

Algebraic Mozart by Tree Synthesis

Keiji Hirata

Future University Hakodate
hirata@fun.ac.jp

Satoshi Tojo

JAIST
tojo@jaist.ac.jp

Masatoshi Hamanaka

Kyoto University
masatosh@kuhp.kyoto-u.ac.jp

ABSTRACT

Thus far, we have been automatizing the time-span analysis of Jackendoff and Leirdahl's Generative Theory of Tonal Music (GTTM). We have also introduced the distance between two time-span trees and verified by an experiment that the distance was properly supported by the psychological similarity. In this paper, we synthesize a new piece of music using the algebraic operations on time-span trees, with this notion of distance. For this process, we need an operation to retain a certain number of pitch events as well as reduction, then we employ *join* operation on two input pieces of music. But, the result of the *join* operation is not obvious as two or more pitch events may occupy the same position on a score in a conflicting way. Therefore, in this research, we distinguish the tree representation from actual music written on a score and define *join* and *meet* in the domain of the tree representation in the algebraic manner. Then, to demonstrate the validity of our approach, we compose artificial variations of K.265/300e by Wolfgang Amadeus Mozart by a morphing technique using *join* and *meet*. We examine the results with human intuitive similarity and show that algebraic operations such as *join* and *meet* suffices to produce viable Mozartoid variations.

1. INTRODUCTION

The main aim of conventional music theories is analyzing and understanding music, not composing. Although there have been various attempts at applying conventional music theories to composition [7], Roads pointed out the difficulty in these attempts as follows [9, p.909]:

The surface of any music can be encoded into such rules. But no one would mistake the logic of a style template as anything resembling the actual process of human composition. . . . Emotional involvement is inseparable from musical behavior of all kinds, yet there have been only a few attempts to consider affect as part of a model of compositional thought . . . A model that relates musical structure to its emotional significance, however crude, may lessen the disparity that ex-

ists between our experience of music and the rationalizations we use to specify it.

We have been investigating the algebraic framework for manipulating music pieces under the principle that reduction corresponds to the partial order. Among music theories that have been proposed so far, we think that the *time-span tree* introduced by Leirdahl and Jackendoff's Generative Theory of Tonal Music (GTTM; hereafter) [5] is suitable for the domain in which we formalize reduction. Let us consider the time-span tree and reduction. The *time-span analysis* in GTTM assigns structural importance to each pitch events, derived by the *grouping analysis*, in which a sequence of notes forms a short phrase called a group, and by the *metrical analysis*, where strong and weak beats are properly assigned to each pitch event. As neighboring notes can be compared by this structural importance in the bottom-up way, the hierarchy forms a *time-span tree*, where a branch from a less important event is absorbed into that from a more important event. We illustrate this process in Fig. 1. This theory, therefore, includes the *reduction hypothesis*; in the sequence of reduction of pitch events, the original piece is simplified and is abstracted, and thus, we can retrieve a basic skeleton [6] of the original music piece¹.

Thus far, we have automatized the process of time-span analysis [1], and proposed various applications [2]. In [11], we defined a notion of distance among the time-span trees, and then we compared the tree distance with human cognitive similarity, among 12 variations of *Ah vous dirai-je, maman*, K. 265/300e by Wolfgang Amadeus Mozart [4].

In this paper, we propose a technique for creating a music piece based on our algebraic framework which is both mathematically and cognitively well-grounded. As an application of the technique, we demonstrate the composition of new variations from two existing variations, combining the two time-span trees of the variations in the algebraic manner with the *join* and *meet* operations. For *meet* as an operation to reduce uncommon pitch events, the *meet* operation is rather naturally defined as the intersection part of two music pieces. Thus, if we restrict our interest in the calculation of distance, *meet* may sufficiently serve as an edit distance such as earth mover's distance (EMD) or Rizo-Valero's [8]. For *join* as an operation to increase pitch events, in contrast, it is problematic because the *join* operation does not always function, when the two music scores contain unmatched pitch events. Here, our idea is to intro-

¹ Although a pitch event means a single note or a chord, in this paper, we restrict our interest to monophonic analysis as the method of chord recognition is not included in the original theory.

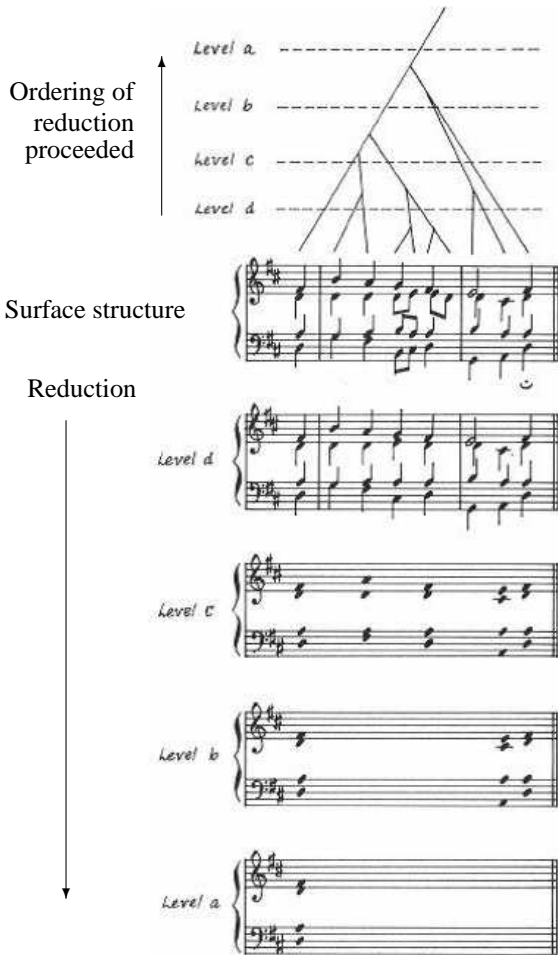


Figure 1. Reduction hierarchy of chorale ‘O Haupt voll Blut und Wunden’ in St. Matthew’s Passion by J. S. Bach [5, p.115]

duce an algebraic domain in which a virtual representation of a *join*-ed time-span tree is allowed.

This paper is organized as follows. In Section 2, we provide basic algebraic operations for time-span trees, and the notion of distance, as a background theory. In Section 3, we propose a new notion of abstract *join* by which we would represent a virtual tree, and clarify the morphing algorithm in Section 4. Next in Section 5, we actually show new variations generated by our method, and evaluate the pieces from a human psychological viewpoint. Finally, in Section 6, we mention the limitations of our method, and discuss the possibility of further development.

2. JOIN REVISITED AS A SYNTHESIS OPERATION

To provide a prerequisite for the following sections, we explain our approach, excerpting necessary definitions and properties from our previous works [4, 11].

2.1 Subsumption, *join*, and *meet*

Hereafter, we identify the reduction in trees with the subsumption relation, which is the most fundamental relation in knowledge representation. Let σ_1 and σ_2 be tree struc-

tures. σ_2 subsumes σ_1 , that is, $\sigma_1 \sqsubseteq \sigma_2$ if and only if for any branch in σ_1 there is a corresponding branch in σ_2 .

Let σ_A and σ_B be tree structures for two music pieces A and B , respectively.

join If we can fix the least upper bound of σ_A and σ_B , that is, the least y such that $\sigma_A \sqsubseteq y$ and $\sigma_B \sqsubseteq y$ is unique, we call such y the *join* of σ_A and σ_B , denoted as $\sigma_A \sqcup \sigma_B$.

meet If we can fix the greatest lower bound of σ_A and σ_B , that is, the greatest x such that $x \sqsubseteq \sigma_A$ and $x \sqsubseteq \sigma_B$ is unique, we call such x the *meet* of σ_A and σ_B , denoted as $\sigma_A \sqcap \sigma_B$.

We can define $\sigma_A \sqcup \sigma_B$ and $\sigma_A \sqcap \sigma_B$ by recursive functions. Thus, the partially ordered set of time-span trees becomes a *lattice*, where $\sigma_A \sqcup x = \sigma_A$ and $\sigma_A \sqcap x = x$ if $x \sqsubseteq \sigma_A$. Moreover, if $\sigma_A \sqsubseteq \sigma_B$, $x \sqcup \sigma_A \sqsubseteq x \sqcup \sigma_B$ and $x \sqcap \sigma_A \sqsubseteq x \sqcap \sigma_B$ for any x .

2.2 Maximal Time-Span and Reduction Distance

The *head* pitch event of a tree is the most salient event in the tree; i.e., the saliency is extended to the whole tree. As the situation is the same in each subtree, we consider that each pitch event has its maximal length of saliency, called *maximal time-span*. We hypothesize that if a branch with a single pitch event is reduced, the amount of information corresponding to the length of its maximal time-span is lost.

In Fig. 2 (a), there are four contiguous pitch events, e_1 , e_2 , e_3 , and e_4 ; each has its own temporal span (duration on surface), s_1 , s_2 , s_3 , and s_4 , denoted by thin lines. Fig. 2 (b) depicts time-span trees and corresponding maximal time-span hierarchies, denoted thick gray lines. The relationships between spans in (a) and maximal time-spans in (b) are as follows. At the lowest level in the hierarchy, the length of a span is equal to that of a maximal time-span; $mt_2 = s_2$, $mt_3 = s_3$. At the higher levels, $mt_1 = s_1 + mt_2$, and $mt_4 = mt_1 + mt_3 + s_4 = s_1 + s_2 + s_3 + s_4$. That is, every span extends itself by concatenating the span at a lower level along the configuration of a time-span tree. When all subordinate spans are concatenated up into a span, the span reaches the maximal time-span.

The distance d_{\sqsubseteq} of two time-span trees such that $\sigma_A \sqsubseteq \sigma_B$ in a reduction path is defined by

$$d_{\sqsubseteq}(\sigma_A, \sigma_B) = \sum_{e \in \zeta(\sigma_B) \setminus \zeta(\sigma_A)} s_e.$$

For example in Fig. 2, the distance between σ_1 and σ_4 becomes $mt_1 + mt_2 + mt_3$. Note that if e_3 is first reduced and e_2 is subsequently reduced, the distance is the same. Although the distance appears at a glance to be a simple summation of maximal time-spans, there is a latent order in the addition, for the reducible branches are different in each reduction step. In order to give a constructive procedure to this summation, we introduce the notion of total sum of maximal time-spans as:

$$tmts(\sigma) = \sum_{e \in \zeta(\sigma)} s_e.$$

When $\sigma_A \sqsubseteq \sigma_B$, $d_{\sqsubseteq}(\sigma_A, \sigma_B) = tmts(\sigma_B) - tmts(\sigma_A)$. As a special case of the above, $d_{\sqsubseteq}(\perp, \sigma) = tmts(\sigma)$.

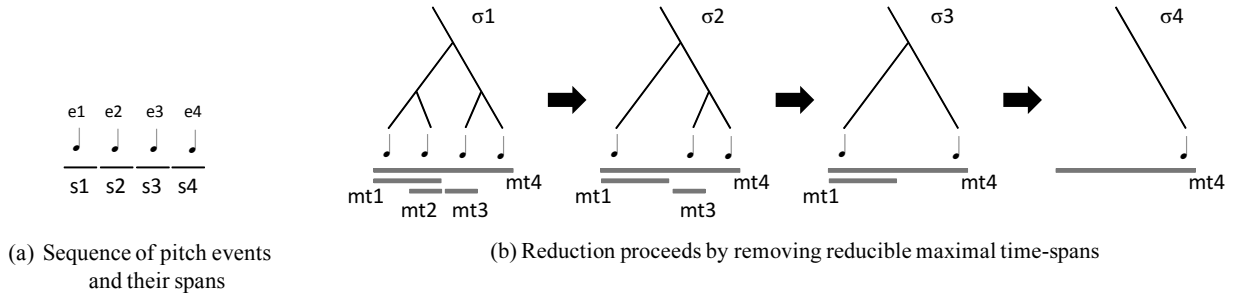


Figure 2. Reduction of time-span tree and maximal time-span hierarchy; thick gray lines denote maximal time-spans while thin ones denote pitch durations.

2.3 Requirement on Distance

As there is a reduction path between $\sigma_A \sqcap \sigma_B$ and $\sigma_A \sqcup \sigma_B$, and $\sigma_A \sqcap \sigma_B \sqsubseteq \sigma_A \sqcup \sigma_B$, $d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_A \sqcup \sigma_B)$ is unique. Here let us define two distance metrics.

$$d_{\sqcap}(\sigma_A, \sigma_B) \equiv d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_A) + d_{\sqsubseteq}(\sigma_A \sqcap \sigma_B, \sigma_B)$$

$$d_{\sqcup}(\sigma_A, \sigma_B) \equiv d_{\sqsubseteq}(\sigma_A, \sigma_A \sqcup \sigma_B) + d_{\sqsubseteq}(\sigma_B, \sigma_A \sqcup \sigma_B)$$

We immediately obtain $d_{\sqcup}(\sigma_A, \sigma_B) = d_{\sqcap}(\sigma_A, \sigma_B)$ by the uniqueness of reduction distance.

Hereafter, we omit $\{\sqcap, \sqcup\}$ from $d_{\{\sqcap, \sqcup\}}$, simply expressing it as ‘ d ’. Here, $d(\sigma_A, \sigma_B)$ is unique among the shortest paths between σ_A and σ_B . Finally, we obtain

$$d(\sigma_A, \sigma_B) + d(\sigma_B, \sigma_C) \geq d(\sigma_A, \sigma_C),$$

which is the triangle inequality. For more details on the theoretical background, see [11].

2.4 Framework for Music Synthesis

To synthesize a new piece of music, one may plan to use *meet* to reduce and *join* to increase the number of pitch events from two concrete music scores. In actual fact, *meet* mostly works well, while the result of *join* is, however, often not obvious as two or more pitch events may occupy the same position on a score in a conflicting way. Therefore we propose to provide a virtual *join* representation, not for concrete music score, but for time-span trees, to apply it to the morphing, as described in the following section.

Here, we state that the time-span tree representation should be strictly distinguished from the actual music represented on scores (Figure 3). The left-hand image in Figure 3 refers to the algebraic domain which we mentioned in preceding subsections. On the contrary, the right-hand side of the figure refers to the domain of actual music. To go from a tree representation to a concrete music score, we need another process of *music rendering*, which is independent of the process of analysis from music scores to trees [1]. At the same time, however, this implies that we do not need to concern ourselves with the actual image of music in these algebraic operations.

Instead of an algebraic lattice where *meet* and *join* exist uniquely, we need to specify the requirements for the tree representation of *join*; we should summarize this as follows:

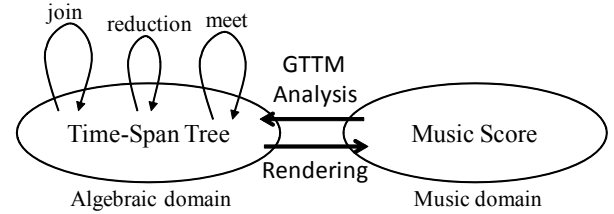


Figure 3. The Proposed Framework For Music Synthesis

Absorption Law $(\sigma_A \sqcup \sigma_B) \sqcap \sigma_A = \sigma_A$ and $(\sigma_A \sqcap \sigma_B) \sqcup \sigma_A = \sigma_A$.

Parallelism of distance $d_{\sqcup}(\sigma_A, \sigma_B) = d_{\sqcap}(\sigma_A, \sigma_B)$

We can easily confirm that these two conditions ensure the uniqueness of *join*.

3. REPRESENTATION OF TIME-SPAN TREE

In this and the following sections, we present new contributions of the paper.

Thus far, *join* and *meet* have only been applicable to unifiable pairs of trees, in the sense of branch configuration. If we could amend the definitions of these, preserving the two requirements mentioned in Section 2.4, the applicability would be greatly improved. If we could provide the *join* and *meet* operations satisfying the absorption law and the parallelism of distance in the previous section, the applicability of the operations would greatly increase, and we could design more varieties of musical application. Thus, we propose a new time-span tree representation and improved *join* and *meet* operations for it.

3.1 Ternary Branching Representation

In Section 3, we have proposed the framework in which a time-span tree is distinguished from a written score. Now, disregarding *join* of two melodies on a score, we introduce a ternary-branching tree, which represents the superimposition of the left-branching and right-branching binary trees. A new representation for a time-span tree is introduced, shown in BNF as follows:

$$\langle n \rangle ::= p \mid c(\langle n \rangle, \langle t \rangle, \langle t \rangle)$$

$$\langle t \rangle ::= \perp \mid p \mid c(\langle n \rangle, \langle t \rangle, \langle t \rangle)$$

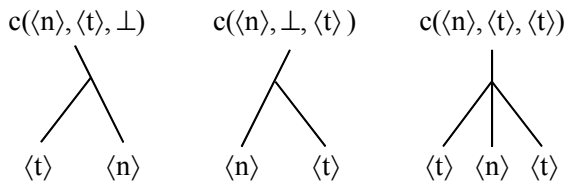


Figure 4. Three Node Forms in Novel Representation of Time-Span Tree

Symbol p means a pitch event as a terminal symbol, and \perp the bottom which means the identify element for the *join* operation. Pitch event p contains the information of pitch, maximal time span, and corresponding note on a score. $\langle n \rangle$ and $\langle t \rangle$ stand for a time-span tree; $\langle t \rangle$ can be \perp , while $\langle n \rangle$ is not. \perp may occur only at the second or third place, not at the first. Term $c(\langle n \rangle, \langle t \rangle, \langle t \rangle)$ represents a node of a time-span tree; the first place of the term $\langle n \rangle$ represents a primary branch, the second place (first $\langle t \rangle$) a secondary left branch, and the third place (second $\langle t \rangle$) a secondary right branch (Fig. 4).

The idea here is that node $c(\langle n \rangle, \langle t \rangle, \langle t \rangle)$ may be synthesized by the joining of unmatched-branching trees and joining with fully-instantiated tree $c(\langle n \rangle, \langle t \rangle, \langle t \rangle)$. The new tree representation enables the *join* operation to yield a proper result for those cases which have thus far not been unifiable. The joining of unmatched-branching trees comprises cases such as $join(c(\langle n \rangle, \langle t \rangle, \perp), c(\langle n \rangle, \perp, \langle t \rangle))$ (the upper part of Fig. 5) and $join(c(\langle n \rangle, \langle t \rangle, \langle t \rangle), c(\langle n \rangle, \perp, \perp))$; joining with fully-instantiated tree $c(\langle n \rangle, \langle t \rangle, \langle t \rangle)$ comprises cases such as $join(c(\langle n \rangle, \langle t \rangle, \langle t \rangle), c(\langle n \rangle, \langle t \rangle, \langle t \rangle))$ and $join(c(\langle n \rangle, \langle t \rangle, \langle t \rangle), c(\langle n \rangle, \perp, \langle t \rangle))$. Simply, the *join*

operation recursively computes the argument-wise *join*. The ternary branching representation can be regarded as the superposition, abstracting the distinction of left-/right-branching, of a binary tree, not as a node having three branches.

Moreover, the lower part of Fig. 5 shows the calculation of *meet* in one of the formerly-nonviable cases. Similarly, the *meet* operation recursively computes the argument-wise *meet*. Thus, in this case, the *meet* operation takes into account only the primary branches, ignoring secondary branches, which is equivalent to the treatment in the previous research [4].

Note that the ternary-branching tree representation introduced here is distinguished from a ternary branching time-span tree which may occur in ternary meter². The ternary-branching appears only when we calculate *join* operation tentatively. There is still the necessary condition that we are able to calculate the *join* operation, which is a joined maximal time-span being concatenated, otherwise the result is undefined. Let $[b, e]$ be a time-span beginning at b and ending at e ; we may assume the *join* of $[1, 3]$ and $[2, 4]$ would be the connected interval of $[1, 4]$ while that of $[1, 2]$ and $[3, 4]$ would remain as two separated intervals. Incidentally, the *meet* of $[1, 3]$ and $[2, 4]$ is $[2, 3]$, and that of $[1, 2]$ and $[3, 4]$ is undefined, not as \perp .

3.2 Theoretical Properties

To introduce the proper *join*, we assume some useful concepts of the time-span tree beforehand.

Definition 1 (Structural Equivalence) Given a node c in a time-span tree representation,

$$c(p, \perp, \perp) \equiv p$$

where p is atomic, either a pitch event or \perp

It follows that \perp is equivalent to $c(\perp, \perp, \perp)$, $c(c(\perp, \perp, \perp), \perp, \perp)$, $c(\perp, c(\perp, \perp, \perp), \perp)$, and so on. As a result, there are an infinite number of such trees equivalent to \perp . For example in the lower part of Fig. 5, let t be a tree, then $c(t, \perp, \perp)$ cannot be rewritten to t if t is not atomic. Suppose p_i means a pitch event, then $c(c(p_1, \perp, \perp), \perp, p_2)$ can be rewritten to $c(p_1, \perp, p_2)$.

As we have extended the new representation of time-span tree with ternary branching node c and the structural equivalence rule, we can similarly extend all the definitions on reduction path, reduction distance, total maximal time-span, and the lemmas on uniqueness of reduction distance that we have developed in Sections 2.1, 2.2 and 2.3. Finally, we can prove the theorem on triangle inequality of distance with the new representation of time-span tree, although we would like to omit the details of the definitions and the proofs of the lemma and the theorem.

We show an example in which given two pieces, the *join* and *meet* are calculated (Fig. 6). The two pieces are taken from the Mozart's variations K.265/300e 'Ah, vous dirai-je, maman', the variations No.2 and No.5. Actually, in the process of calculating the *join* and *meet* operations of

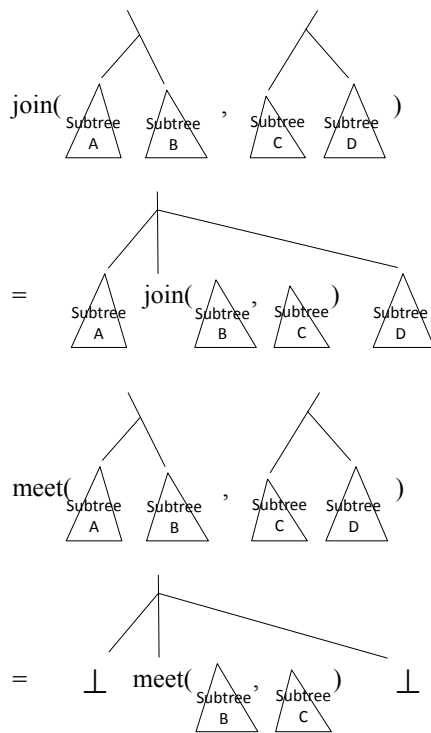


Figure 5. Join and Meet of Unmatched-Branching Trees

² Since GTTM restricts a time-span tree to a binary tree, a ternary branching time-span tree is not allowed [5, pp.326-330].

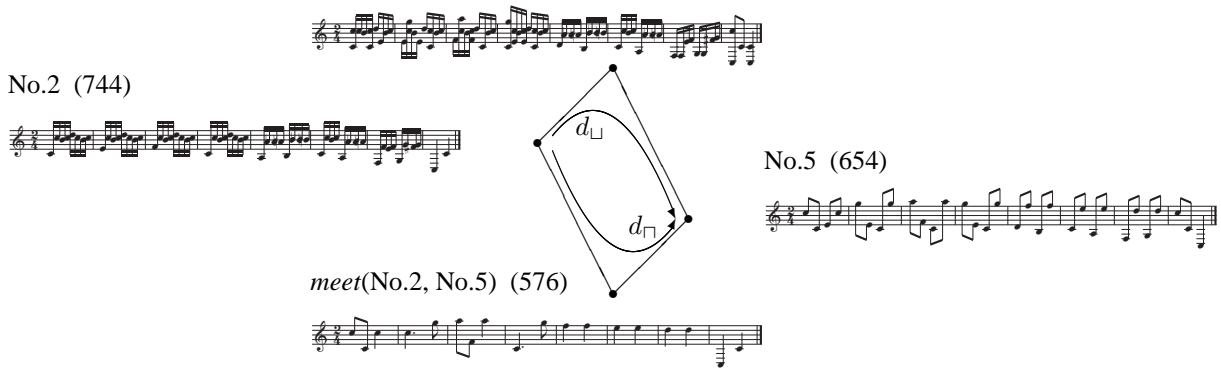


Figure 6. Parallelogram Composed of Variations No.2 and No5, *join* and *meet*. The values in the parentheses are total maximal time-spans.

these two time-span trees, *join* and *meet* of unmatched-branching ones occur nine times, respectively, and the distances via *join* and *meet*, d_{\perp} and d_{\lrcorner} , are the same. The value in the parenthesis shows the total maximal time-span of each time-span tree; according to the definition of distance, we obtain $d_{\perp} = (822 - 744) + (822 - 654) = 246$ and $d_{\lrcorner} = (744 - 576) + (654 - 576) = 246$. Notice that the four time-span trees form a parallelogram because the lengths of the opposite sides are equal respectively. Then, we have confirmed the lemma on uniqueness of reduction distance in the proposed framework.

4. MORPHING ALGORITHM

Morphing is an algorithm to find an intermediate graphical image, given two images. We provide a similar methodology to compose an intermediate piece of music, given two music pieces; especially given two existing variations with a common theme as in the paper [4]. Let σ_A and σ_B be two pieces of music, and σ_C be an expected result of morphing; we require that σ_C should reside at an internally dividing point of σ_A and σ_B by $N:M$. The ratio $M:N$ means the one in terms of the total sum of maximal time-spans (denoted as *tmts* in Section 2.2). Notice that there are infinitely many σ_C 's such that the ratio of the distance between σ_A and σ_C to that between σ_C and σ_B is $M:N$ because σ_C resides on so-called Apollonian circles. Thus, we should restrict σ_C to the one that resides at the shortest distances from σ_A and σ_B , respectively.

Our morphing algorithm is shown in Fig. 7, consisting of:

1. Find such a reduction α of σ_A that divides σ_A and $meet(\sigma_A, \sigma_B)$ by the ratio of $N:M$ in terms of the given distance.
2. Find such a reduction β of σ_B that divides σ_B and $meet(\sigma_A, \sigma_B)$ with the ratio of $M:N$.
3. *join* α and β .

We see that four time-span trees α , β , $meet(\sigma_A, \sigma_B)$, and $join(\alpha, \beta)$ also form a parallelogram as in Fig. 6. Apparently, in terms of the distance between σ_A and σ_B , we have $d(\sigma_A, \sigma_B) = d(\sigma_A, join(\alpha, \beta)) + d(join(\alpha, \beta), \sigma_B)$.

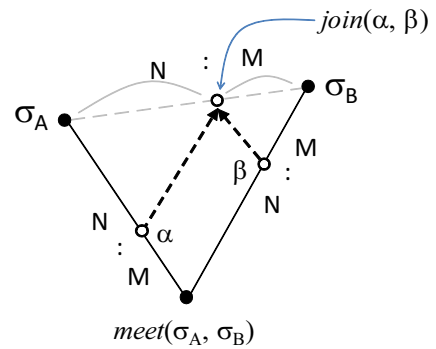


Figure 7. Morphing Algorithm

Moreover, $tmts(\sigma_A) \leq tmts(join(\sigma_A, \sigma_B)) \leq tmts(\sigma_B)$ holds if $tmts(\sigma_A) \leq tmts(\sigma_B)$.

We mention three points in implementing the morphing algorithm. The first is related to the fact that our current framework disregards matching of pitch events; the reduction operation takes only the information of the configuration of time-spans. Although we omit the technical details, for obtaining the appropriate values of α and β , we prefer to avoid the ratio $N:M$ near to $1:0$ or $0:1$.

The second is related to rendering of the fully-instantiated node $c(\langle n \rangle, \langle t \rangle, \langle t \rangle)$, which can be regarded as the superimposition of the differently-branching nodes of two binary trees, not as a node having three branches. In the current implementation, a fully-instantiated node is simply rendered as a chord of two notes, that is, sounding both at the same time. Otherwise, for instance, it could be rendered as a transformation of the superimposed time-spans³.

The third is rendering itself. In the present rendering algorithm, a maximal time-span is basically considered as a line segment in a piano roll score, and the time-spans at a lower level (closer to leaves) overwrites those at a higher level. Thus, it may occur that the entirety of the maximal time-span is overwritten by the lower-level pitch events; that is, even though a pitch event is quite salient, that pitch event may become inaudible, or its duration assigned on a real score may be very short.

³ It is like a transformation head [5, p.155].

No.1



No.1&No.2



No.2



No.2&No.5



No.5



No.5&No.1



Figure 8. Variations No.1, No2, and No.5, and morphed melodies between them

5. EXPERIMENT AND RESULTS

The morphing algorithm is implemented in SWI-Prolog [10]. The set piece is Mozart’s variations K.265/300e ‘Ah, vous dirai-je, maman’. The piece consists of the famous theme and twelve variations of it. In our experiment, we take variations No.1, 2, and 5 as the sources for morphing, and excerpt the first eight bars (Fig. 8). We have chosen these three variations because for every pair of these two we can calculate the result of *join*, that is, joined maximal time-spans are all concatenated. To make comparison easy, the morphed melodies generated by the improved algorithm are shown between the variations. For example, in the figure, “No.2&No5” means the morphed melody at the midpoint of variations No.2 and 5.

For the similarity assessment of the morphed melodies by human listeners, six university students (2 females and 4 males) participated in our study, four of whom have experience of playing music instruments for five years or more. We use the method similar to the previous research[4]. An examinee listens to all pairs $\langle m_1, m_2 \rangle$ in random order without duplication, where $m_{\{1,2\}}$ is either variations No.1, No.2, No.5 and the morphed melodies between them, such as No.1&No.2. Every time he/she listens to it, he/she is asked “how similar is m_1 to m_2 ?”, and rates it using one of following five grades: *quite similar* = 2, *similar* = 1, *neutral* = 0, *not similar* = -1, and *quite different* = -2. At the very beginning, for cancelling the cold start bias, every examinee hears the theme and twelve variations (eight bars long) without rating them. In addition, when an examinee listens to and rates pair $\langle m_1, m_2 \rangle$, he/she should try the same pair later to avoid the order effect. Finally, the average ratings of each examinee are calculated and then the average for all the examinees is determined.

The experimental results are obtained in the distance-ma-

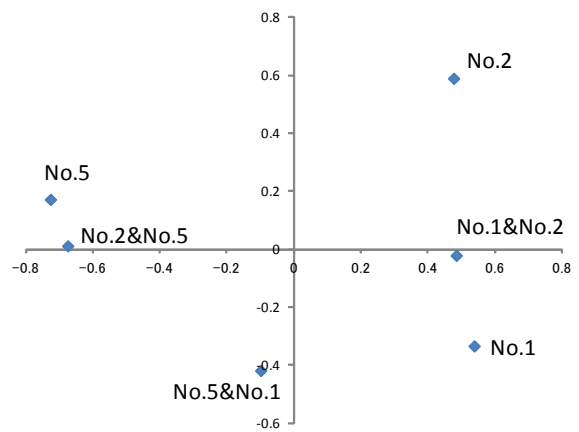


Figure 9. Relative Distance Among Variations and Morphed Melodies According to the Impression of Human Listeners

trix between variations No.1, No.2, No.5 and the morphed melodies between them at first. Since it is difficult to examine the results as they are, we employ multidimensional scaling (MDS) [12] to visualize the results (Fig. 9). To explain briefly, MDS plots items on a coordinate plane so that the more similar items are, the closer they are.

In terms of Nos. 1 and 2 pair and Nos. 1 and 5 pair, the morphed melodies are plotted at the midpoint of their source variations almost as expected. In contrast, the position of No.2&No.5 is problematic. As can be seen in Fig. 8, No.2&No.5 is the internally dividing point of No.2 and No.5 by 1: 1, and the number of notes of No.2&No.5 is approximately the average of No.2 and No.5. However, No.2&No.5 is almost entirely made of eighth notes, and as the result of *join*, many of the notes which have the same pitch or which sound at the same time. Consequently, it can be thought that the human impression of No.2&No.5 is closer to that of No.5.

6. CONCLUSIONS

In this paper, we have proposed the time-span tree representation and the *join* operation, applied to two time-span trees. In general, the result of the *join* operation on two arbitrary input pieces of music is not obvious. That is, it is not straightforward to construct the valid *join* satisfying the basic properties such as the absorption law that is consistent with the notion of reduction provided by GTTM. We explained that we strictly distinguished the tree representation from the actual music represented on scores. By use of the *join* and *meet* operations, we implemented an automatic morphing system in Prolog, and composed virtual variations of K.265/300e by Wolfgang Amadeus Mozart from existing variations. Since the distance between time-span trees defined in the paper satisfies the properties desired for morphing, we can identify the internal dividing point of time-span trees σ_A and σ_B by $N : M$ as if we draw a figure using a triangle ruler and a compass (Fig. 7). We have evaluated these synthesized variations according to the impression of human listeners, and

found an interesting correspondence between the theoretical distance and psychological distance. As a result, we have shown that the use of *join* and *meet* operations in our algebraic framework could suffice to produce viable Mozartoid variations.

We think the tree distance proposed should be only applied to short pieces, for instance, consisting of eight to sixteen bars; otherwise, we need to consider whether or not a single tree exists for a longer piece of music. In effect, our definition of distance strongly depends on the strength of heads, and if these heads are changed it affects the distance inadequately. Investigating the relationships between the adequacy of the distance versus the length of music piece should be our immediate future work.

We can imagine many possible algorithms for re-rendering besides the current one as we discussed in Section 4. For example, a rendering algorithm may take into account the original notes from which the relevant time-spans are derived. Another one may employ the technique of case-based reasoning with a database consisting of the melody / time-span tree pairs. On the other hand, rendering can be viewed as the inverse process of the GTTM analysis as shown in Fig. 3. Here let us consider the piece obtained by the following two steps: the GTTM analysis builds a time-span tree from an original piece, and a rendering algorithm synthesizes the resulting piece from a time-span tree. Then, a pair of the GTTM analysis and a rendering algorithm that restores the original piece may be proper. Therefore, we think that a rendering algorithm should always be investigated with GTTM analysis.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers 23500145 and 25330434.

7. REFERENCES

- [1] M. Hamanaka, K. Hirata, S. Tojo, Implementing “A Generative Theory of Tonal Music”, *Journal of New Music Research*, 35(4), 249–277, 2007.
- [2] M. Hamanaka, K. Hirata, S. Tojo, Melody Morphing Method Based on GTTM, *Proceedings of ICMC 2008*, pp.155-158, 2008.
- [3] K. Hirata, S. Tojo, and M. Hamanaka, Melodic Morphing Algorithm in Formalism, In *Proceedings of MCM2011*, also in *LNAI6726*, Springer, pp.338-341, 2011.
- [4] K. Hirata, S. Tojo, and M. Hamanaka, Cognitive Similarity grounded by tree distance from the analysis of K.265/300e, *Proceedings of CMMR 2013*, pp.415-430, 2013.
- [5] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*, The MIT Press, Cambridge, 1983.
- [6] A. Marsden, Generative Structural Representation of Tonal Music, *Journal of New Music Research*, 34(4), 409–428, 2005.
- [7] G. Nierhaus, *Algorithmic Composition*, Springer-Verlag (2009).
- [8] D. Rizo-Valero, *Symbolic Music Comparison with Tree Data Structure*, Ph.D. Thesis, Universitat d’ Alacant, Departamento de Lenguajes y Sistemas Informáticos, 2010.
- [9] C. Roads, *The Computer Music Tutorial*, The MIT Press (1996).
- [10] SWI-Prolog, <http://www.swi-prolog.org/> Accessed on April 1, 2014.
- [11] S. Tojo and K. Hirata, Structural Similarity Based on Time-span Tree, *Proceedings of CMMR 2012*, pp.645-660, 2012.
- [12] Wikipedia, Multidimensional scaling, http://en.wikipedia.org/wiki/Multidimensional_scaling, Accessed on April 1, 2014.