Automatic Generation of Grouping Structure based on the GTTM

Masatoshi Hamanaka[1), 2)], Keiji Hirata[3)] and Satoshi Tojo[4)]

1) Research Fellow of the Japan Society for the Promotion of Science,
2) National Institute of Advanced Industrial Science and Technology (AIST),
3) NTT Communication Science Laboratories,
4) Japan Advanced Institute of Science and Technology
m.hamanaka@aist.go.jp

Abstract

This paper describes an automatic grouping system, which segments the music into units such as phrases or motives, based on the Generative Theory of Tonal Music *(GTTM in short, hereafter). The GTTM is considered to be one of the most promising theories of music in regard to computer implementation; however, no order in applying those rules is given and thus, more often than not, may result in conflict among them. To solve this problem, we introduce adjustable parameters, which enable us to give priority among rules. We show the experimental results that our method outperformed the baseline performance by over thirty percent, tuning the parameters. In addition, we show that the system displays the time-span tree based on these grouping rules together with metric information given by input MusicXML.*

1 Introduction

The purpose of this study is to automatically derive a time-span tree that assigns a hierarchy of `structural importance' to the notes of a piece of music. The hierarchy is based on the generative theory of tonal music (GTTM) (Lerdahl and Jackendoff 1983). Automatic generation of a time-span tree from the music surface enables us to analyze the deeper structure (Hirata and Aoyagi, 2003). It also provides a summarization of the music, which can be used as a representation of search, and thus results in music retrieval systems (Hirata and Matsuda 2003).

The GTTM is composed of four modules, each of which assigns a separate structural description to a listener's understanding of music. These four modules output the grouping structure, metrical structure, time-span reduction, and prolongational reduction, respectively. The grouping structure is a hierarchical segmentation which results in motives, phrases, and sections. The result of grouping is used to derive a time-span tree, together with metrical information of MusicXML. In this paper, we describe a method to articulate automatically a transition of notes into groupings by the GTTM.

Previous segmentation methods have been unable to construct hierarchical grouping structures because they have focused on detecting the local boundaries of the melody (Stammen and Pennycook 1994), (Temperley 2001), (Cambouropoulos 2001), (Ferrand, Nelson and Wiggins 2003).

Attempts have been made to implement several of the rules of the GTTM grouping structure in computer systems, but these methods have been incapable of resolving the conflict between the rules (Ida, Hirata, and Tojo 2001), (Touyou, Hirata, and Tojo 2002).

Our system of segmentation based on the GTTM makes it possible to construct hierarchical grouping structures in a top-down process using bottom-up detection of local boundaries. The system is equipped with adjustable parameters, and they enable us to control the strength of each rule. When a user changes the parameters, the hierarchical grouping structures change as a result of the new segmentation. With this system, we came to generate time-span trees, searching for plausible parameter sets.

2 Problems of applying grouping rules

The grouping structure is intended to formalize the intuitive belief that tonal music is organized into groups that are in turn composed of subgroups. These groups are presented graphically as several levels of arcs below a music staff. There are two types of rules for grouping in the GTTM: grouping well-formedness rules (GWFR) and grouping preference rules (GPR). Grouping well-formedness rules are necessary conditions for the assignment of a grouping structure and restrictions on these structures. When more than one structure can satisfy the well-formedness rules of grouping, the grouping preference rules (GPR) indicate the superiority of one structure over another. The GPRs consist of seven rules: GPR1 (alternative form), GPR2 (proximity), GPR3 (change), GPR4 (intensification), GPR5 (symmetry), GPR6 (parallelism), and GPR7 (time-span and prolongational stability). GPR2 has two cases: (a) (slur/rest) and (b) (attack-point). GPR3 has four cases: (a) (register), (b) (dynamics), (c) (articulation), and (d) (length).

In this section, we specify the problems with GPRs in terms of computer implementation.

2.1 Conflict between rules

Because there is no strict order for applying GPRs, the conflict between rules often occurs when applying GPRs and results in ambiguities in analysis. Figure 1 shows a simple example of the conflict between GPR2b (Attack-Point) and GPR3a (Register). GPR2b states that a relatively greater interval of time between attack points initiates a grouping boundary. GPR3a states that a relatively greater pitch difference in between smaller neighboring intervals initiates a grouping boundary. Because GPR1 (alternative form) strongly prefers that note 3 alone not form a group, a boundary cannot be perceived at both 2-3 and 3-4.

To solve this problem, we use adjustable parameters $S^{GPR\ j(=\ 2a,\ 2b,\ 3a,\ 3d,\ 4,\ 5,\ 6)}$ (0≦ $S^{GPR\ j}$ ≦1) that enable us to control the strength of each rule.

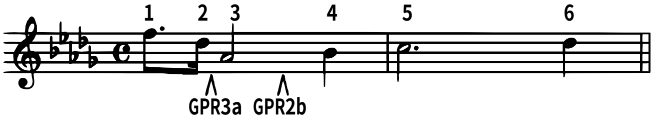

Figure 1. Simple example of conflict between rules.

2.2 Ambiguity in defining GPR4, 5, and 6

The GTTM does not resolve much of the ambiguity that exists in applying GPR4, 5, and 6. For example, GPR6 (Parallelism) does not define the decision criteria for construing whether two or more segments are parallel or not. The same problems occur with GPR4（Intensification）and GPR5 (Symmetry).

To solve this problem we attempted to formalize the criteria for deciding whether each rule is applicable or not.

3 Automatic segmentation system based on the GTTM

Figure 2 shows the processing flow of the system. As a primary input format we chose MusicXML (Recordare LLC 2004) because the format is expected to be a common interlingua in music notation, analysis, retrieval, and other applications. A hierarchical grouping structure was constructed top-down using bottom-up detection of local boundaries. We then designed Grouping-XML as the output format for segmentation results. In our experiments, we restrict the music structure to mononphony to correctly evaluate the performance of each rule.

3.1 MusicXML

MusicXML is a music representation format based on XML (extensible mark-up language). It has attribute elements and note elements. The attribute element contains the musical attributes of scores, such as the key signature, time signature, and clef. The time signature includes the numerator (beats) and denominator (beat-type). The note element has a pitch defined by step and octave elements. Every note also has a duration based on divisions of a quarter note. Several additional elements may be associated with a note. Tied notes, slurs, fermatas, and arpeggios are represented by top-level children of the notation element. Dynamics, ornaments, articulations, and technical indications specific to particular instruments are also top-level children of the notation element.

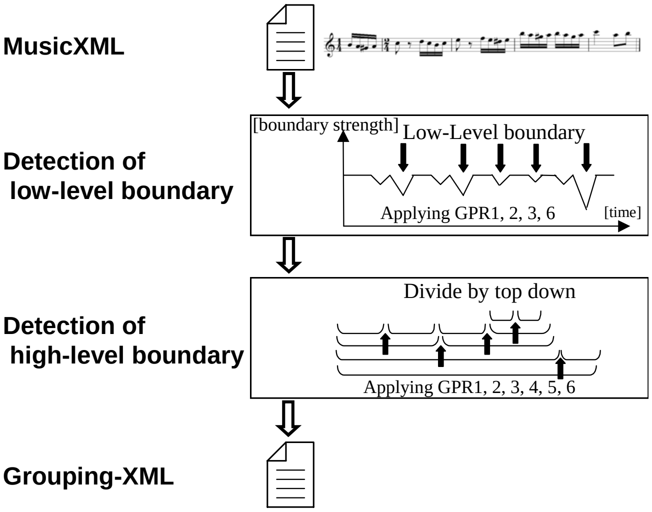

Figure 2. Processing flow of the system

In this experiment, we prepared the input date in MusicXML with *Finale*$^{\text{TM}}$[†] and *Dolet*$^{\text{TM}}$ *for Finale plug-in*, in which such tags as dynamics and articulations were not included.

3.2 Application of GPRs

In this section, we discuss the application of GPR1, GPR2a, GPR2b, GPR3a, GPR3d, GPR4, GPR5, and GPR6. We were unable to apply GPR3b (articulation) and GPR3c (dynamics) because MusicXML for our system input does not have dynamic and articulation elements. The degree of boundary for each rule can be expressed as $D_i^{\mathrm{GPR}\,j=(1,\,2a,\,2b,\,3a,\,3d,\,4,\,5,\,6)}$ (0≦ $D_i^{\mathrm{GPR}\,j}$ ≦1).

Our segmentation system has thirteen adjustable parameters, which include $S^{GPR\ j(=\ 2a,\ 2b,\ 3a,\ 3d,\ 4,\ 5,\ 6)}$, σ, W$_s$, W$_r$, W$_l$, T^{GPR4} , and $T^{low\text{-}level}$ (Table 1).

Table 1: Thirteen adjustable parameters

Parameters	Description
$S^{GPR\ j}$	The strength of each rule. j =(2a,2b,3a,3d,4,5,6)
σ	The standard deviation of a normal distribution for GPR5.
W$_s$	Weight of priority of the same rhythm compared with the same register in parallel segments.
W$_r$	Weight of priority of one end of a parallel segment compared with the start of a parallel segment.
W$_l$	Weight of priority of large parallel segments.
T^{GPR4}	The value of the threshold that decides whether GPRs 2 and 3 are relatively pronounced or not.
$T^{low\text{-}level}$	The value of the threshold that decides whether transition i is a low-level boundary or not.

GPRs 2, 3, and 4. GPRs 2, 3, and 4 are the rules for a transition of four notes. The degree of the boundary of each rule indicates whether the transition of i is heard as a group boundary ($D_i^{\mathrm{GPR}j}$ =1) or not ($D_i^{\mathrm{GPR}j}$ =0). GPR4 has an adjustable parameter T^{GPR4} (0≦ T^{GPR4} ≦1) to control the value of the threshold that decides whether GPRs 2 and 3 are relatively pronounced or not.

[†] See http://www.recordare.com/

$$D_i^{GPR2a} = \begin{cases} 1 & rest_{i-1} < rest_i \ and\ rest_i > rest_{i+1} \\ 0 & rest_{i-1} \geq rest_i \ or\ rest_i \leq rest_{i+1} \end{cases} \quad (1)$$

$$D_i^{GPR2b} = \begin{cases} 1 & ioi_{i-1} < ioi_i \ and\ ioi_i > ioi_{i+1} \\ 0 & ioi_{i-1} \geq ioi_i \ or\ ioi_i \leq ioi_{i+1} \end{cases} \quad (2)$$

$$D_i^{GPR3a} = \begin{cases} 1 & regist_{\,i-1} < regist_{\,i} \ and\ regist_{\,i} > regist_{\,i+1} \\ 0 & regist_{\,i-1} \geq regist_{\,i} \ or\ regist_{\,i} \leq regist_{\,i+1} \end{cases} \quad (3)$$

$$D_i^{GPR3d} = \begin{cases} 1 & len_{i-1} = 0 \ and\ len_i \neq 0 \ and\ len_{i+1} = 0 \\ 0 & len_{i-1} \neq 0 \ or\ len_i = 0 \ or\ len_{i+1} \neq 0 \end{cases} \quad (4)$$

$$D_i^{GPR4} = \begin{cases} 1 & P_i^{rest} > T^{GPR4} or\ P_i^{ioi} > T^{GPR4} \ or\ P_i^{regist} > T^{GPR4} \\ 0 & P_i^{rest} \leq T^{GPR4} and\ P_i^{ioi} \leq T^{GPR4} and\ P_i^{regist} \leq T^{GPR4} \end{cases} \quad (5)$$

where

$rest_i$: interval between current offset and next onset.
ioi_i : inter onset intervals.
$regist_i$: pitch intervals.
len_i : subtraction of duration.

$$P_i^{rest} = \frac{rest_i}{\sum_{j=i-1}^{i+1} rest_j},\ P_i^{ioi} = \frac{ioi_i}{\sum_{j=i-1}^{i+1} ioi_j},\ P_i^{regist} = \frac{regist_i}{\sum_{j=i-1}^{i+1} regist_j}$$

GPR5. GPR5 is the rule for symmetry in a grouping structure. We use a normal distribution with the standard deviation σ as a symmetry level D^{GPR5} so that there is a preference to subdivide groups into two parts of equal length.

$$D_i^{GPR5} = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\left(\sum_{j=start}^{i} ioi_j - \sum_{j=start}^{end} ioi_j \Big/ 2\right)^2}{2\sigma^2}} \quad (6)$$

where

$start$: start transition of a group.
end : end transition of a group.

GPR6. GPR6 is the rule for parallelism in the grouping structure. GPR6 has three adjustable parameters, which are W$_s$ (weight of priority of the same rhythm compared with the same register in parallel segments), W$_r$ (weight of priority of one end of a parallel segment compared with the start of a parallel segment), and W$_l$ (weight of priority of large parallel segments) (0≦W$_s$, W$_r$, W$_l$ ≦1). We define the parallel level D^{GPR6} as follows:

$$D_i^{GPR6} = \sum_j \sum_r \begin{cases} G_{i\,j}^{start} \times (1 - W_s) & m_{i\,j} = 1 \\ G_{i\,j}^{end} \times W_s & m_{i\,j} = 2 \\ G_{i\,j}^{start} \times (1 - W_s) + G_{i\,j}^{end} \times W_s & m_{i\,j} = 3 \\ 0 & m_{i\,j} = 0 \end{cases} \quad (7)$$

where

$division$: duration of a quarter note.
r : length of parallel segments based on the division of quarter notes.

$$G_{i\,j}^{start} = \frac{z_{q_i\ q_j\ r}}{x_{q_i\,r} + x_{q_j\,r}} \times (1 - W_r) \times r^{1+W_l} + \frac{y_{q_i\ q_j\ r}}{z_{q_i\ q_j\ r}} \times W_r \times r^{1+W_l}$$

$$G_{i\,j}^{end} = \frac{z_{q_i\text{-}r\ q_j\text{-}r\ r}}{x_{q_i\text{-}r\ r} + x_{q_j\text{-}r\ r}} \times (1 - W_r) \times r^{1+W_l} + \frac{y_{q_i\text{-}r\ q_j\text{-}r\ r}}{z_{q_i\text{-}r\ q_j\text{-}r\ r}} \times W_r \times r^{1+W_l}$$

$$m_{i\,j} = \begin{cases} 1 & q_i \neq q_{i-1} \ and\ q_j \neq q_{j-1} \ and\ q_i = q_{i+1} \ and\ q_j = q_{j+1} \\ 2 & q_i = q_{i-1} \ and\ q_j = q_{j-1} \ and\ q_i \neq q_{i+1} \ and\ q_j \neq q_{j+1} \\ 3 & q_i \neq q_{i-1} \ and\ q_j \neq q_{j-1} \ and\ q_i \neq q_{i+1} \ and\ q_j \neq q_{j+1} \\ 0 & else \end{cases}$$

$$x_{q_i\,r} = \sum_j \begin{cases} 1 & q_i \leq q_j \ and\ q_j \leq q_i + r \\ 0 & else \end{cases}$$

$$y_{q_i\,q_j\,r} = \sum_k \sum_l \begin{cases} 1 & (q_i - q_j) \times division = \sum_{g=1}^{k} ioi_g - \sum_{g=1}^{l} ioi_g \\ 0 & else \end{cases}$$

$$z_{q_i\,q_j\,r} = \sum_k \sum_l \begin{cases} 1 & (q_i - q_j) \times division = \sum_{g=1}^{k} ioi_g - \sum_{g=1}^{l} ioi_g \ \ and\ \ regist_i = regist_j \\ 0 & else \end{cases}$$

$$q_i = \left[\frac{\sum_{k=1}^{i} ioi_k}{division} \right] \qquad ([\ ] : Gausian\ \ integer)$$

GPR1. GPR1 is designed to prevent a group from containing a single event. Therefore, the boundary must be stronger than the neighboring transition. The degree of the GPR1 boundary indicates whether the transition of i is heard as a group boundary (D_i^{GPR1} =1) or not (D_i^{GPR1} =0).

$$D_i^{GPR1} = \begin{cases} 1 & \sum_{j=(2a,2b,3a,3d,6)} D_{i-1}^{GPR\,j} \times S^{GPR\,j} \leq \sum_{j=(2a,2b,3a,3d,6)} D_i^{GPR\,j} \times S^{GPR\,j} \\ & and \ \sum_{j=(2a,2b,3a,3d,6)} D_i^{GPR\,j} \times S^{GPR\,j} \geq \sum_{j=(2a,2b,3a,3d,6)} D_{i+1}^{GPR\,j} \times S^{GPR\,j} \\ 0 & else \end{cases} \quad (8)$$

3.3 Detection of low-level boundary

Low-level boundaries are detected by MusicXML using D_i^{GPR1}, D_i^{GPR2a}, D_i^{GPR2b}, D_i^{GPR3a}, D_i^{GPR3d} and D_i^{GPR6}. $T^{low\text{-}level}$ is an adjustable parameter for controlling the value of the threshold that decides whether transition i is a low-level boundary or not. The degree of low-level boundaries $D_i^{low\text{-}level}$ can be expressed as follows:

$$D_i^{low\text{-}level} = \begin{cases} 1 & \sum_{j=(2a,2b,3a,3d,6)} D_i^{GPRj} \times S^{GPRj} > T^{low-level} \ and \ \sum_{j=(2a,2b,3a,3d,6)} D_{i-1}^{GPR1} = 1 \\ 0 & else \end{cases} \quad (9)$$

3.4 Detection of high-level boundaries

A hierarchical grouping structure is constructed in the top-down method while local-boundaries are found in the bottom-up way. A group that contains a local boundary detected iteratively by the next level boundary $\hat{i}$ is calculated as follows:

$$\hat{i} = \underset{i}{argmax} \quad D_i^{low-level} \times \sum_{j=(2a,2b,3a,3d,4,5,6)} D_i^{GPR\,j} \times S^{GPR\,j} \quad (10)$$

3.5 Grouping-XML

We designed Grouping-XML as an export format for hierarchical grouping structures. Grouping-XML has group elements, note elements, and applied elements. Note elements align the order of onset times, which connect to notes in MusicXML using Xpointer (W3C 2002) and Xlink (W3C 2001). All note elements are inside hierarchical group elements. The applied elements are located between the end of a group tag and the start of the next group tag, which is where the GPR are applied. Figure 3 shows a simple example of Grouping-XML. We developed a Grouping-XML viewer to display grouping structures (Figure 4).

```
-<group>
   -<group>
      +<note id="P1-1-1"/>
      +<note id="P1-1-2"/>
      +<note id="P1-1-3"/>
      +<note id="P1-1-4"/>
      +<note id="P1-2-1"/>
   </group>
   <applied  rule=”2a”/>
   <applied  rule=”6”/>
   -<group>
      +<note id="P1-2-3"/>
      +<note id="P1-2-4"/>
      +<note id="P1-2-5"/>
      +<note id="P1-2-6"/>
      +<note id="P1-3-1"/>
   </group>
</group>
```

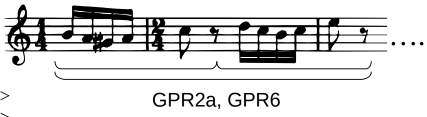

Figure 3 Simple example of Grouping-XML.

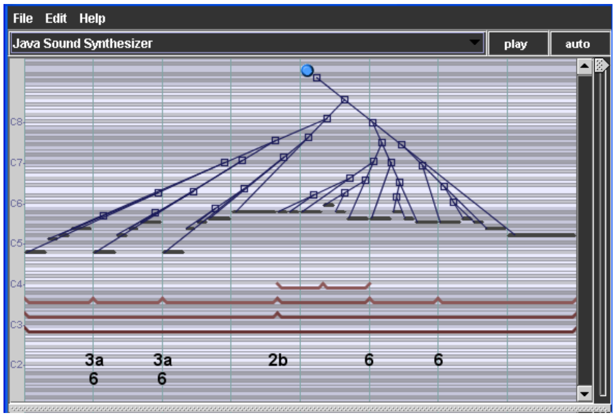

Figure 4 Screen snapshot of Grouping-XML viewer.

4 Experimental results

We evaluated the performance of segmentation using an *F*-measure, which is given by the weighted harmonic mean of *Precision P* (the proportion of selected groupings that are correct) and *Recall R* (the proportion of correct groupings that were identified). We did not take care that low-level error is propagated up to higher-level; we counted wrong answers with ignoring the difference of grouping levels.

$$F_{measure} = 2 \times \frac{P \times R}{P + R} \tag{11}$$

This evaluation required us to prepare correct grouping data. We collected a hundred pieces of 8-bar length, monophonic, classical music pieces, and asked those who have expertise in musicology to give them groupings manually, faithfully with regard to GPR's. Those manual results were cross-checked by three other experts.

The segmentation changes depending on the parameters configured. To evaluate the baseline segmentation performance of our system, we used default parameters, which were $S^{GPR\,j(=\,2a,\,2b,\,3a,\,3d,\,4,\,5,\,6)}$=0.5, σ=0.05, W_s,=0.5 W_r =0.5, W_l=0.5, T^{GPR4} =0.5, and $T^{low\text{-}level}$=0.5. It costed us about 10 minutes on average for a piece to find a plausible tuning of parameter sets. As a result of configuring the parameters, the performance of segmentation outperformed the baseline *F*-measure by more than thirty percent (Table 2).

Table 2: *F*-measure for our method

Melody	Baseline performance	Our method with configured parameters
1. TurkishMarch	0.09	0.95
2. Wiegenlied	0.41	1.00
3. Brindisi	0.03	0.90
4. My dearest father	0.03	0.11
5. The Nutcracker March	0.01	0.05
⋮	⋮	⋮
Total (100 melodies)	0.32	0.67

5 Conclusion

We developed a segmentation system based on the GTTM. This system makes it possible to construct hierarchical grouping structures. Experimental results show that the performance of segmentation outperformed the baseline *F*-measure by more than thirty percent as a result of configuring the parameters. At the current stage, the time-span tree is generated only by GPR's together with metric information given by musicXML. We are now planning to improve the precision of trees, implementing the further details of Time-span tree generation rules.

References

Lerdahl, F., and R. Jackendoff. (1983). A Generative Theory of Tonal Music. Cambridge, Massachusetts: MIT Press.

Hirata, K., and T. Aoyagi. (2003). “Computational Music Representation on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database.” *Computer Music Journal* 27(3), 73–89.

Hirata, K., and S. Matsuda. (2003). “Interactive Music Summarization based on Generative Theory of Tonal Music.” *Journal of New Music Research,* 32:2, 165-177.

Stammen, D. R., B. Pennycook. (1994). “Real-time Segmentation of Music using an Adaptation of Lerdahl and Jackendoff’s Grouping Principles.” In *proceedings of the International Conference on Music Perception and Cognition*, pp. 269-270.

Temperley, D. (2001). The Cognition of Basic Musical Structures. Cambridge, Massachusetts: MIT Press.

Cambouropoulos, E., (2001). “The Local Boundary Detection Model (LBDM) and its application in the study of expressive timing.” In *Proceedings of the International Computer Music Conference,* pp. 290–293. Havana, Cuba: International Computer Music Association.

Ferrand, M., P. Nelson, and G. Wiggins. (2003). “Memory and Melodic Density: A Model for Melody Segmentation.” In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003),* pp. 95–98. Firenze, Italy.

Ida, K, K. Hirata, and S. Tojo. (2001). “The attempt of the Automatic Analysis of the Grouping Structure and Metrical Structure based on GTTM.” SIG Technical Report, Vol. 2001, No. 42, pp 49-54, (in Japanese).

Touyou, K, K. Hirata, S. Tojo, and K. Satoh. (2002). “Improvement of Grouping Rule Application in Implementing GTTM.” SIG Technical Report, Vol. 2002, No. 47, pp. 121-126, (in Japanese).

Recordare LLC. (2004) “MusicXML 1.0 Tutorial.” http://www.recordare.com/xml/musicxml-tutorial.pdf.

W3C. (2001) “XML Linking Language (XLink) Version 1.0.” http://www.w3.org/TR/xlink/.

W3C. (2002) “XML Pointer Language (XPointer).” http://www.w3.org/TR/xptr/.