

グルーピング規則適用を改良した GTTM の実装

東洋 武士

北陸先端科学技術大学院大学

t-touyou@jaist.ac.jp

東条 敏

北陸先端科学技術大学院大学

tojo@jaist.ac.jp

平田 圭二

NTT コミュニケーション科学基礎研究所

hirata@bri.ntt.co.jp

佐藤 健

国立情報学研究所

ksatoh@nii.ac.jp

人間の音楽の認知活動を楽譜という情報源から構造的に分析し、音楽認知を客観的に捉えようとする理論の中に Generative Theory of Tonal Music(GTTM) という理論がある。この理論は構成するルール群が箇条書きされているという特徴などから、計算機上に実装し分析を自動化させることが有望視されている理論である。GTTM は4つの理論(グルーピング構造分析, 拍節構造分析, タイムスパン簡約, 延長的簡約) からなる。本稿ではこれまでに作成したグルーピング構造分析の自動化プログラムについて実装するに際しての問題点と、その解決法を示す。

Improvement of Grouping Rule Application in Implementing GTTM

Takeshi Touyou

Japan Advanced Institute of Science and Technology

Keiji Hirata

NTT Communication Science Laboratories

Satoshi Tojo

Japan Advanced Institute of Science and Technology

Ken Satoh

National Institute of Informatics

Generative Theory of Tonal Music (GTTM) is one of the theories that analyze the structure of music from scores and explain the process of human recognition objectively. The theory is expected to be implemented on computers to analyze music automatically, because its rules are declaratively mentioned. GTTM consists of four sub-theories, that is, Grouping Structure, Metrical Structure, Time-span Reduction, and Prolongation Reduction. In this paper, we show the implementation of several rules of Grouping Structure, and discuss their problems.

1 はじめに

人間が音楽を聴くとき、はじめて聞くような旋律でも心地よく感じたり、音が外れている様に聞こえるのはなぜなのだろうか、このような問いに対する研究は古くから研究者によりさまざまな研究が行われてきた。その中で、楽曲を音符列という符号化された情報であるという視点からこれを構造的に分析し、音楽認知を客観的に捉えようという理論がある。Generative Theory of Tonal Music(GTTM) はそのような理論の中のひとつで、さまざまな理由により計算機上での自動化が有望視されている。GTTM による楽曲の分析が自動化されれば、これまでの音楽検索エンジンとは違ったアプローチの楽曲の検索エンジンの作成や、自動伴奏システム、作曲支援などへの応用が期待できる。

しかし、楽譜を計算機上で扱うための音楽知識表現

や、楽曲構造解析に関する研究はこれまでいくつか散見されるものの具体的な成果はあまり報告されていない。GTTM は元々計算機上への実装を目指した理論ではなく、自動化の実現には多くの問題点がある。本稿では、GTTM の構造分析の一つである、グルーピング構造分析に関して計算機上に実装する際に生じる問題点と、その解決法について述べる。

2 GTTM について

2.1 GTTM の概要

GTTM は1983年、Fred Lerdahl と Ray Jackendoff によって提唱された理論である。この理論は調性音楽の経験をもつ聴衆に共通した音楽的直感によって得られる内容を形式的に記述することを目的とした理論であり、理論基盤を Schenker の音楽解析理論と、Chomsky の生成言語文法にもつ。GTTM は以下

に示す 2 つの基本構造分析と 2 つの簡約から構成される。

- グルーピング構造分析
- 拍節構造分析
- タイムスパン簡約
- 延長的簡約

グルーピング構造分析の結果と拍節構造分析の結果を用いてタイムスパン簡約を行い、その次に延長的簡約を行う。これら各々の構造分析や簡約は以下の 2 種類の規則からなる、

- 構成規則 (Well-Formedness Rule): 構造を生成するためのルール
- 選好規則 (Preference Rule): 複数の構造が構成規則を満たす場合、好ましい構造を示すためのルール

これら規則が箇条書きで記述されている点が、GTTM が計算機上での自動化が有望視されている理由の一つである。

2.2 グルーピング構造分析

グルーピング構造分析は、楽曲を音楽的なまとまり感をもつ小さいグループに分けさらにそのグループの階層構造を決定する分析 (図 1) である。グルーピ



図 1: グルーピング構造の例

ング構造分析の選好規則 Grouping Preference Rules (GPR) は 7 つの規則から成る。ここではこれまでに実装した 2 番目と 3 番目の GPR (本稿では GPR2, 3 と書く) の GPR2, 3 について簡単な説明を記す。GPR2, 3 は音符の並びからグループの境界がどこに認識されるかを記述したルール群である。

- GPR 2 (proximity), GPR 3 (change) : 連続した 4 つの音符をそれぞれ n_1, n_2, n_3, n_4 とすると、以下の条件が成り立つとき n_2, n_3 の間がグループの境界と認識される。

- GPR2a: 演奏された音符の間に休符がある。
- GPR2b: オンセット時間の間隔が変化した。
- GPR3a: 連続した音符の間の相対的な音程が変化した。
- GPR3b: 音量が変化した。
- GPR3c: アーティキュレーションパターンが変化した。
- GPR3d: 音長が変化した。

アーティキュレーションとは、楽譜上の音符の長さ、実際に演奏された音の長さの比である。

3 GPR2, 3 実装のためのアプローチ

3.1 本分析プログラムが対象にするデータ形式

本分析プログラムが分析の対象にする楽譜のデータは一般に Web 上より容易に手にいれることができるという理由から、Standard MIDI File (SMF) とした。また、現在入手できる SMF データは MIDI シーケンスソフトなどを用いて作られたデータばかりではなく、MIDI 入力楽器などにより打ち込まれた手弾きのデータも数多く存在するので、手弾きの SMF も入力データとして取り扱えるよう前処理を行う。

3.2 先行研究のアプローチと問題点

井田 [2] による GTTM の自動化に関する先行研究では、グルーピング構造分析の自動化の処理手順は、以下のものであった。

1. 上下 2 半音以内の音に対してボイスリーディングが成立しているとし、そのような音すべてを時間順にリンクする。
2. 第一声部に限ってボイスリーディングの成立不成立にかかわらずリンクを結ぶ。
3. そのリンクをたどりながら 4 つの音符を取り出し、GPR2, 3 を適用する。

このアプローチでは以下の点が問題点となる。

- 手弾きの SMF について。
手弾きの SMF を扱う際には、奏者が同時に弾いているつもりでも各音符のオンセット時間と音の大きさが異なるという問題が生じる。しかし

先行研究の方式は、オンセット時間のずれと音の大きさのずれの対処を行っていない。

- ポリフォニーからなる楽曲を入力とした際の声部の切り分け処理が不十分。

GTTM はすべての楽曲を本質的にホモフォニーとして扱う。そのためポリフォニーからなる楽曲は、各声部がホモフォニーになるまで分解してから分析しなければならない。しかし SMF では楽曲の情報は音がどの時間で始まりどのくらいの強さで、どれくらいの長さであるといったいわばピアノロールのような情報しかなく、声部を分離するための情報が欠落している。例えば小節線や、音符がト音記号の五線譜上にあるのかへ音記号の五線譜上にあるのかなどである。先行研究では、ボイスリーディング、第一声部進行という手法を用いたが第一声部以外の声部の切り分けは、まだ不十分であった。

- GPR3c が未実装。
SMF はスラー、スタッカートなどのアーティキュレーションパターンの情報を記述できない。このため、GPR3c が実装できず、グルーピングのための情報が不足する。

3.3 本研究でのアプローチ

GPR2, 3 の適用は基本的に、以下の順序で行う:

- (1) 声部を切り分ける,
- (2) 4 つの音符を探し出す,
- (3) GPR2, 3 を適用する。

ここで、(1) の声部の切り分けに関して、3.2 節で述べた方法の他に、まずすべてのリンクを結んでしまい声部進行の候補に成り得ないリンクを消していき、残ったリンクに対して GPR2, 3 を適用するという手法が考えられる。本分析プログラムではこの手法を実装した。

4 GPR2, GPR3 の定式化

本章では、GPR2, 3 の定式化を行いアルゴリズムとして本質的な部分と、ヒューリスティクスとの切り分けを行う。この定式化により様々なグルーピング手法が採用しているヒューリスティクスどうしの比較が可能となる。

4.1 述語 κ_S , *change*, *proximity* の導入

同じ声部に属すると思われる音符同士を時間順に結びリンクのことを本稿では隣接音接続と呼ぶ。楽譜 S において、音符 $n_1 \sim n_4$ が隣接音接続している場合に真を返す述語 $\kappa_S(n_1, n_2, n_3, n_4)$ を導入する。GTTM では κ_S が明示的に記述されていなかった。

$change(n_1, n_2, n_3, n_4)$ は GPR2, 3 に基づいてグループ境界を判定する述語で、 n_2, n_3 の間に GPR2, 3 が成立すると真を返す。 $proximity(n_1, n_2, n_3, n_4)$ はその逆で、GPR2, 3 に基づいて n_2, n_3 がグループ境界でないことを判定する述語である。つまりこれら 2 つの述語の間には、

$$change(n_1, n_2, n_3, n_4) = \neg proximity(n_1, n_2, n_3, n_4)$$

という関係が成立している。

4.2 定式化

以上の述語を用いることで、GPR2, 3 を以下のように定式化できる。

$$\exists n_2, n_3 \text{ if } \forall n_1 n_4 \quad \kappa_S(n_1, n_2, n_3, n_4) \rightarrow change(n_1, n_2, n_3, n_4)$$

then n_2, n_3 の間をグループ境界とする。... (a)

\leftrightarrow

$$\exists n_2, n_3 \text{ if } \exists n_1 n_4 \quad \kappa_S(n_1, n_2, n_3, n_4) \cap proximity(n_1, n_2, n_3, n_4)$$

then n_2, n_3 の間はグループ境界としない。... (b)

(a) は最初にリンクした隣接音接続を切断していく手法であり、(b) は逆に隣接音接続を残していく手法である。これら 2 つの手法は互いに等価である。従って、(a) と (b) どちらの手法を用いても求める解は同じになる。先行研究の手法は (b) であり、本分析プログラムでは (a) の手法を実装している。

5 実装と実験

5.1 手弾き SMF のための前処理

現在実装中の分析プログラムでは、手弾き SMF によるオンセット時間のずれに対応するために、ある時刻範囲を決めてそれを同一時刻と見なす前処理を加えた (図 2)。

5.2 κ_S の実装

本分析プログラムでは、 κ_S の実装について、2 つの音符間の隣接音接続を 3 つ連結するという手法を

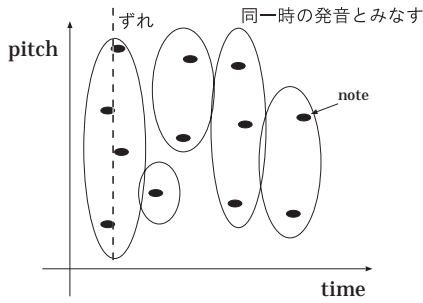


図 2: 手弾き SMF のための前処理

とった。例えば、...C4, D4, E4, F4... というメロディを考える、まず C4 と D4 について下に述べるヒューリスティクスを適用して隣接音接続判定をし、同様に D4 と E4, E4 と F4 の判定をする。それらがすべてつながっていると判定されれば $\kappa_S(C4, D4, E4, F4)$ を真とする。本分析システムで隣接音接続判定として導入したヒューリスティクスは以下の通りである。

- 12 半音以内。
着目している音から 12 半音以内の後続音を隣接音接続する。
- 4 拍以内。
着目している音から 4 拍以内の後続音を隣接音接続する。
- 後続音を n , n の後続音を m として, n との音高差を考慮した m との隣接音接続判定。
これは、着目している音と n との音高差により、着目している音と m を隣接音接続するかどうかを判定するものである。例えば、同じ高さの音が 3 つ続いたとしたら着目している音から n , n から m という隣接音接続は考えられるが、着目している音から直接 m という隣接音接続は成立しないと仮定する (図 3 a), また C, C#, D, D# のような時にも C から D や、C から D# のような隣接音接続は成立しないと仮定する (図 3 b)。

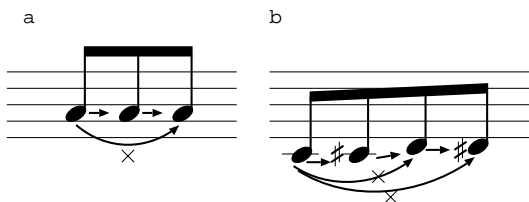


図 3: 音高に関する隣接音接続判定

- 着目している音符のオンセット時間を t , 後続音のオンセット時間を s として, $(s-t) \times 2 + t$ より後のオンセット時間をもつ音は着目している音と隣接音接続しない。

前頁のヒューリスティクスと同様にボイスリーディングとして無意味な隣接音接続のリンクを取り除くためのものであり、一定比率以上オンセット時間に差がある隣接音とはリンクを張らない (図 4)。本分析プログラムが採用している比率 2 という値に特別な意味はない。

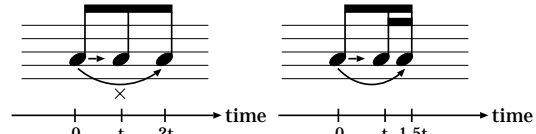


図 4: 時間による隣接音接続判定

- 楔形の選択範囲の適用。
これは、音高差/時間差の絶対値が大きい音符同士の隣接音接続を切ることを意図したヒューリスティクスである。12 半音, 4 拍以内に属している音同士でも急激な音高差の変化がある場合は同じ声部には属しないと判定する (図 5)。

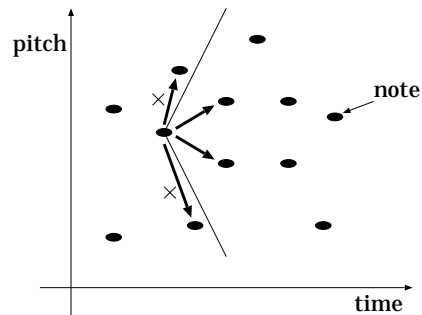


図 5: 楔形の選択範囲の適用

5.3 一般的な楽曲の分析

本手法を, W. A. Mozart 作曲, 交響曲 第 40 番ト単調 K. 550 第一楽章 (はじめの 2 小節, 図 6, RWC-MDB-C-2001-No.2 [3]), 及び J. S. Bach 作曲 Invention No.1 (はじめの 2 小節, 図 7) に適用した結果を示す。図 6, 図 7 中, 音符間をリンクする線分が計算された隣接音接続を表す。

図6は、本分析プログラムの結果 (a) と文献 [1] の結果 (b) の違いを示している。(a) 中央にある四分休符で GPR 2a, 2b が適用されていないのは、 κ_S のオンセット時間差のヒューリスティクスにより隣接音接続されないためである。9と10の音符についても GPR 3a が適用されていないのは同様の理由である。また、3と4, 5と6, 13と14, 16と17に GPR 2a の適用がみられるのは、当該楽曲の SMF を作る際に使用したシーケンスソフトがオフセット時間と隣接音のオンセット時間の間自動的に休符を挿入するためである。18と19のリンクに GPR 3c が適用されていないのは、SMF がアーティキュレーションパターンの情報を持っていないためである。

図7では19と22, 22と30の隣接音接続がなされていない。これは κ_S の音高差及びオンセット時間差のヒューリスティクスにより隣接音接続されないためである。また、上声部の中央付近、17, 20, 21の音の間に三角形のリンクが現れている。隣接音接続としては17→20→21が正しいように思えるが、この正しいリンクをどのように導き出すかは今後の課題である。

6 まとめと今後の課題

本分析プログラムでは、井田 [2] による GTTM の実装に関する研究を元に、より精度の高い自動解析プログラムの作成を目指した。GTTM の自動化に関する研究には MAURO BOTELHO [5] によるグルーピング構造分析に関する研究や、Peter Halasz [6] によるタイムスパン簡約に関する研究、上符 [7] による延長的簡約に関する研究などがある。また、GTTM の応用に関する研究としては、平田、青柳 [4] による研究がある。ここでは、タイムスパン簡約の結果より得た簡約木を用いて、演繹オブジェクト指向データベース (Deductive Object-Oriented Database, DOOD) を用いた音楽システムを構築している。

本稿で述べたことを以下にまとめる。

- ポリフォニーに対応できるようなグルーピング手法を提案した。
- 手弾き SMF のオンセット時間のバラつきに対応するための前処理を加えた。
オンセット時間のバラつきに対してはある程度対応できるような処理を加えたが、音量のバラつきに対しては未処理である。この部分をどう

実装するかは今後の課題である。

- GPR2, 3 の定式化を行った。
- κ_S を実装した。
より適切な隣接音接続判定法が望まれる。例えば、...-C#4 - Bb3 - C4 - ... のようなメロディの隣接音接続には曖昧さがある。この問題に対しては音程/時間差比の導入を考えている。音程/時間差比の値設定により (a), (b) どちらかの接続を選択することが可能となる (図8)。このヒューリスティクスを適切に設定すれば示した Invention No.1 (図7) における三角形のリンクが解決できるであろう。音程/時間差比の実装は今後の課題である。

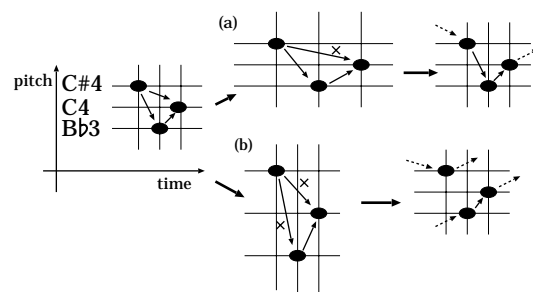


図8: 音程/時間差比の導入

さらに今後は、上位のグルーピング構造を決定する規則である GPR4, 5, 6 の実装、拍節構造分析の実装、タイムスパン簡約の実装を行う予定である。

参考文献

- [1] Fred Lerdahl, Ray Jackendoff, A Generative Theory of Tonal Music, The MIT Press, 1983.
- [2] 井田健太郎, 延長的簡約を拡張した GTTM による楽曲の構造解析, 北陸先端科学技術大学院大学修士論文, 2002.
- [3] 後藤真孝, 橋口博樹, 西村拓一, 岡隆一, RWC 研究用音楽データベース:クラシック音楽データベースとジャズ音楽データベース, 情報処理学会 音楽情報処理研究会 研究報告, No. 44, pp. 25- 32, 2002.

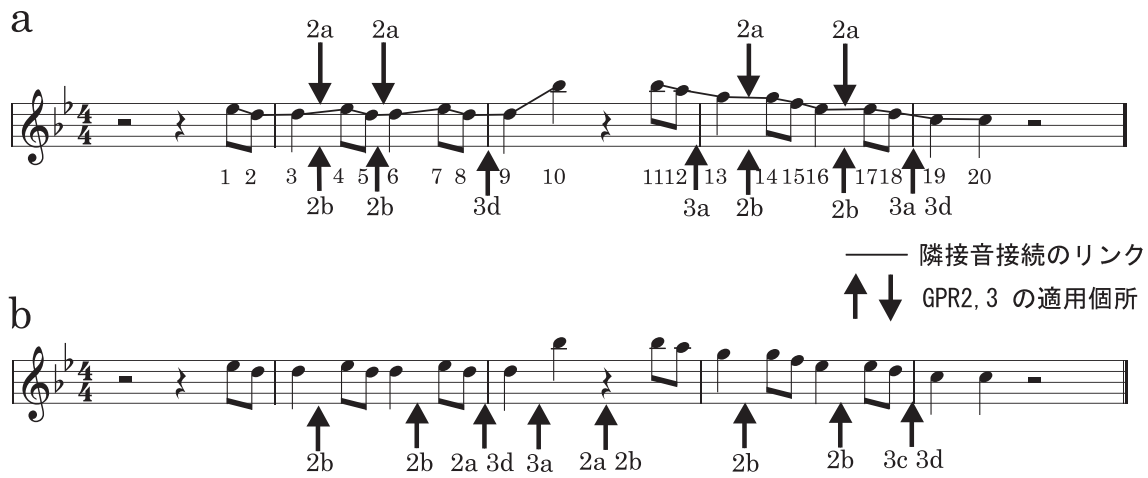


図 6: “ 交響曲 第 40 番ト単調 K. 550 第一楽章 ”(a 本システムの分析結果, b 文献 [1], 47 頁)

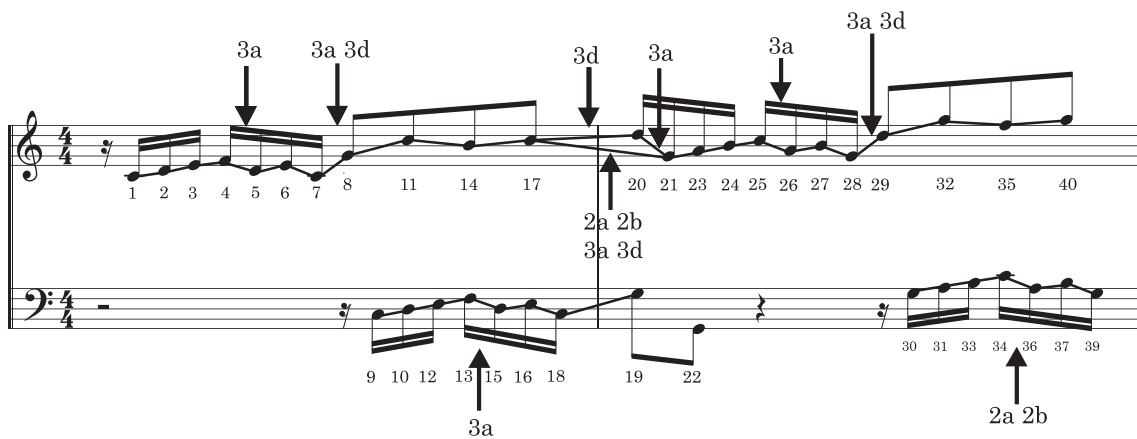


図 7: “ Invention No.1 ”分析結果

[4] 平田圭二, 青柳龍也, パービーブン: 音符レベルでユーザの意図を把握して編曲を行う事例ベースシステム, 情報処理学会 音楽情報処理研究会 研究報告, No. 37, pp. 17- 23 ,2000.

[7] 上符裕一, コーパスを用いた音楽構造解析, 大阪大学 基礎工学部 修士論文, 1998.

[5] MAURO BOTELHO, TONAL GROUPING:AN ADDENDUM TO LERDAHL AND JACKENDOFF' S “ A GENERATIVE THEORY OF TONAL MUSIC ”, International Conference of Music Perception and Cognition(ICMPC), No.3, pp.265-266, 1994.

[6] Peter Halasz, COMPUTER SIMULATION OF TIME-SPAN REDUCTION, International Conference of Music Perception and Cognition(ICMPC), pp267-268, 1994.