on introducing new correct labels but removing false positives, and (2) the primary qualitative improvement comes from duration LM instead of timbre homogeneity. To elaborate, since timbral homogeneity serves only to introduce more segments, it tends to increase the recall or decrease precision; therefore, the improved precision can be explained only by the duration fitness.

In summary, DNN boundary detector is already quite good at finding the boundary candidates, so adding more schemes to detect *additional* boundaries helps only marginally; on the other hand, false positives that the DNN detector emits can be filtered out by the duration LM.

5. CONCLUSION

We have presented a music boundary detection method that simultaneous takes into account various measures of fitness of segmentation. Evaluation showed improved boundary detection quality, mostly by improving the precision. Future work includes models for improving the segmentation recall.

6. REFERENCES

[1] Wei Chai and Barry Vercoe, "Music thumbnailing via structural analysis," in *ACM International Conference on Multimedia*, 2003, pp. 223–226.

[2] Thomas Grill and Jan Schluter, "Music boundary detection using neural networks on combined features and two-level annotations," in *International Conference on Music Information Retrieval*, 2015.

[3] Jouni Paulus, Meinard Müller, and Anssi Klapuri, "State of the Art Report: Audio-Based Music Structure Analysis," in *International Conference on Music Information Retrieval*, 2010, pp. 625–636.

[4] 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*, vol. 17, no. 6, pp. 1159, 2009.

[5] Tian Cheng, Jordan B. L. Smith, and Masataka Goto, "Music structure boundary detection and labelling by a deconvolution of path-enhanced self-similarity matrix," in *International Conference on Music Information Retrieval*, 2018, pp. 106–110.

[6] Karen Ullrich, Jan Schlüter, and Thomas Grill, "Boundary detection in music structure analysis using convolutional neural networks," in *International Conference on Music Information Retrieval*, 2014.

[7] Mark Levy and Mark Sandler, "Structural Segmentation of Musical Audio by Constrained Clustering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 318–326, 2 2008.

[8] Akira Maezawa, "Fast and accurate: improving a simple beat tracker with a selectively-applied deep beat identification," in *International Conference on Music Information Retrieval*, 2017, pp. 309–315.

[9] Diederik P. Kingma and Jimmy Lei Ba, "Adam: a Method for Stochastic Optimization," *International Conference on Learning Representations 2015*, 2015.

[10] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, pp. 1735–1780, 1997.

[11] Slava Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 3, pp. 400–401, March 1987.

[12] M Mauch, C Cannam, M Davies, S Dixon, C Harte, S Kolozali, D Tidhar, and M Sandler, "OMRAS2 Metadata Project 2009," in *International Conference on Music Information Retrieval (Late-breaking)*.

[13] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, "RWC Music Database: Popular, Classical, and Jazz Music Databases," in *International Conference on Music Information Retrieval*, 2002, pp. 287–288.

[14] Masataka Goto, "AIST Annotation for the RWC Music Database," in *International Conference on Music Information Retrieval*. 2006, pp. 359–360, University of Victoria.

[15] Jordan B. L. Smith, J. Burgoyne Ashley, Ichiro Fujinaga, David De Roure, and J. Stephen Downie, "Design and creation of a large-scale database of structural annotations," in *International Conference on Music Information Retrieval*, 2011, pp. 555–560.

[16] Colin Raffel, Brian Mcfee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P W Ellis, "mir_eval: a transparent implementation of common MIR metrics," in *International Conference on Music Information Retrieval*, 2014.

[17] Oriol Nieto, Morwaread M. Farbood, Tristan Jehan, and Juan Pablo Bello, "Perceptual Analysis of the F-measure for Evaluating Section Boundaries in Music," *International Conference on Music Information Retrieval*, 2014.

[18] Jasper Snoek, Hugo Larochelle, and Ryan P Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[19] Oriol Nieto and Juan Pablo Bello, "Systematic Exploration of Computational Music Structure Research," *International Conference on Music Information Retrieval*, 2016.

[20] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Arcos, "Unsupervised Detection of Music Boundaries by Time Series Structure Features," in *AAAI Conference on Artificial Intelligence*, 2012, pp. 1613–1619.

[21] Brian McFee and Daniel P. W. Ellis, "Analyzing song structure with spectral clustering," in *International Conference on Music Information Retrieval*, 2014.