# AUTOMATIC GENERATION OF METRICAL STRUCTURE BASED ON GTTM

*Masatoshi Hamanaka*
PRESTO, Japan Science and Technology Agency
m.hamanaka@aist.go.jp

*Keiji Hirata*
NTT Communication Science Laboratories
hirata@brl.ntt.co.jp

*Satoshi Tojo*
Japan Advanced Institute of Science and Technology
tojo@jaist.ac.jp

## ABSTRACT

This paper describes an automatic system of music analysis based on the generative theory of tonal music (GTTM). The system works by acquiring the metrical structure of the music. The GTTM is considered one of the most promising theories of music in regard to computer implementation; however, no order is given for applying its rules and as a result, they frequently conflict with each other. To solve this problem, we introduced adjustable parameters, enabling us to assign priority to the rules. Our GTTM-based system makes it possible to construct hierarchical metrical structures in a top-down process using bottom-up detection of local metrical strength. Experimental results showed that after these parameters were tuned, our method outperformed the baseline performance.

## 1. INTRODUCTION

We are developing a system of music analysis based on the generative theory of tonal music (GTTM) [1]. The GTTM is composed of four modules, each of which assigns a separate structural description to a listener's understanding of a piece of music. These four modules output a grouping structure, metrical structure, time-span tree, and prolongational tree, respectively. Our goal is to automatically derive a time-span tree that assigns a hierarchy of 'structural importance' to the notes of a piece of music (Figure 1). Automatic generation of a time-span tree from the music surface enables us to analyse the deeper structure [2]. It will also provide music summarization [3] and collaborative music creation [4] systems that will enable users to manipulate music by using a time-span tree, disregarding the surface structure of the music.

To derive a time-span tree, we use the grouping and metrical structures of the music. In previous work [5], we developed a system for automatically generating a grouping structure. In this paper, we describe a method for automatically articulating the transition of notes into a metrical structure.

Previous methods based on beat tracking [6, 7] are only able to acquire the hierarchical metrical structure in a measure because they do not consider larger metrical structure such as two measures, four measures, and so on. We have previously attempted to implement several rules of the GTTM, but when these rules conflicted we were unable to adequately resolve the priority of multiple rules [8, 9]. Although a computer model of GTTM [10] is capable of producing a time-span tree, we, human, need to choose applicable rules at each stage of processing.

Our system of metrical analysis based on the GTTM makes it possible to construct a hierarchical metrical structure automatically by iterating calculation of low-level beat strength and choosing the next level structure.

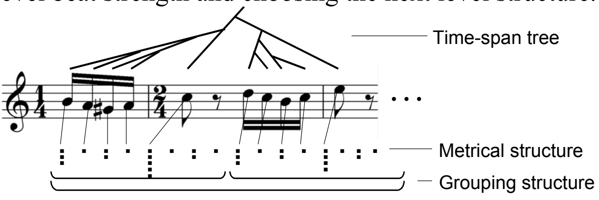

**Figure 1**. Time-span tree, metrical structure, and grouping structure.

## 2. METRICAL STRUCTURE ANALYZER BASED ON GTTM

The metrical structure describes the rhythmical hierarchy of a piece of music by identifying the position of strong beats at the levels of a quarter note, half note, a measure, two measures, four measures, and so on. Strong beats are illustrated as several levels of dots below the music staff (Figure 1).

Figure 2 shows the processing flow of the system. As the primary input formats we chose MusicXML [11] and GroupingXML [5]. A hierarchical metrical structure was constructed in the top-down way, while the lower-level beat strength are detected in the bottom-up way. We then designed MetricalXML as the export format

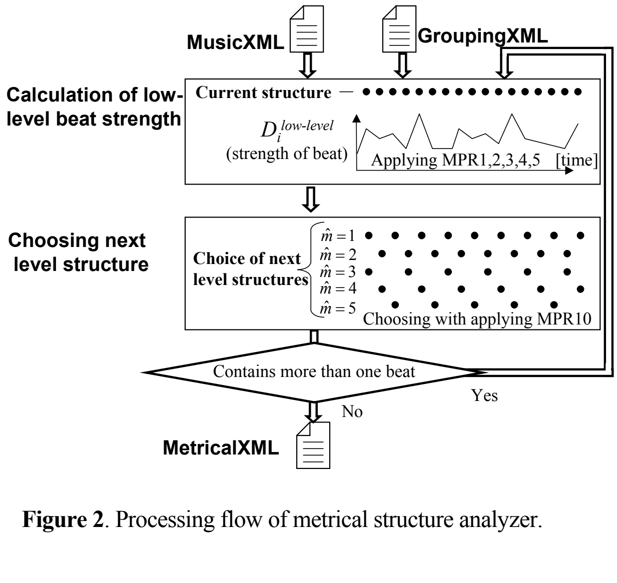

**Figure 2**. Processing flow of metrical structure analyzer.

for our system; note elements in MetricalXML are connected to note elements in MusicXML and GroupingXML by Xlink [12] and Xpointer [13] (Figure 3). In our experiments, we restricted the music structure to monophony so we could more accurately assess the performance of each rule.

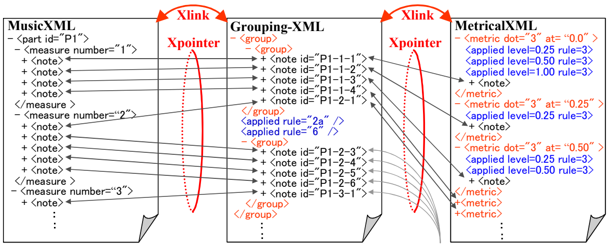

**Figure 3**. MusicXML, GroupingXML, and Metrical XML.

## 2.1. Application of MPRs

There are two types of rules in the GTTM, i.e., "well-formedness" and "preference" rules. Well-formedness rules are necessary conditions for the assignment of a structure and restrictions on the structure. When more than one structure satisfies the well-formedness rules, the preference rules indicate the superiority of one structure over another.

The metrical preference rules (MPRs) consist of 10 rules: MPR1 (parallelism), MPR2 (strong beat early), MPR3 (event), MPR4 (stress), MPR5 (length), MPR6 (bass), MPR7 (cadence), MPR8 (suspension), MPR9 (time-span interaction), and MPR10 (binary regularity). MPR5 has six cases: (a) pitch-event, (b) dynamics, (c) slur, (d) articulation, (e) repeated pitches, and (f) harmony.

Here we discuss the application of MPR1, 2, 3, 4, 5, and 10. The strength of the beat dependent on each rule can be expressed as $D_i^{\mathrm{MPR}\,j\ (=1,2,3,4,5a,5b,5c,5d,5e)}$ $(0 \leqq D_i^{\mathrm{MPR}\,j} \leqq 1)$. Because there is no strict order for applying MPR rules, ambiguities in the analysis may result. To solve this problem, we use adjustable parameters $S^{MPR\ j\ (=1,2,3,4,5a,5b,5c,5d,5e,10)}$ that enable us to control the strength of each rule. Figure 4 indicates the relationship between the parameters and MPRs. Our metrical structure analyzer has sixteen adjustable parameters, which include $S^{MPR\,j}$, $W_r$, and $T^{MPR\,j}$ (Table 1).

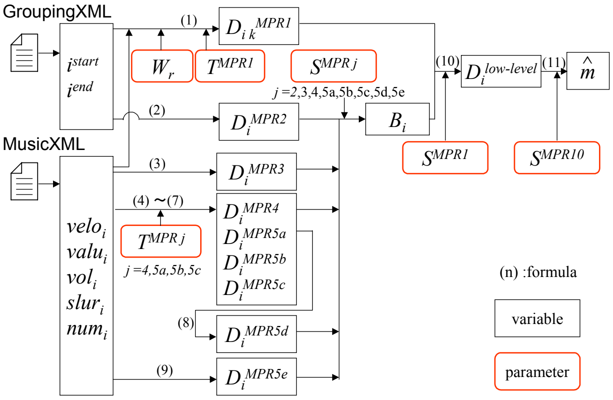

**Figure 4**. Relationship between parameters and MPRs.

| Parameters | Description |
|---|---|
| $S^{MPR\,j}$ | Strength of each rule.<br>$j$ =(1, 2, 3, 4, 5a, 5b, 5c, 5d, 5e, 10) |
| W$_r$ | Weight of priority of the same rhythm compared with the same register in parallel segments. |
| $T^{MPR\,j}$ | Value of the threshold that decides whether or not each rule is applicable. $j$ =(1, 4, 5a, 5b, 5c) |

**Table 1.** Sixteen adjustable parameters.

### *2.1.1. Calculation of basic parameters*

As calculated from MusicXML, the five basic parameters of a note from beat $i$ are: $velo_i$ (velocity), $valu_i$ (length of note), $vol_i$ (duration of dynamic[1]), $slur_i$ (length of slur), and $num_i$ (pitch). $\mu_{velo}$, $\mu_{valu}$, $\mu_{vol}$, $\mu_{slur}$, and $\mu_{num}$ are the average of the basic parameters. As calculated from GroupingXML, the two basic parameters $i^{start}$ and $i^{end}$ are the beginning and ending of a group that is the smallest group containing $i$ and more than one beat in the current structure.

### *2.1.2. Application of MPR1*

MPR1, which is the rule for parallelism in a metrical structure, has two adjustable parameters: W$_r$ (weight of priority of the same rhythm compared with the same register in parallel segments), and $T^{\mathrm{MPR1}}$ (threshold that decides whether beat $i$ and beat $k$ are parallel, i.e., ($D_{i\,k}^{\mathrm{MPR1}}$=1) or not ($D_{i\,k}^{\mathrm{MPR1}}$=0)).

$$D_{i\,k}^{MPR1} = \begin{cases} 1 & \dfrac{y_{i\,k}}{x_{i\,k}} \times W_r + \dfrac{z_{i\,k}}{y_{i\,k}} \times \left(1 - W_r\right) > T^{MPR1} \\ 0 & else \end{cases} \qquad (1)$$
,

where

$$x_{i\,k} = \sum_{i'=i^{start}}^{i^{end}} \begin{cases} 1 & velo_{i'} > 0 \\ 0 & velo_{i'} = 0 \end{cases} + \sum_{k'=k^{start}}^{k^{end}} \begin{cases} 1 & velo_{k'} > 0 \\ 0 & velo_{k'} = 0 \end{cases}$$

$$y_{i\,k} = \sum_{i'=i^{start}}^{i^{end}} \begin{cases} 1 & velo_i > 0 \ \ and \ \ velo_{k+i'+i} > 0 \\ 0 & else \end{cases}$$

$$z_{i\,k} = \sum_{i'=i^{start}}^{i^{end}} \begin{cases} 1 & velo_i > 0 \ and \ num_{i'\text{-}1} = num_{i'} \ \ and \ \ num_{k+i'-i\text{-}1} = num_{k+i'-i} \\ 0 & else \end{cases}$$
.

### *2.1.3. Application of MPR2 and 3*

MPR2 has a weak preference for a metrical structure in which the strongest beat in a group appears relatively early in the group. We formalized $D_i^{\mathrm{MPR2}}$ so that the closer it is to the beginning of the group, the higher the values. MPR3 prefers a metrical structure in which the inceptions of pitch-events are strong beats (Figure5).

$$D_i^{MPR2} = \left(i^{\ end} - i\right) / \left(i^{\ end} - i^{\ start}\right) \qquad (2)$$

$$D_i^{MPR3} = \begin{cases} 1 & velo_i > 0 \\ 0 & velo_i = 0 \end{cases} \qquad (3)$$

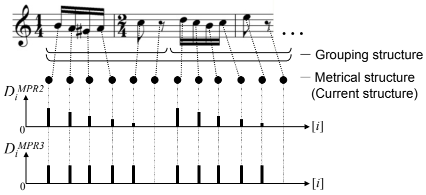

**Figure 5**. Application of MPR2 and 3.

[1] The GTTM does not define the meaning of 'duration of dynamic'. Therefore, we define it as the length from one beat to the next beat or rest, depending on the example of applying MPR5b in the GTTM [1].

### *2.1.4. Application of MPR4, 5a, 5b, and 5c*

MPR4, 5a, 5b, and 5c prefer a metrical structure in which a relatively strong beat occurs at the inception of either a stressed note, a long note, a long duration of a dynamic, or a long slur. We introduced the adjustable parameters $T^{MPR\,j(=4,\,5a,\,5b,\text{ and }5c)}$ ($0 \leqq T^{MPR\,j} \leqq 1$) to control the value of the threshold that decides whether each rule is applicable, i.e., ($D_i^{MPRj}=1$) or not ($D_i^{MPRj}=0$) (Figure 6).

$$D_i^{MPR4} = \begin{cases} 1 & velo_i > 2\times\mu_{velo}\times T^{MPR4} \\ 0 & else \end{cases} \quad (4)$$

$$D_i^{MPR5a} = \begin{cases} 1 & valu_i > 2\times\mu_{valu}\times T^{MPR5a} \\ 0 & else \end{cases} \quad (5)$$

$$D_i^{MPR5b} = \begin{cases} 1 & vol_i > 2\times\mu_{vol}\times T^{MPR5b} \\ 0 & else \end{cases} \quad (6)$$

$$D_i^{MPR5c} = \begin{cases} 1 & slur_i > 2\times\mu_{slur}\times T^{MPR5c} \\ 0 & else \end{cases}$$

(7)

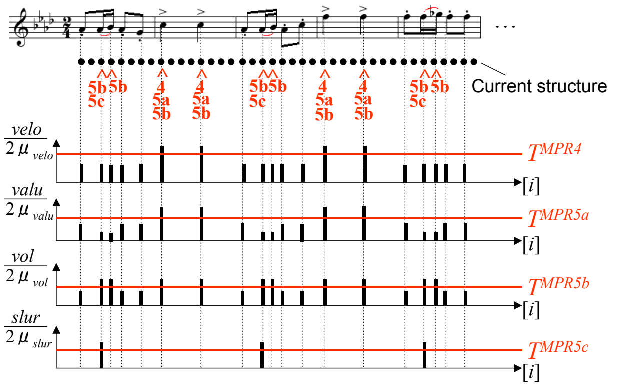

**Figure 6**. Application of MPR4, 5a, 5b and 5c.

### *2.1.5. Application of MPR5d and 5e*

MPR5d is the rule for repetition of an articulation pattern. We apply MPR5d to those positions where there are inceptions of MPR5a repetition. MPR5e is the rule for pitch-repetition.

$$D_i^{MPR5d} = \begin{cases} 1 & D_i^{MPR5a} = 1 \text{ and } D_{i+1}^{MPR5a} = 1 \\ 0 & else \end{cases} \quad (8)$$

$$D_i^{MPR5e} = \begin{cases} 1 & num_i = num_{i+1} \\ 0 & else \end{cases} \quad (9)$$

## 2.2. Calculation of low-level beat strength

Low-level beat strength is calculated by weighted summation of $D_i^{\mathrm{MPR}\,j\,(=1,2,3,4,5a,5b,5c,5d,5e)}$.

$$D_i^{low\text{-}level\ metrical} = B_i + \sum_k \begin{cases} B_k \times S^{MPR1} & D_{i\,k}^{MPR1} = 1 \\ 0 & D_{i\,k}^{MPR1} = 0 \end{cases}, \quad (10)$$

where

$$B_i = \sum_{j=(2,3,4,5a,5b,5c,5d,5e)} D_i^{MPR\,j} \times S^{MPR\,j}.$$

## 2.3. Acquisition of hierarchical metrical structure

A hierarchical metrical structure is constructed by iterating calculation of the low-level strength of the beat for the current structure and choosing the next level structure (Figure 7). When the current structure contains more than one beat, the next level structure $\hat{m}$ is calculated as follows:

$$\hat{m} = \underset{m(=1,2,3,4,5)}{argmax} \sum_i \begin{cases} D_i^{low\text{-}level\,metrical} & (i-m)\,mod\,2 = 0 \\ D_i^{low\text{-}level\,metrical} \times S^{MPR10} & (i-m)\,mod\,3 = 1 \\ 0 & else \end{cases} \quad (11)$$

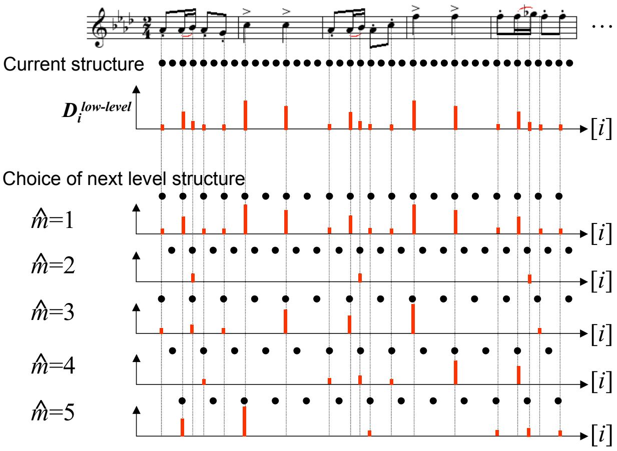

**Figure 7**. Selecting the next level structure.

## 3. GTTM EDITOR

We developed a GTTM editor to construct grouping and metrical structures, and time-span trees. The GTTM editor has two modes as follows.

### 3.1. Automatic-analysis mode

The automatic-analysis mode displays the results of our grouping and metrical structure analyzer (Figure 8). The structure changes depending on the parameters configured.

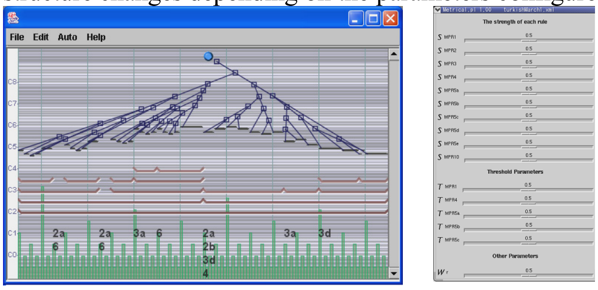

**Figure 8**. GTTM editor (automatic-analysis mode).

### 3.2. Manual-edit mode

The manual-edit mode assists in manual editing of the grouping structure, metrical structure, and time-span tree (Figure 9). It can be used to edit the results of our grouping and metrical structure analyzer in automatic-analysis mode.

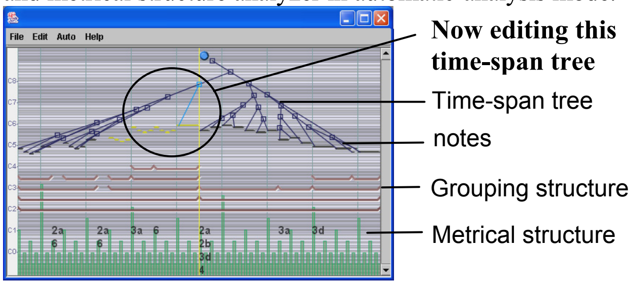

**Figure 9**. GTTM editor (manual-edit mode).

## 4. EXPERIMENTAL RESULTS

We evaluated the performance of our metrical structure analyzer using an *F*-measure, which is given by the weighted harmonic mean of *Precision P* (the proportion of selected metrical dots that are correct) and *Recall R* (the proportion of correct metrical dots that were identified). We did not take care that low-level error is propagated up to higher-level; we counted wrong answers in each metrical level.

$$F_{measure} = 2 \times \frac{P \times R}{P + R} \tag{12}$$

This evaluation required us to prepare accurate data for the metrical structure. We collected one hundred pieces of 8-bar length, monophonic, classical music, and asked people with expertise in musicology to analyze them manually with faithful regard to the MPRs using the GTTM editor. These manually produced results were cross-checked by three other experts.

To evaluate the baseline performance of our system, we used the following default parameters: $S^{MPR\,j}$=0.5, $W_r$=0.5, and $T^{MPR\,j}$=0.5. The average *F*-measure for the baseline performance was 84%. We took about 10 minutes in average for finding a plausible tuning of the parameter set by hand. As a result of configuring the parameters, the *F*-measure of our metrical structure analyzer reached 90%, outperforming the baseline by 6% (Table 2).

| Melody | Baseline performance | Our method with configured parameters |
|---|---|---|
| 1. Spinnerlied | 0.88 | 1.00 |
| 2. Petit Chien | 0.86 | 0.92 |
| 3. Solvajg's Song | 0.86 | 0.86 |
| 4. L'Arlesienne | 0.92 | 0.98 |
| 5. Tarantella | 0.63 | 0.69 |
| 6. The Moldau | 0.73 | 1.00 |
| 7. Tristesse | 0.88 | 1.00 |
| 8. Waves of the Danube | 0.72 | 0.97 |
| 9. Barcarolle | 0.72 | 0.79 |
| 10. Gymnopedie | 0.66 | 0.82 |
| : | : | : |
| : | : | : |
| Total (100 melodies) | 0.84 | 0.90 |

**Table 2.** *F*-measure for our method

## 5. CONCLUSION

We developed a system for analyzing the metrical structure of music based on the GTTM. The system makes it possible to construct a hierarchical metrical structure. Our experimental result showed that the *F*-measure reached 90%, that outperforms the baseline measure by 6%, with optimally configured sets of the parameters. At present, time-span trees are generated manually using only MPRs together with grouping-structure information provided by GroupingXML. We are now planning to improve the precision of the trees, implementing further details of the time-span tree generation rule.

## 6. REFERENCES

[1] Lerdahl, F., and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, Massachusetts, 1983.

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

[3] Hirata, K., and S. Matsuda. "Interactive Music Summarization based on Generative Theory of Tonal Music", *Journal of New Music Research, 32:2, 165-177.* 2003.

[4] Hirata, K., S. Matsuda, K. Kaji, and K. Nagao. "Annotated Music for Retrieval, Reproduction, and Sharing", *Proceedings of the International Computer Music Conference*, Miami, USA, 2004.

[5] Hamanaka, M., K. Hirata, and S. Tojo. "Automatic Generation of Grouping Structure based on the GTTM", *Proceedings of the International Computer Music Conference*, Miami, USA, 2004.

[6] Rosenthal, D. "Emulation of human rhythm perception", *Computer Music Journal 16(1), 64–76.* 1992.

[7] Goto, M. "An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds", *Journal of New Music Research, 30:2, 159-171.* 2001.

[8] Ida, K., K. Hirata, and T. Satoshi. "The Attempt of the Automatic Analysis of the Grouping Structure and Metrical Structure based on GTTM", *SIG Technical Report* 2001*(42):49-54,* Japan, 2001, (in Japanese).

[9] Touyou, T., K. Hirata, and T. Satoshi. "Improvement of Grouping Rule Application in Implementing GTTM", *SIG Technical Report* 2002*(47):121-126,* Japan, 2002, (in Japanese).

[10] Nord, T. *Toward Theoretical Verification: Developing A Computer Model of Lerdahl and Jackendoff's Generative Theory of Tonal Music*, Dissertation, The University of Wisconsin, Madison, 1992.

[11] Recordare LLC. "*MusicXML 1.0 Tutorial*" http://www.recordare.com/xml/musicxml-tutorial.pdf., 2004.

[12] W3C. "*XML Linking Language (XLink) Version 1.0.* " http://www.w3.org/TR/xlink/, 2001.

[13] W3C. "*XML Pointer Language (XPointer).* " http://www.w3.org/TR/xptr/, 2002.