On Hierarchical Clustering of Spectrogram

Shun Sawada*, Yoshinari Takegawa, and Keiji Hirata

Future University Hakodate

Abstract. We propose a new method of applying Generative Theory of Tonal Music directly to a spectrogram of music to produce a time-span segmentation as hierarchical clustering. We first consider a vertically long rectangle in a spectrogram (bin) as a pitch event and a spectrogram as a sequence of bins. The texture feature of a bin is extracted using a gray level co-occurrence matrix to generate a sequence of the texture features. The proximity and change of phrases are calculated by the distance between the adjacent bins by their texture features. The global structures such as parallelism and repetition are detected by a self-similarity matrix of a sequence of bins. We develop an algorithm which is given a sequence of the boundary strength between adjacent bins, iteratively merges adjacent bins in the bottom-up manner, and finally generates a dendrogram, which corresponds to a time-span segmentation. We conducted an experiment with inputting Mozart's K.331 and K.550 and obtained promising results although the algorithm does not take into account almost any musical knowledge such as pitch and harmony.

Keywords: Generative Theory of Tonal Music, Time-Span Segmentation, Gray Level Co-occurrence Matrix, Self-Similarity Matrix, Dendrogram

1 Introduction

A Generative Theory of Tonal Music (GTTM) is known as one of the most reliable music theories, which proposed intuitive and effective concepts and data structures for representing and understanding music, such as reduction, time-span tree and prolongational tree [8]. It is, however, widely recognized that there are intrinsic difficulties in the analysis by GTTM [5, 6]; (i) although many preference rules are specified to retrieve the information in music to generate time-span and prolongational trees, there is not given the method to resolve the competitive preference rules, and (ii) only a homophonic music written on a score can be handled, but neither polyphony nor musical audio. For (i), the musical factors that often make the preference rules competitive contain the distances made of pitch and temporal intervals, the local structural constraint and global dependency, and the boundaries made of harmony and metrical structure. In general, it depends on cases to give priority to either of them, and the definitive rules for controlling the priority for relevant preference rules have not been found

* email: g2116022@fun.ac.jp

yet. For (ii), GTTM was originally developed for analyzing a homophonic music written on a score. However, in reality, there are several cases in which the music to be analyzed is given in the audio format and/or a polyphonic music.

The significance of this work is as follows. If GTTM is applicable to musical audio, any style and/or format of music could be analyzed in the musically reliable way: for example, polyphony, any genre of music, music without a score, classical music to pop music, string quartet to orchestra, and music with expression. Here, we are interested in the musical structure as the result of analyzing a spectrogram without musical knowledge and only with human's innate hearing capability, that is, gestalt.

For GTTM to be applicable directly to a musical audio, however, the following problems should be solved: (a) how to translate the GTTM preference rules into the ones applicable to a musical audio, and (b) how to integrate the results of the applications of the preference rules into a single musical structure. For (a), since we can consider a score as the 2-dimensional coordinate system of beat (x-axis) and pitch (y-axis), by translating beat into time and pitch into frequency, the preference rules may be applicable to a spectrogram. Since a spectrogram, however, contain many confusing partials, fuzzy unstable patterns, and so on, it is difficult to recognize and segregate each note in melodies and chords precisely. Hence, the straightforward way of the original GTTM preference rules being applied to the notes extracted from a musical audio or a spectrogram does not seem promising. For (b), even if the results of the original rules being applied to the notes extracted from a musical audio would be precise, the problem of integrating the results of the preference rules are not yet resolved. As long as the problem of integration is naively transformed into that of the weight adjustment for each preference rule as in the previous research [5], we might be staying far from a fundamental solution.

In the paper, we propose a new method of applying GTTM directly to a musical audio, which we think has a potential to resolve the above two difficulties. In addition, we investigate how accurate the musical structure analysis can be done from the spectrogram without using musical knowledge only with the ability of gestalt cognition which the human hearing originally possesses. We focus on the alternative features extracted from a musical audio, texture features of a spectrogram. While admitting the effectiveness of the low-level audio features such as a chromagram and the MFCC features, as a feasibility study, we would investigate a new method based on the texture features of a spectrogram to produce a time-span segmentation. A time-span segmentation is one of the basic musical structures introduced by Lerdahl and Jackedoff, which is defined as the domains over which reduction takes place. For samples, see Fig. 6.5 (p.127) and Fig. 6.8 (p.129) in [8]. It is constructed of the results of the grouping and metrical structure analyses so that the extracted grouping structure as the upper-level is placed on the extracted metrical structure as the lower-level.

Furthermore, a time-span tree is generated by combining the head selection within each segment with a time-span segmentation. Since we focus on the texture features of a spectrogram, not handling pitch and harmonic information, we do not handle a time-span tree but a time-span segmentation.

2 Related Work

First, let us briefly survey the typical methods employed in the previous work for detecting and extracting the musical structures, such as boundaries, repetitions, and sections within a piece of music, from low-level audio features. Chen and Li proposed a method for decomposing an audio of music into segments such as intro, verse, bridge, and outro [1]. Chen and Li used the harmonic information based on chroma features and the timbral information based on MFCC to produce segment labels, respectively. Next, a new representation *score matrix*, which serves the similar purpose of visualizing music structures as Foote's self-similarity matrix [3], were introduced for combining the two different aspects of an audio of music, harmony and timbre. Then, a score matrix was factorized into the multiplication of the templates of segment types and the activations along time by NMF. The Chen and Li's method is inspired by the observation that music structure is perceived based on various kinds of sources of sound information, among which harmony and timbre play a primary role.

McFee and Ellis proposed a compact representation for effectively encoding repetition structures within a song at multiple levels of granularity [9]. Their method begins with producing a binary recurrence matrix made of audio-level features, such as a chromagram and an MFCC sequence. Here, the two contrasting features are used: harmonic features for detecting long-term repetitions and timbral features for detecting local consistency. Then, to facilitate the discovery of repetition structures, the internal local and long-term connectivities among samples are properly developed and, finally, with balancing local and global linkages, a sequence-augmented affinity matrix that encodes repetition structures is obtained.

Ullrich, Schlüter, and Grill also tackled a similar problem of music segmentation, in which the boundaries within music such as chorus and verse are detected as humans annotate [12]. Ullrich *et al.* let a Convolutional Neural Network (CNN) directly learn the corpus of Mel-scaled magnitude spectrograms with human annotations. Then, they claimed that while many of existing music segmentation algorithms are nearly hand-designed and need much fine-tuning to optimize performance, supervised learning with CNN outperform hand-design ones without domain knowledge. CNN is advantageous for a computer because CNN can identify by itself the features relevant to music segmentation.

Next, let us briefly review the methods developed in the previous work for classifying a piece of music in terms of genre and mood. Costa *et al.* employed the gray level co-occurrence matrix (GLCM) and Local Binary Patterns (LBP) as textural features for automatic music genre classification [2]. Intuitively, GLCM provides the quantitative measures of textural properties of a picture such as smoothness, coarseness, and regularity [7] (see more in Section 3.1). Among the set of the 14 properties originally suggested by Haralick, Costa *et al.* used seven ones, Entropy, Correlation, Homogeneity, 3rd Order Momentum, Maximum Likelihood, Contrast, and Energy, and used SVM as the classifier with the Gaussian kernel. They considered the two different strategies of extracting features: global (holistic) and local (zoning). In the former, the features are extracted from the entire spectrogram and, then, classified by a single genre. In the latter, a spectrogram is firstly divided into several zones (bins), the bins are classified independently, and the final decision is made by combining all the partial classifications of the bins. As a result, the latter with division by 5 was better than the former and achieved the highest performance.

Nakashika, Garcia, and Takiguchi also basically employed GLCM for feature extraction and CNN for a classifier of musical genre classification [11]. Nakashika *et al.* provided multiple GLCM maps with different offset parameters (distance and angle) from a short-term Mel-scaled spectrogram. After several pre-experiments, they fixed the distance of the offset parameters as 1 and set the angle as either of 0°, 45°, 90°, and 135°. The set of GLCM maps with these different offset parameters integratively produced the input data to CNNs as a classifier. Since the set of GLCM maps cooperatively capture the local music patterns and, as a result, outperformed the cases in which a single GLCM map was only used.

3 Method for Hierarchical Clustering of Spectrogram

The method for translating the application of each GTTM preference rule into pattern recognition of a spectrogram is as follows. Grouping Preference Rule (GPR) 2 and 3 prescribe the way of forming groups and boundaries based on the proximity and change between pitch events, respectively. We first consider a spectrogram as a sequence of vertically long rectangles and a vertically long rectangle in a spectrogram as a pitch event. Using a pattern recognition technique, the distance between adjacent vertically long rectangles in a spectrogram is calculated and used for the measures of proximity and change. GPR 4 prescribes that the higher the extent to which GPRs 2 and 3 hold is, the more the effects of GPRs 2 and 3 are taken into account. Thus, the measures of proximity and change are defined as real numbers. GPR 6 prescribes that if parallel (repetitive) motives or phrases are found, the endpoints of each motive or phrase work as the boundaries with the same effects. In our method, we employ the technique of a self-similarity matrix for detecting parallelism (repetition). At present, we do not take into account GPRs 1 (avoiding a group of a single pitch event), 5 (symmetry), 7 (time-span and prolongational stabilities) due to a feasibility study[1].

Fig. 1 shows the overview of our method, which produces the hierarchical clustering of a spectrogram. In the top row of the figure, applying short-time fourier transform (STFT) to input audio with the window size being 1024 and the hop size 256, the spectrogram is plotted in a gray scale of 256 levels. After

[1] Since the space is limited, for more detail, see literatures [8, 5, 6].

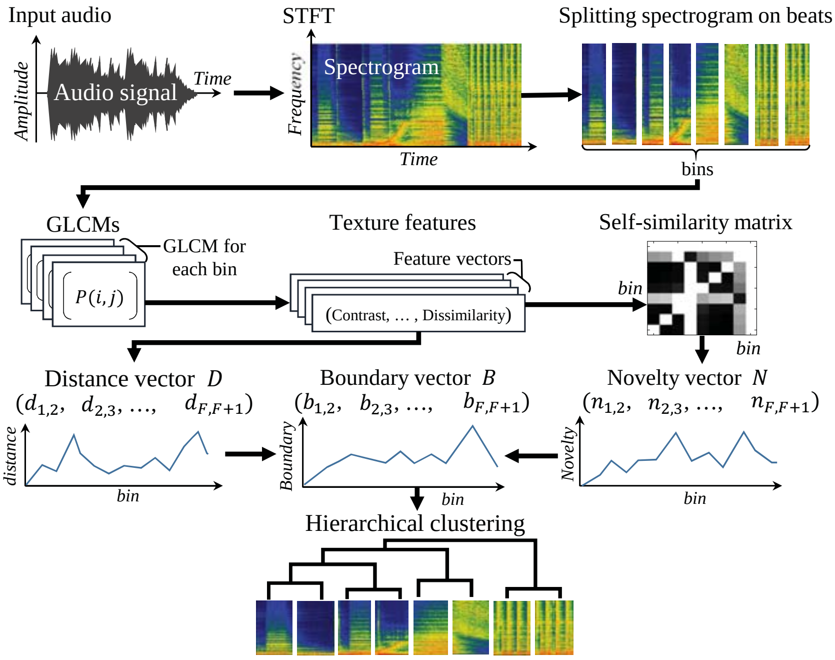

Fig. 1. Overview of Our Method

Ullrich *et al.* [12] and Nakashika *et al.* [11], the frequency axis of the spectrogram is Mel-scaled. Then, the spectrogram is split on every beat position into the short spectrograms the length of which is a beat, called bins. We employed the same beat synchronization technique as in McFee and Ellis [10]; Costa *et al.*'s work [2] also supported the method of dividing a spectrogram into several bins outperformed the holistic processing. We used the *onset_detect* function of the librosa library for beat detection. Since the estimated onsets calculated by the *onset_detect* function may include wrong beat positions, we have selected correct ones from the calculated onsets by hand.

3.1 Gray Level Co-Occurrence Matrix and Texture Features

In the second row of Fig. 1, the texture features are extracted for each bin, using a gray level co-occurrence matrix (GLCM) [7]. GLCM is a matrix representing the frequencies of co-occurring pixel values at neighboring pixel pairs over an image (Fig. 2 (Left)). At first, the co-occurrence of the current pixel value and the value of a neighboring pixel located at a specific offset (in our model, the angles are 0°, 45°, 90°, and 135°; the distance is always 2) is taken into account. Next, for instance in Fig. 2 (Right), for the four offsets, if the co-occurrence of pixel values is (i, j), the value of GLCM at (i,j)-position is incremented. Usually, each element of GLCM is normalized to a value from 0.0 to 1.0 so that the

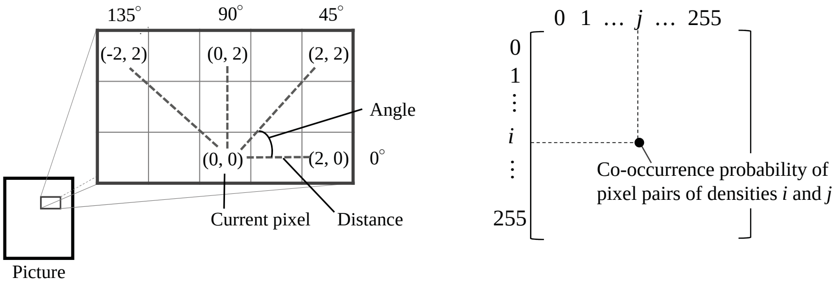

Fig. 2. (Left) Direction and distance of pixel pair; (Right) GLCM

sum of all elements is equal to 1.0. By the definition, the GLCM is apparently invariant to the parallel transposition of patterns. Thus, if GLCM is applied to a spectrogram of a musical audio with the frequency axis plotted in the log scale, the GLCM properties are invariant to the phrases located at parallel. For the offset in our method, the same four angles and distance as Nakashika *et al.* [11] are used; the angles are 0°, 45°, 90°, and 135°; the distance is always 1.

Furthermore, Haralick proposed a method for classifying textures by calculating the secondary features from a GLCM such as contrast and dissimilarity, which represent the higher-order statistical information of an image [7]. Among the set of the 14 secondary features originally suggested by Haralick, we adopt five out of them: contrast, dissimilarity, homogeneity, angular second moment (ASM) and correlation. Contrast and dissimilarity are defined relevant to the density difference of pixel pairs. The more number of pixel pairs with a large density difference, the higher the both values of contrast and dissimilarity are. However, the value of contrast increases exponentially, yet that of dissimilarity linearly. Homogeneity is a feature indicating how close the elements in GLCM are to the diagonal line. This is because the elements on the diagonal line represent the frequency of the co-occurrence (i, i). If the number of the elements distant from the diagonal line increases, the value of homogeneity decreases exponentially. If the texture is in order, the value of ASM becomes high. In case of all the elements in GLCM having a same value, the value of ASM reaches the maximum, 1.0. Correlation is a feature indicating the degree of the linear dependency in pixel pairs over an image. For the mathematical definitions of these secondary features, see [7].

Given the GLCM for a bin, the above five secondary features are computed and standardized so that the mean of the value of each secondary feature is 0.0 and the deviation 1.0. Finally, a feature vector is constructed of these standardized values, which represents the texture feature of a bin that is a partial spectrogram.

3.2 Boundary Vector

A distance vector D is generated from the series of feature vectors by calculating the Euclidean distance between two feature vectors. For instance in Fig. 1, the distance between the i-th bin and the j-th bin is denoted as $d_{i,j}$. To construct the time-span segmentation from the series of bins, the closest bins are basically being grouped in the bottom-up manner along the time as we make a dendrogram. There are, however, two points that we should take care of; one is the global properties of the time-span segmentation, symmetry and parallelism, and the other is that a usual algorithm for making a dendrogram which may merge two bins that are close yet not next to each other.

For the former problem, the measure of novelty [4] is introduced; the novelty value is computed by correlating a (possibly Gaussian-tapered) "checkerboard" kernel matrix along the main diagonal of a self-similarity matrix made of the above feature vector; in our model, the size of a "checkerboard" kernel matrix is set 2 by 2. Since a self-similarity matrix reflects the information of relatively large structures such as repetition and section, peaks in the correlation intuitively mean the strength of structural boundaries of music, taking into account its global structure and dependency. Novelty is here represented in the form of a novelty vector of the same length as a given feature vector, N, in the middle row of Fig. 1. Finally, a boundary vector B representing the total strength of a boundary between adjacent bins are obtained by multiplying ith-elements of a distance vector D and a novelty vector N; that is, $b_{i,i+1} := d_{i,i+1} \cdot n_{i,i+1}$.

For the latter problem, we develop a new algorithm of hierarchical clustering for time-span segmentation to be described in the next section so that the adjacent bins are only merged (the bottom of Fig. 1).

3.3 Hierarchical Clustering

Figure 3 (Left) shows the algorithm for hierarchical clustering for time-span segmentation; (Right) shows the example of bins being merged into larger ones. In our method, each time two bins are merged, the GLCM features of the newly created bin are re-calculated, and accordingly the distance and the boundary vectors are also re-calculated. In (Left) of Fig. 3, the first 6 steps of the algorithm are for initialization, which have been explained in the previous section. The next 5 steps make the loop for iteratively merging bins with updating distance and boundary vectors, D and B (not novelty vector N) until all the bins are merged into a single bin (the original whole spectrogram). At step “Weighting B by number of bins”, the boundary strength is augmented by the number of unit bins contained in merged bins relevant to a boundary. The weighting process is inspired by the observation that the larger a bin merged is, the stronger the effective strength of boundary is, and the harder a bin merged is further merged to adjacent one. Checking the values in the weighted boundary vector, the closest neighboring bins are identified, which have the weakest boundary, and merged into a larger bin.

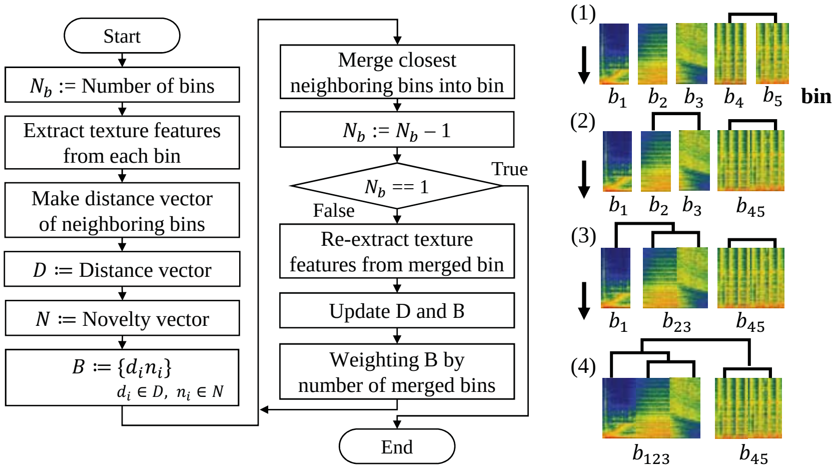
Start
N_b := Number of bins
Extract texture features from each bin
Make distance vector of neighboring bins
D ≔ Distance vector
N ≔ Novelty vector
$B ≔ \{d_i n_i\}$
$d_i \in D,\ n_i \in N$
Merge closest neighboring bins into bin
$N_b := N_b - 1$
$N_b == 1$
True
False
Re-extract texture features from merged bin
Update D and B
Weighting B by number of merged bins
End
(1)
b_1 b_2 b_3 b_4 b_5 bin
(2)
b_1 b_2 b_3 b_{45}
(3)
b_1 b_{23} b_{45}
(4)
b_{123} b_{45}

Fig. 3. (Left) Algorithm of hierarchical clustering; (Right) Iteratively merged bins

The process of merging bins is depicted as the evolution of a dendrogram as in (Right) of the figure. For instance, at stage (1), since bins b_4 and b_5 are the closest, they are merged into b_{45} [2]. The height of the line segment connecting b_4 and b_5 stands for the strength of boundary; the higher the line segment is, the stronger the strength of boundary is.

At stage (2), for the series of b_1, b_2, b_3 and b_{45}, distance and boundary vectors, D and B, are first re-calculated. Then, the boundary strength is augmented as follows; for boundary strength $b_{3,4}$, since b_3 is made of a unit bin and b_{45} two unit bins, $b_{3,4}$ is weighted by 2 ($= 1 \times 2$) to yield $b'_{3,4}$ ($= 2 \times b_{3,4}$). As a result, the boundary strength gets stronger, and it becomes hard for b_3 and b_{45} to be merged equivalently. On the other hand, for boundary strength $b_{2,3}$, since the bins on both sides, b_2 and b_3, are made of a unit bin, weighted boundary strength $b'_{2,3}$ is still the same as $b_{2,3}$. Then, $b'_{2,3}$ and $b'_{3,4}$ are compared, and b_2 and b_3 are merged because $b'_{2,3}$ is weaker than $b'_{3,4}$ in this example.

At stage (3), for the series of b_1, b_{23}, and b_{45}, distance and boundary vectors, D and B, are also first re-calculated. Then, since b_{23} and b_{45} are both made of two unit bins, boundary strength $b_{23,45}$ is weighted by 4 ($= 2 \times 2$), and $b'_{23,45}$ ($= 4 \times b_{23,45}$) is obtained. On the other hand, $b_{1,23}$ is weighted by 2 because b_1 is made of a unit bin and b_{23} two unit bins, and $b'_{1,23}$ ($= 2 \times b_{1,23}$) is obtained. Finally, $b'_{1,23}$ and $b'_{23,45}$ are compared, and b_1 and b_{23} are merged in this case.

[2] Note that $b_{i,i+1}$ means the strength of boundary between bins b_i and b_{i+1}, and $b_{i,i+1i+2}$ means that between b_i and b_{i+1i+2}.

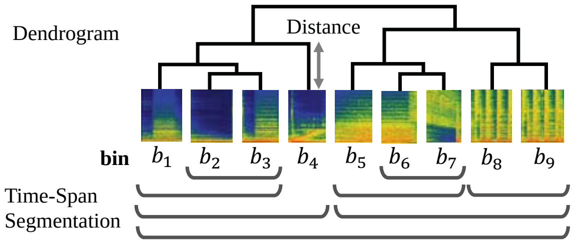

Fig. 4. Dendrogram to Time-Span Segmentation

3.4 Time-Span Segmentation

When the algorithm finishes merging all the bins, a dendrogram representing the process of merging the bins is obtained (Fig. 4). Translating a dendrogram to a time-span segmentation is straightforward. The resulting dendrogram is parsed in the bottom-up manner, and when the point of merging two bins in the dendrogram is met, a new group that spans the previous groups corresponding to these bins is formed. The time-span segmentation obtained in this manner is surely well-formed in the sense that it satisfies the five well-formedness rules listed in [8, pp.37–39]. Since the groups with similar spans, such as 1 beat long and 4 beats long, are perceived as an almost same duration in reality, they are usually plotted at the same height.

4 Experimental Results

To demonstrate our method, we used the two themes from the opening of the Mozart's G Minor Symphony, K.550 and the first movement of Mozart's piano sonata in A major, K.331 from the RWC Music Database [13] (RWC-MDB-C-2001 Nos. 2 and 26). In addition, we used the performance of K.331 by Maria João Pires to compare the analysis results for the same piece. Here, K.550 is polyphonic music performed by a string quartet, and K.331 is homophonic music performed on a piano. For the ground truth of a time-span tree, we have referred to the literature [8]. In figures 5 to 7, system outputs are shown upper and the ground truth lower.

4.1 Mozart's G Minor Symphony, K.550

Fig. 5 shows the result of K.550 of RWC Music Database. Out method succeeded to detect the strongest boundary located between b_4 and b_5, and the analysis result of the first half of the piece was correct. However, that of the second half was wrong; when merging b_6 to either b_5 or b_{78}, the algorithm compared the boundary strengths $b_{5,6}$ and $b_{6,78}$ and made the wrong decision of merging b_6 to b_{78}. The heights of branching nodes in a dendrogram indicates the order of merging bins in reality. Firstly, b_3 and b_4 are merged, and then, so does b_7 and b_8.

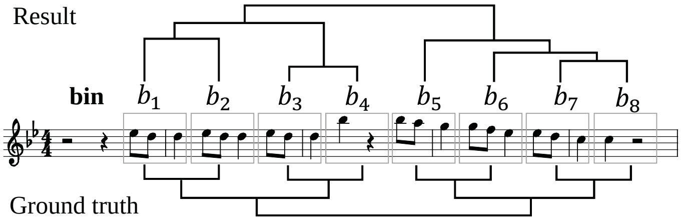

Fig. 5. Dendrogram Obtained from K.550 of RWC MDB

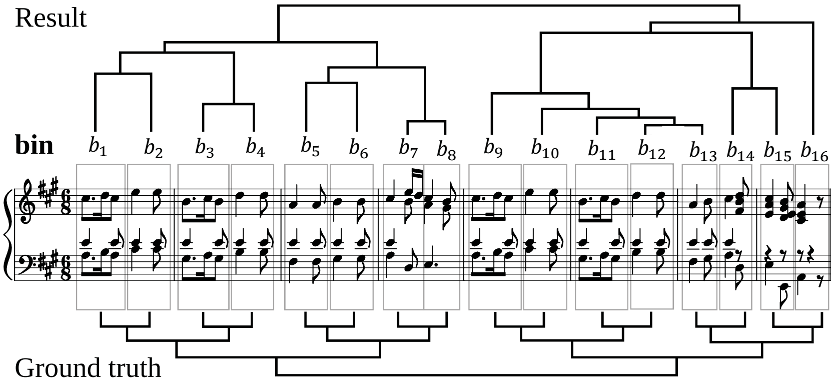

Fig. 6. Dendrogram Obtained from K.331 in RWC Music Database

4.2 Mozart's Piano Sonata in A Major, K.331

Fig. 6 shows the result of K.331 of RWC Music Database, which is more problematic. Although the strongest boundary should occur between b_8 and b_9, the algorithm judged that the strongest is between b_5 and b_6. Although the pairs at the lowest level, such as b_1 and b_2, b_3 and b_4,b_9 and b_10, b_11 and b_12, should be first merged, those pairs were all unfortunately 180°-degree shifted in the real result shown in the figure. Since those wrong pairs were formed at the early stage in generating the dendrogram, the influences of the wrong pairs were propagated up to the top.

Fig. 7 shows the result of K.331 performed by Maria João Pires. In contrast, the strongest boundary between b_8 and b_9 was correctly detected, and among the pairs at the lowest level previously pointed out, the first half of them were also correctly merged, b_1 and b_2, and b_3 and b_4. As for the second half, the configuration of a dendrogram was far from the correct answer. The reason of the wrong merging process in the second half is similar to that in Fig. 5. That is, b_{12} and b_{13} were first merged, and, accordingly, b_{14} and b_{15} were merged with no choice. In this way, the influences at the early stage were propagated up to the top.

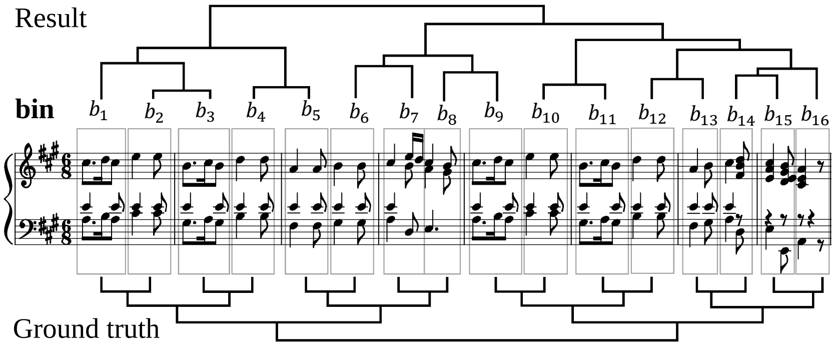
Result
bin
b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8 b_9 b_{10} b_{11} b_{12} b_{13} b_{14} b_{15} b_{16}
Ground truth

Fig. 7. Dendrogram Obtained from K.331 Performed by Maria João Pires

5 Discussion

Let us examine the reason of the wrong merging process occurring at the lowest level. In K.550, after b_7 and b_8 are merged, the distance between b_6 and b_7 are updated. Then, since the measure of novelty between b_5 and b_6 becomes higher, b_6 and b_{78} is merged first. In K.331 (Fig. 6), since the measure of novelty between b_1 and b_2 is high, and those from b_2 through b_4 are zero. Hence, b_2 and b_3 are merged first, the boundary strength between b_3 and b_4 is updated and becomes higher, and as a result, b_4 and b_5 are merged. As for the second half, the undesirable values of novelty are also observed where we do not suppose they are zero, for instance, b_{10} and b_{11}, b_{10} and b_{11}, b_{12} and b_{13}, and b_{14} and b_{15}.

In contrast, the result of K.331 performed by Maria João Pires in Fig. 7 is successful. This is because the correct measures of novelty are calculated here. We suppose that the size of the "checkerboard" kernel is critical. Since the size of the self-similarity matrix (SSM) in our method is 8 by 8 in K.550 and 16 by 16 in K.331, respectively, we cannot use the large size of the "checkerboard" kernel and actually use the "checkerboard" kernel of 2 by 2. However, in the original work by Foote [4], the larger size of the "checkerboard" kernel are used, for instance, 64 by 64, possibly with Gaussian taper for smoothing. Therefore, we need to develop the measure of novelty which works well to a small-sized SSM.

6 Concluding Remarks

We propose a new method of applying Generative Theory of Tonal Music directly to a spectrogram of music to produce a time-span segmentation. Although the attempt to extract a time-span segmentation almost only from the textural features of a spectrogram seemed somehow contradictory, the results shown in Figs. 5, 6, and 7 were more promising than we expected. This result suggests that the hierarchical clustering in music is not a cognitive function peculiar to music, but one of the general cognitive functions that humans are using for understanding other media. To improve the precision of our method, musical information, such as pitch, harmony, and rhythm, may be helpful.

Future work contains the following three points. The first is conducting a large-sized quantitative experiment. The next is generating the boundary vector with taking into account the musical information, such as pitch, harmony, and rhythm. The last is developing an algorithm for hierarchical clustering which employs grouping preference rule no. 7 (symmetry) that is not implemented at present as well as the other preference rules.

Acknowledgement

This work has been supported by JSPS Kakenhi 16H01744.

References

1. R. Chen and M. Li: Music Structural Segmentation By Combining Harmonic and Timbral Information, In *Proc. of ISMIR*, pp.477–482 (2011).
2. Y. M. G. Costa, L. S. Oliveira, A. L. Koerich, and F. Gouyon: Comparing textural features for music genre classification, *Proc. of The 2012 International Joint Conference on Neural Networks*, pp.1867–1872 (2012).
3. J. Foote: Visualizing Music and Audio using Self Similarity, In *Proc. of the 7th ACM int'l conf. on Multimedia*, pp.77–80 (1999).
4. J. Foote: Automatic audio segmentation using a measure of audio novelty, In *Proc. of IEEE International Conference on Multimedia and Expo*, vol.1, pp.452–455 (2000).
5. Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo, Implementing "A Generative Theory of Tonal Music", *Journal of New Music Research*, 35:4, pp.249-277 (2007).
6. Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo, Implementing Methods for Analysing Music Based on Lerdahl and Jackendoff's Generative Theory of Tonal Music, David Meredith (Ed), *Computational Music Analysis*, Chapter 9, pp.221-249, Springer (2016).
7. R. M. Haralick: Statistical and structural approaches to texture, In *Proc. of the IEEE*, vol.67, No.5, pp.786–804 (1979).
8. F. Lerdahl and R. Jackendoff: *A Generative Theory of Tonal Music*, The MIT Press (1983).
9. B. McFee and Daniel P. W. Ellis: Analyzing Song Structure with Spectral Clustering, In *Proc. of ISMIR*, pp.405–410 (2014).
10. B. McFee and Daniel P. W. Ellis: Learning to Segment Songs with Ordinal Linear Discriminant Analysis, In *Proc. of ICASSP* (2014).
11. T. Nakashika, C. Garcia, and T. Takiguchi: Local-feature-map Integration Using Convolutional Neural Networks for Music Genre Classification, In *Proc. of Interspeech*, pp.1752–1755, ISCA (2012).
12. K. Ullrich, J. Schlüter, and T. Grill: Boundary Detection in Music Structure Analysis using Convolutional Neural Networks, In *Proc. of ISMIR*, pp.417–422 (2014).
13. M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka: RWC Music Database: Popular, Classical and Jazz Music Databases, In *Proc. of ISMIR*, pp.287–288 (2002).