Automatic Generation of Grouping Structure based on the GTTM

Masatoshi Hamanaka\textsuperscript{1)}, 2), Keiji Hirata\textsuperscript{3)} and Satoshi Tojo\textsuperscript{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 $S_{GPR}^j (\leq 2, 2b, 3a, 3d, 4, 5, 6)$ ($0 \leq S_{GPR}^j \leq 1$) 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 monophony 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-

The standard deviation of a normal distribution for GPR5.

Weight of priority of large parallel segments.

Weight of priority of the same rhythm compared with the start of a parallel segment.

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 ($D_{GPR}^{GRP} = 1$) or not ($D_{GPR}^{GRP} = 0$). GPR4 has an adjustable parameter $D_{GPR}^{GRP} (0 \leq D_{GPR}^{GRP} \leq 1)$ to control the value of the threshold that decides whether GPRs 2 and 3 are relatively pronounced or not.

Figure 2. Processing flow of the system.

Table 1: Thirteen adjustable parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{GPR}^j$</td>
<td>The strength of each rule. $j = (2a, 2b, 3a, 3d, 4, 5, 6)$</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>The standard deviation of a normal distribution for GPR5.</td>
</tr>
<tr>
<td>$W_s$</td>
<td>Weight of priority of the same rhythm compared with the same register in parallel segments.</td>
</tr>
<tr>
<td>$W_e$</td>
<td>Weight of priority of one end of a parallel segment compared with the start of a parallel segment.</td>
</tr>
<tr>
<td>$W_p$</td>
<td>Weight of priority of large parallel segments.</td>
</tr>
<tr>
<td>$T_{GPR}$</td>
<td>The value of the threshold that decides whether GPRs 2 and 3 are relatively pronounced or not.</td>
</tr>
<tr>
<td>$T_{low-level}$</td>
<td>The value of the threshold that decides whether transition $i$ is a low-level boundary or not.</td>
</tr>
</tbody>
</table>

† See http://www.recordare.com/
\[ D_{GPR2\alpha}^{i} = \begin{cases} 1 & \text{rest}_{i+1} < \text{rest}_{i} \text{ and rest}_{i} > \text{rest}_{i+1} \\ 0 & \text{rest}_{i+1} \geq \text{rest}_{i} \text{ or rest}_{i} \leq \text{rest}_{i+1} \end{cases} \]

\[ D_{GPRb}^{i} = \begin{cases} 1 & \text{ioi}_{i+1} < \text{ioi}_{i} \text{ and } \text{ioi}_{i} > \text{ioi}_{i+1} \\ 0 & \text{ioi}_{i+1} \geq \text{ioi}_{i} \text{ or } \text{ioi}_{i} \leq \text{ioi}_{i+1} \end{cases} \]

\[ D_{GPR3\alpha}^{i} = \begin{cases} 1 & \text{regi}_{i+1} < \text{regi}_{i} \text{ and regi}_{i} > \text{regi}_{i+1} \\ 0 & \text{regi}_{i+1} \geq \text{regi}_{i} \text{ or } \text{regi}_{i} \leq \text{regi}_{i+1} \end{cases} \]

\[ D_{GPR4}^{i} = \begin{cases} 1 & \text{p}_{i}^{\text{rest}} > T_{GPR4} \text{ or } \text{p}_{i}^{\text{rest}} > T_{GPR4} \text{ or } \text{p}_{i}^{\text{rest}} > T_{GPR4} \\ 0 & \text{p}_{i}^{\text{rest}} \leq T_{GPR4} \text{ and } \text{p}_{i}^{\text{rest}} \leq T_{GPR4} \text{ and } \text{p}_{i}^{\text{rest}} \leq T_{GPR4} \end{cases} \]

where

rest\(_i\): interval between current offset and next onset.

ioi\(_i\): inter on set intervals.

regi\(_i\): pitch intervals.

duration\(_i\): subtraction of duration.

\[ r_{\text{sum}} = \sum_{j=i}^{\text{end}} \text{rest}_j, r_{\text{ioi}} = \sum_{j=i}^{\text{end}} \text{ioi}_j, r_{\text{regi}} = \sum_{j=i}^{\text{end}} \text{regi}_j \]

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

\[ D_{GPR5} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left( -\frac{x_{\text{sum}}^2}{2\sigma^2} \right) \]

where

start: start transition of a group.

division: duration of a quarter note.

3.3 Detection of low-level boundary

Low-level boundaries are detected by MusicXML using \( D_{GPR}^{i} \) and \( T_{\text{low-level}} \) is an adjustable parameter for controlling the value of the threshold that decides whether a transition \( i \) is a low-level boundary or not. The degree of low-level boundaries \( D_{\text{low-level}} \) can be expressed as follows:

\[ D_{\text{low-level}}^{i} = \begin{cases} 1 & B_i > T_{\text{low-level}} \text{ and } D_{GPR}^{i} = 1 \\ 0 & \text{else} \end{cases} \]

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:

\[ \hat{i} = \arg \max_i \left( D_{\text{low-level}}^{i} \times \sum_{j \in \{2,3,4,5,6\}} D_{GPR}^{j} \right) \]

3.5 GroupingXML

We designed GroupingXML as an export format for hierarchical grouping structures. GroupingXML 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 GroupingXML. We developed a GroupingXML viewer to display grouping structures (Figure 4).
which were performance of our system, we used default parameters, configured. To evaluate the baseline segmentation results were cross-checked by three other experts. Those manual samples have expertise in musicology to give them groupings. 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. The segmentation changes depending on the parameters configured. To evaluate the baseline segmentation performance of our system, we used default parameters, which were $S_{GPR}^{j=c, 2a, 2b, 3a, 3d, 4, 5, 6} = 0.5$, $a = 0.05$, $W_p = 0.5$, $W_s = 0.5$, $T_{GPR} = 0.5$, and $T_{low-level} = 0.5$. 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).

### 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.

$$F_{measure} = \frac{2 \times P \times R}{P + R} \quad (12)$$

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 $S_{GPR}^{j=c, 2a, 2b, 3a, 3d, 4, 5, 6} = 0.5$, $a = 0.05$, $W_p = 0.5$, $W_s = 0.5$, $T_{GPR} = 0.5$, and $T_{low-level} = 0.5$. 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).

### 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


http://www.w3.org/TR/xlink/.


http://www.w3.org/TR/xptr/.