JAIST Repository
https://dspace.jaist.ac.jp/

Title	グルーピング規則適用を拡張したGTTMの実装
Author(s)	東洋，武士
Citation	
Issue Date	2003-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1699
Rights	
Description	Supervisor:東条 敏，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

グルーピング規則適用を拡張したGTTMの実装

指導教官　東条 敏教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

東洋 武士

2003年2月

Copyright © 2003 by Takeshi Touyou

要 旨

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, 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 は理論基盤を Schenker の音楽解析理論と, Chomsky の生成文法理論に持ち. 上位下位関係という階層的な構造に解析することでツリー構造を生成し楽曲を解析する. GTTM はそのツリー構造を作るための 2 つの構造分析と簡約のための理論からなる. それらはそれぞれ, グルーピング構造分析, 拍節構造分析, タイムスパン簡約, 延長的簡約と呼ばれる. これらすべてを計算機上で自動化し 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. 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システムを実装するに際しての問題点と, その解決法を示す.

目 次

1 はじめに **1**

1.1 研究の背景と目的 1

1.2 本論文の構成 2

2 A Generative Theory of Tonal Music **3**

2.1 GTTM の概要 3

2.2 グルーピング構造分析 4

2.2.1 グルーピング構成規則 4

2.2.2 グルーピング選好規則 5

2.3 拍節構造分析 6

2.3.1 拍節構成規則 6

2.3.2 拍節選好規則 7

2.4 タイムスパン簡約 8

2.4.1 タイムスパン簡約構成規則 9

2.4.2 タイムスパン簡約選好規則 9

2.5 GTTM の自動化についての関連研究 11

2.5.1 他の音楽解釈理論との比較 11

2.5.2 GTTM に関する研究 11

2.5.3 音楽の類似度に関する研究 11

2.5.4 GTTM の応用に関する研究 17

2.5.5 先行研究 17

3 グルーピング構造分析の自動化 **18**

3.1 本分析プログラムが対象にするデータ形式 18

3.2 GTTM のグルーピング構造分析が持つ問題点 21
3.2.1 規則の優先度についての問題 22
3.2.2 用語定義の曖昧性 22
3.3 先行研究のアプローチと問題点 24
3.4 グルーピング構造分析のアプローチ 26
3.4.1 GPR2, 3 実装のためのアプローチ 26
3.4.2 述語 κ_S , *change* の導入 27
3.4.3 GPR2,3 の定式化 27
3.4.4 GPR4 実装のためのアプローチ 27
3.4.5 GPR6 実装のためのアプローチ 28
3.4.6 グルーピング構造木作成 29

4 実装と実験 31

4.1 システムの概要図 31
4.2 前処理 31
4.2.1 xsmf 31
4.2.2 note on, note off の対応付けを行った note 情報を作り出すプログラム 32
4.2.3 手弾き SMF のための前処理 33
4.3 κ_S の実装 33
4.4 GPR4 のパラメータ 36
4.5 GPR5 のパラメータ 36
4.6 GPR6 のパラメータ 36
4.7 実験 37
4.7.1 声部切り分け処理と GPR2, 3 の結果 37
4.7.2 上位のグルーピングのための規則実装の実験 39

5 まとめと今後の課題 43

5.1 本研究での成果 43
5.2 今後の課題 44

第 1 章

はじめに

1.1 研究の背景と目的

近年, 計算機の性能が向上したこともあり様々な分野での計算機の利用が増えてきている. その中で, 計算機上での音楽に関する研究も増えてきている. 自動伴奏, 自動採譜, 楽曲検索, 自動作曲, 作曲支援などがその例である.

人間ははじめて聞くような音楽を心地好い音楽であるとか, ここは音がはずれているななどと認識することができる. そのような人間の音楽認知活動を客観的に分析しようという研究は古くから多くの研究者によりさまざまな研究が行われてきているが, そもそも音楽は人それぞれ感じ方, 捉え方が異なりそのような主観的な音楽解釈を客観的な視点から構造を分析するというのは大変困難な問題である. そしてそのような問題から, 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に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.

そのような背景のなか, 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音楽理論がある. Generative Theory of Tonal Music(GTTM) はそのような理論の中のひとつである. GTTM はグルーピング構造分析, 拍節構造分析, タイムスパン簡約, 延長的簡約という 2 つの構造分析と 2 つの簡約のための理論から構成され, それら各理論を構成する規則が箇条書きされているなどの理由により計算機上での自動化が有望視されている. GTTM はある音楽語法の経験を持つ聴衆が共通して持つ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にを目標としている. ある音楽ジャンルとは調性音楽のことであり, 調性音楽とは一

般にクラシック音楽と呼ばれる音楽の中に多く含まれ, クラシック音楽の中で, ある楽曲に対して, 一つの音 (主音と呼ばれる) が旋律, 和声の中心として働き, 他の音が従属的に関わっているような楽曲を調性音楽と呼ぶ.

GTTM の理論基盤は Schenker の音楽解析理論と, Chomsky の生成文法理論にある. その双方に共通するのは, 上位下位関係という階層的な構造に解析することであり, GTTM では Chomsky のツリー解析手法を音楽に適用している. GTTM による楽曲の分析が自動化され楽曲をツリー構造へと解析することができれば, 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, 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. 例えば, 規則定義の曖昧性という問題がある, これは自然言語で述べられた曖昧な表現をどのように数値化したり定式化するかという問題である. また, 規則の優先度の問題なども挙げられる. これは箇条書きされた規則同士をどのような順序で適用すればいいのかという問題点である.

本稿ではそれら問題点についての詳細と, その解決法について述べ, 作成したグルーピング構造分析システムについて実験結果と考察を行う.

1.2 本論文の構成

本稿では, 2 章にて GTTM についてと GTTM に関する研究について, を記した後, 3 章で 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について述べる. 4 章では考案したシステムの計算機上への実装と, 実際の楽曲を入力とした実験とその結果についての考察を述べ, 5 章でまとめと今後の課題について述べる.

第 2 章

A Generative Theory of Tonal Music

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記述することを目標とした理論であり, 理論基盤を Schenker の音楽解析理論と, Chomsky の生成文法理論にもつ. GTTM は以下に示す 2 つの基本構造分析と 2 つの簡約のための理論から構成される.

- 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 節から, 解析木を作るまでの構造分析 2 つ (グルーピング構造分析, 拍節構造分析) と, 簡約ための理論であるタイムスパン簡約について簡潔に述べる.

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 2.1) である.

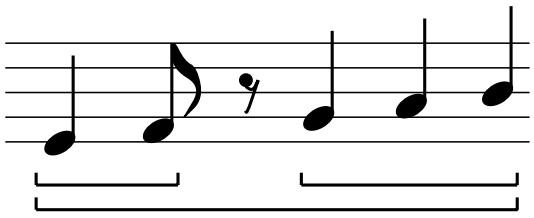

図 2.1 グルーピング構造の例

グルーピング構造分析は構成規則 (Grouping Well Formedness Rules, 以下 GWFR) , 選好規則 (Grouping Preference Rules, 以下 GPR) から成る.

2.2.1 グルーピング構成規則

GWFR は 5 つのルールから成り, 以下のようである.

- GWFR1: 構成要素が連続している場合のみグルーピングを形成することができる.
- GWFR2: 1 つの曲は 1 つのグループである.
- GWFR3: グループはより小さなグループ (サブグループ) を内部に含んでもよい.
- GWFR4: グループはサブグループの一部だけを含むことは許されない. サブグループ全体を内部に含まなければならない.
- GWFR5: グループがサブグループを含むなら, グループ構造が交差しないサブグループ群によって内部を埋めつくされねばならない.

2.2 は GWFR を適用したグルーピング構造分析の例である．a は正しい例であり, b は 1, 2, 3 の各個所が誤っている．1 はサブグループ同士が密接せずに離れてしまっているため

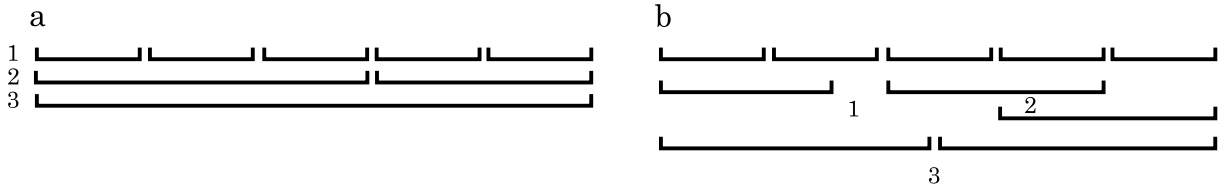

図 2.2 GWFR 適当例

に誤りである．2 はサブグループ同士が交差してしまっているために誤りである．3 はサブグループの境界とその上のレベルにおけるグループの境界が一致していないために誤りである．

本稿では, 2.2, a の 1 のように楽譜を細かくグループ分けした時の集合を “下位のグループ”, 3 のように大きくグループしたときの集合を “上位のグループ” と呼ぶこととする．

2.2.2 グルーピング選好規則

GPR は 7 つのルールからなり, 以下のようである．

- GPR 1 (alternative form): 非常に小さいグループへの解析は避ける．特に単音をグループとすることは避ける．
- 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．

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

- GPR4 (intensification): GPR2, 3 で示される効果が比較的明白なところは大きなレベルにおいてもグループの境界がそこで位置づけられる可能性が高い.
- GPR5 (symmetry): グループの分割が長さの等しい 2 つの部分からなるようにグルーピングすることを優先する.
- GPR6 (parallerism): グループ間で並行した部分を形成することができる 2 つもしくはそれ以上のグルーピングは, 並行性のあるグルーピングを行う.
- GPR7 (Time-Span and Prolongational Stability): タイムスパン簡約や延長的簡約がより安定するグルーピング構造を優先する.

2.3 拍節構造分析

拍節構造とは, 指揮者が指揮棒を振る時のようなアクセントの周期的反復の構造である. GTTM における拍節構造分析はそのような拍節を強拍と弱拍をもつ階層構造に分析することである.

拍節構造分析は以下の拍節構成規則 (Metrical Well Formedness Rules, 以下 MWFR), 拍節選好規則 (Metrical Preference Rules, 以下 MPR) からなる.

2.3.1 拍節構成規則

MWFR は以下の 4 つから成る.

- MWFR1: すべてのアタックポイントは，曲中の各部分における最小の拍節レベルの拍でなければならない.
  アタックポイントとは, 音がなり始める点である. つまり, すべての音符に拍節構造の拍がなければならないということである.

- MWFR2: あるレベルのすべての拍は，より小さなレベルでの 1 つの拍でもある.
- MWFR3: 各々の拍節的レベルで，強拍は 2 または 3 拍の間隔をもつ.
- MWFR4: タクトゥスや大きな拍節レベルは，同等な間隔をもつ拍によって構成される.
  タクトゥスとは, 拍を意味するラテン語で, 楽曲中の最小の音符の長さのことである.

図 2.3 は MWFR を適用して W. A. Morzart 作曲, ピアノソナタ イ長調 K. 331 (はじめの 4 小節, RWC-MDB-C-2001-No. 26[4]) を解析した例である.

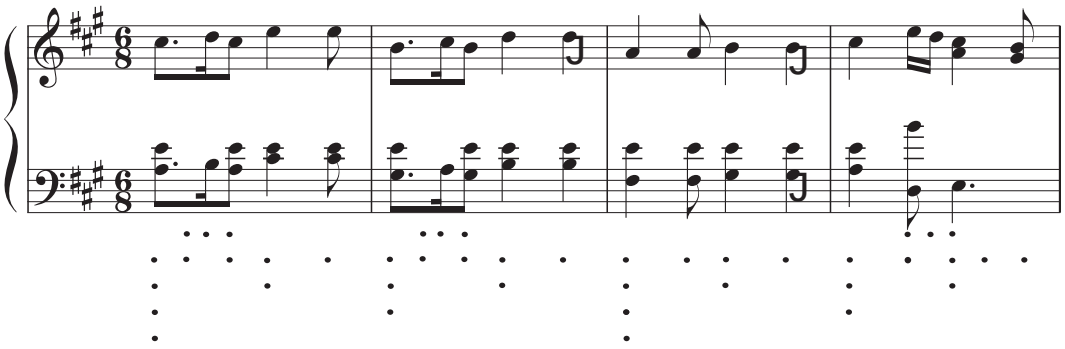

図 2.3 MWFR 適用例

2.3.2 拍節選好規則

MPR は以下の 10 のルールから成る.

- MPR1(parallelism): 複数のグループ, またはグループの各部を並行的と解釈できる場合, 並行的な拍節構造を優先する.
- MPR2(strong beat early): 最も強い拍がグループ内で比較的早く現れる拍節の構造を優先する.
- MPR3(event): 拍点に音符がある (さらに強拍となる) 拍節構造を優先する.
- MPR4(stress): 強く演奏された拍が強拍である拍節構造を優先する.

- MPR5(length): 以下の a〜f のような,「より長い」という条件を満たす拍を強拍とする拍節構造を優先する.
  a. 相対的に長い音
  b. 相対的に長く続く一定の音量
  c. 相対的に長いスラー
  d. 相対的に長い同じアーティキュレーションパターンの繰り返し
  e. タイムスパン簡約による相対的に長く続く 1 つの音高 (同一音高音の連続)
  f. タイムスパン簡約による相対的に長く続く 1 つの和声 (同一和音の連続)

- MPR6(bass): バス音が拍節的に安定した拍節構造を優先する.

- MPR7(cadence): カデンツでは拍節的に安定した構造を優先する. つまり他の場合よりもカデンツ内での局所的な選好ルールの違反は避けなければならない.
  カデンツとは, 機能和声でいう終止形に向かうグループのことである.

- MPR8(suspension): 掛留音はその解決よりも強拍である拍節構造を優先する.
  掛留音とは協和している和音の発声時間をずらすことにより, 意図的に不協和状態を作り出す音のことである.

- MPR9(time-span interaction): タイムスパン簡約における競合が最小になるような拍節構造を優先する.

- MPR10(binary regularity): 各レベルにおいて, 強拍が 1 つおきにくる拍節構造を優先する.

2.4 タイムスパン簡約

タイムスパン簡約は, 楽曲を階層的な時間間隔 (タイムスパン) に分割し, 各タイムスパンを構造的に重要な音とそうでない音に簡約化することである. このような各音の構造関係を階層的に表示するのが, タイムスパン簡約木である.

タイムスパン簡約はタイムスパン簡約構成規則 (Time-Span Reduction Well-Formedness Rules, 以下 TSRWFR) と, タイムスパン簡約選好規則 (Time-Span Reduction Preference Rules, 以下 TSRPR) からなる.

2.4.1 タイムスパン簡約構成規則

- TSRWFR1: 全てのタイムスパンTは，Tのヘッドとなるイベントe(もしくはイベント列e_1e_2)をもっている.

- TSRWFR2: Tが他のタイムスパンを含んでいない(つまり最小レベルのタイムスパン)ならば，eはTで起こるイベントである.
  これはつまりTが最小レベルのタイムスパンでTが含むイベントがeの場合, eはTのヘッドだと言うことを示している.

- TSRWFR3: Tが$T_1, ..., T_n$というタイムスパンを含んでいて, $e_1, ..., e_n$を$T_1, ..., T_n$のヘッドとすると，以下のようである.
  a. (Ordinary Reduction): Tのヘッドはイベント$e_1, ..., e_n$の中の１つである.
  b. (Fusion): もし$e_1, ..., e_n$がグループ境界によって分けられないなら，Tのヘッドは2つ以上の$e_1, ..., e_n$を組み合わせたものからできている可能性がある.
  c. (Transformation): もし, $e_1, ..., e_n$がグループ境界によって分けられないなら, Tのヘッドは$e_1, ..., e_n$の中から選んだ相互に調和した音の組み合せである可能性がある.
  d. (Cadential Retention): Tのヘッドはカデンツの場合, その最後のe_nと最後から2番目の2つからなるタイムスパンがヘッドになるかもしれない.
  これはカデンツの場合特殊なタイムスパン木を作ることを意味している.

- TSRWFR4: もし2つのカデンツが直接タイムスパンTのヘッドeに従属するなら，最後のカデンツは直接ヘッドeに従属する．そして最後から2番目のカデンツは最後のカデンツに従属する.

2.4.2 タイムスパン簡約選好規則

- TSRPR1: より強い拍の部分の優先.

- TSRPR2: 協和部や局所的な主音に関連がある部分の優先.

- TSRPR3: 旋律の高い音，より低いバス音の優先.

- TSRPR4: 並行的な部分は，並行したヘッドとなる.
- TSRPR5: より安定した拍節構造部をヘッドとして優先.
- TSRPR6: 可能な侯補がいくつもある場合，延長的簡約において安定度の高い結果を与えるものをヘッドとして優先.
- TSRPR7: カデンツ進行部の優先.
- TSRPR8: 開始部の優先.
- TSRPR9: 開始部よりも終結部 (カデンツ) の優先.

図 2.4 はタイムスパン簡約の各規則を W. A. Morzart 作曲, ピアノソナタ イ長調 K. 331 (はじめの 8 小節, RWC-MDB-C-2001-No. 26[4]) に適用した解析例である.

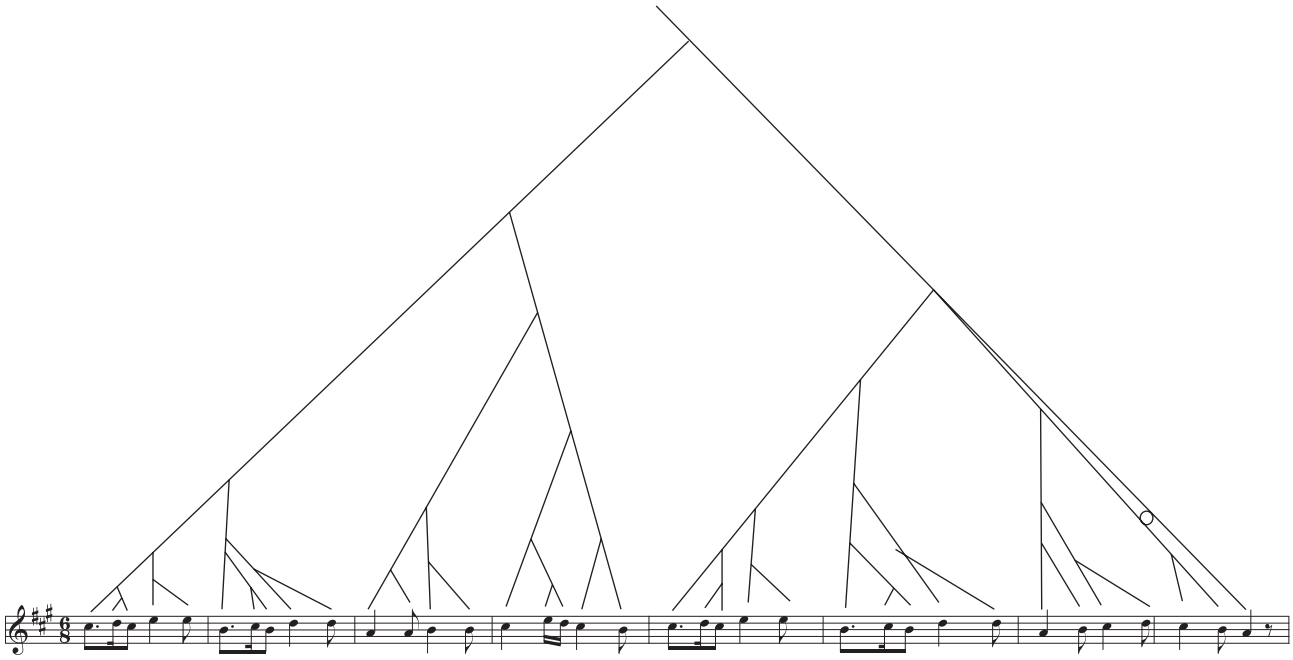

図 2.4 タイムスパン簡約例

2.5 GTTMの自動化についての関連研究

2.5.1 他の音楽解釈理論との比較

片寄, 竹内 [9] は, GTTM を含めた音楽解釈理論を認知的な視点から比較し, 計算機での解析に際してのメリットと解決すべき問題について議論している. そこでは代表的な音楽理論を表 2.1, 2.2 のようにまとめている. ここでは, GTTM について “フレージングや, 拍節表現をルール的に表現するシステムの条件節を導く手法として期待が持てる” とするものの, 各規則の優先度の問題点や, 音楽認知構造を生成文法的に記述しているが, 言語認知と音楽認知の構造的な差異の点に関しても以後検討する必要がある, としている.

2.5.2 GTTMに関する研究

MAURO BOTELHO[11] は，グルーピング構造分析に関する研究を行っている. ここではグルーピング構造をリズム的なグルーピングと, 調性的なグルーピングに分けている, 調性的なグルーピングが GTTM に取り込まれるなら, GPR7 を弱めて GPR6 を強めるであろうと述べている.
Peter Halasz[12] では，タイムスパン簡約をコンピュータ上でシミュレーションするという研究をおこなっている. Peter Halasz はこの際，優先規則の TSRPR4,5,6 を省いている．これらはコンピュータの限られた能力のために省いたが，経験的にこれらを省いても分析過程には影響は無いと述べている. また，シミュレーションの結果からタイムスパン簡約は以下の 2 つの段階に分けることができると述べその内容を紹介している.
上符裕一 [15] による研究では，タイムスパン簡約までの結果が出ていると仮定し，延長的簡約について評価実検，考察をおこなっている.

2.5.3 音楽の類似度に関する研究

楽曲を構造解析する際に, 楽曲の中に含まれる似たような構造 (メロディや拍節などについて) を見つけ出すのは, その楽曲の特徴的な構造を抜き出すことができるという理由から非常に意味のある分析であると考えられる. GTTM でも GPR6 に並行なメロディ列

表 2.1 各理論の音楽的な視点からの比較 (文献 [9])

理論	目的	手法, 特徴	分析の内容と課題
Meyer	リズム構造の認知的分析	詩脚法. 拍節アクセントによる分析	アクセントを拍節的な強拍部とすると, 認知構造の分析でなく楽譜の分析になってしまう.
Narmour, 村尾	認知構造の数量化	「暗意-実現のプロセス」の分析による非メトリカルアクセントの数量化	構造主義分析では切り取られるイディオストラクチャを反映している点はより認知的. 数量的公式化の変数に問題がある. 分析対象を旋律だけでなくカデンツやバス音, 対位旋律などを含めた解析が必要である.
Lerdahl, Jackendoff	音楽文法の生成的記述	グループ, 拍節構造, タイムスパン簡約, 延長的簡約.	木構造シェンカーのウルザッツ概念に基づく構造主義的分析. ２種類の聴取傾向をルール化. 厳格な階層グループに分析. 簡約内部緊張- 弛緩の階層構造を生成文法的に記述. 認知的には分析の一義性が問題とされる. グループ化と緊張- 弛緩の関連が未整理.
保科	演奏解釈のための楽曲分析	グループ, フレーズの分析. 重心, 頂点の明示化.	厳格な階層グループ構造に分析. グループ内部の最強調部を重心 (頂点) として明示. 複合グループをフレーズとして分析. グループ構造と演奏変数との相関性を指唆. 分析ルールが経験的.
竹内	演奏家のための演奏解釈	重心や頂点に対応した演奏変数の対応. 重心を Jackendoff 理論より分析.	アナクルーズとデジナンスに対応した, ディナーミクとアゴーギクの適用. 重心や頂点は Jackendoff の延長的簡約の２重弛緩構造 (2 重の左枝) より求めるが, 階層的な拍節構造に一致しない場合は拍節的に強拍 (表拍) 部位を採用する.

表 2.2 各理論の応用性の視点からの比較 (文献 [9])

理論	音楽表現, 構造解析に対する視野	ストラクチャに対する考え方	自動演奏システムへの応用メリット	問題点
Narmour (Meyer)	認知視点からみた音楽構造にかかわる可能なかぎりの音楽的意図の解析.	イデオストラクチャ指向暗意-実現に基づいたたくさんの関係可能性を見る構造はツリー構造にはならない.	詩脚レベルに対応する表現ルールが分かれば, イデオストラクチャ表現ルールの一般化ができる.	複雑. 離れた暗意-実現に関して演奏表現の関係がつかみにくい. メロディしか解析法が示されていない.
村尾	構成アクセントと認知演奏アクセントの分離. 計算式に基づいた Narmour 詩脚法の簡単化	イデオストラクチャ指向, 構成アクセントの大きいところがクロージャ (グループ) スタート.	構成アクセントレベルに対応する表現ルールが分かれば, イデオストラクチャ表現ルールの一般化ができる.	構成アクセント計算式の正当性. 代償を行なう部分かどうかの決定. メロディしか解析方法が示されていない.
Lerdahl, Jackendoff	一般的な音楽素養を持った聴取者の言語理解に相当する構造化	タイムスパン簡約: 木構造 (スタイルストラクチャ), 延長的簡約: イデオストラクチャ	フレージング, 拍節表現をルール的に表現するシステムの条件節を導く手法としての期待が持てる.	優先規則の扱いが定式化されていない. (一意性, 最善性の問題)
保科	演奏者に演奏表現をより分かりやすく指導することを目指す.	階層はグループとフレーズ. グループ (フレーズ) の中のエネルギーポイント (重心, 頂点) は一つ. (重心や頂点の分析は音楽経験的)	構造解釈が出来てしまった状態から演奏表現を生成する筋道が分かりやすい.	エネルギーポイント (重心) の決定の仕方がヒューリスティックで人間の主観処理を前提とする.

をグルーピングする規則が, またMPR1に拍節的に並行性のある部分を見つけるという規則があり, これら規則の実装のためには, 楽曲中の似たような構造を見つける処理が必要となる.

David Cope[3]は, Experiments in Musical Intelligence (EMI) というシステムの中で, 作曲家に特有の “signature” という構造を見つけ, ある音楽文法に沿って接続する作曲システムを構築している.

Tim Crawford[14]らは文字列検索の手法を音楽に適用し楽曲中から意図する音符列を見つけ出すアルゴリズムを紹介している. そこでは楽曲中の音符列検索をいくつかのパターンに分け, 各パターンに適したアルゴリズムを紹介している. パターンとはまず大きく2つ, 楽曲の声部が切り分けられているデータから意図する音符列を見つけ出すアルゴリズム (Exact- Matching) と, 声部が切り分けられていないデータから意図する音符列を見つけ出すアルゴリズム (Inexact- Matching) に分けさらに, その中でまた, 型分けを行いExact- Match, Inexact-match 合わせて12のパターンに対する解決法を提案している.

Lloyd A. Smith[10]らは音符列の比較にDP Matchingの手法を使って類似度を計算している. DP Matchingはパターン認識のアルゴリズムの一つであるが音符同士のコストを正しく設定することで, 計算された総コストでの比較を意味のあるものにできると主張している.

DP Matching の音符列への応用について

DP MatchingはDP(Dynamic Programing: 動的計画法) を用いたパターン認識アルゴリズムで, 画像処理や音声認識などに用いられている. DPは最終的な解を得るために小さな問題のどれを解くべきかわからない時, 単純にそれらすべてを解いてその答えを記憶しておき, それらを使って元の大きな問題を解くという手法である. 具体的な問題として最適経路問題を考えてみる. 図2.5のようなノードの集合があり $\xi(a,b)$ をノードaからbまで の1ステップのコスト, $\phi(a,b)$ をaからbまでの必要なステップ数を経ていった場合の最低コストとし, $\phi(i,l)$ をiからlまでの最適経路として. iからjまでの最適経路を求めようとするとき, 下式

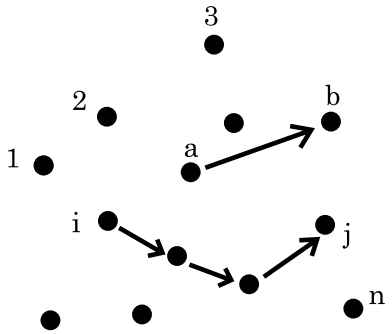

図 2.5 最適経路問題

$$\phi(i,j) = \min_l \left[\phi(i,l) + \xi(l,j)\right]$$

を満たすように最適経路を求める, これはつまりノード i からノード j までの最適系列の任意の部分的な連続した移動の系列はやはり最適でなければいけないことを暗に意味している. これを音符列の比較に応用するのに Lloyd A. Smith[10] では, 以下のような手法を用いている.

例えば図 2.6 のような a, b という 2 つの音符列を比較するとする,

図 2.6 の 2 つの音符列の下に表記した “p - t” という数字は, p: 各音符に SMF の表記に

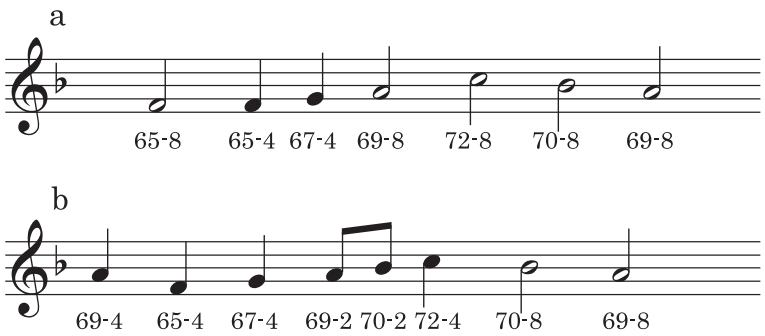

図 2.6 音符列の比較

従った音符の音高 (3.1 節参照、以下音高), t: 16 分音符を単位時間として 16 分音符いくつ分かであらわした各音符の音長 (以下音長), である. この 2 つの値を用いて各音符間のコスト付けを行う. コスト c は, 音符列 n, m の i 番目の音符を n_i, m_i, n_i の音高を p_{ni}, 音長を t_{ni} とし, 以下の計算式

$$c = |p_{ni} - p_{mj}| + \frac{|t_{ni} - t_{mj}|}{2}$$

で求める. この式で求まった各音符間のコストを計算してできた行列を表にまとめたのが図 2.7 である.

図 2.7 から, 左上のセルから右下のセルまで, コストが最小になるように経路を考えてい

a \ b	69-4	65-4	67-4	69-2	70-2	72-4	70-8	69-8
65-8	6	2	4	7	8	9	5	4
65-4	4	0	2	5	6	7	7	6
67-4	2	2	0	3	4	5	5	4
69-8	2	6	4	3	4	5	1	0
72-8	5	9	7	6	5	2	2	3
70-8	3	7	5	4	3	4	0	1
69-8	2	6	4	3	4	5	1	0

図 2.7 コストの行列

く (この場合移動できる方向は右, 下, 右斜め下の 3 パターンである) とこの 2 つの音符列を比較した総コストが計算される. この時に DP を用いて最適経路を計算することになる. その結果求まった行列が図 2.8 である. この図で, 行と列が 1 つずつ増えているが, これは 1 音挿入するコストをヒューリスティクス値 4 として表したからである, つまり一番上の行と左の列は 4 の倍数で増えて行くことになる. 図 2.8 において, 太字で表した数値が最適経路であり, 一番右下のセルがこの 2 つの音符列を比較した際の総コストということになる, このような手順で他の音符列についてもマッチングを行いそのコストで音符列の類似度を比較していく.

a \ b									
	0	4	8	12	16	20	24	28	32
	4	**6**	6	10	14	18	22	26	30
	8	8	**6**	8	12	16	20	24	28
	12	10	10	**6**	10	14	18	22	26
	16	14	14	10	**9**	**13**	18	19	22
	20	18	18	14	13	14	**15**	19	22
	24	22	22	18	17	16	18	**15**	19
	28	26	26	22	21	20	21	19	**15**

図 2.8 コストの行列 (DP 適用後)

2.5.4 GTTM の応用に関する研究

平田, 青柳 [7] は GTTM のタイムスパン簡約の結果得られたタイムスパン解析木を元に 演繹オブジェクト指向データベース (Deductive Object - Oriented Database: DOOD) という音楽知識表現手法と, 事例ベース推論を用いパーピープンというアレンジシステムを構築している. これは入力として与えられた単純なコード進行をジャズ風にアレンジするシステムである.

2.5.5 先行研究

本研究では, 井田 [8] による GTTM の自動化研究を元に, より精度の高いグルーピング構造分析システムの構築を目指した. 井田によるグルーピング構造分析プログラムのアプローチについては 3.3 節でその問題点とともに述べる.

第 3 章

グルーピング構造分析の自動化

本章では, グルーピング構造分析を自動化するに際しての問題点とそれを解決するアプローチについて述べる.

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, SMF(Standard MIDI File) 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.

SMF について

SMF は MIDI(Musical Instrument Digital Interface) プロトコルで規定されたデータ形式で保存されるデータファイルである. MIDI データは MIDI 対応機器間でやり取りするためのデータである. ここでは MIDI 規格で定められた MIDI メッセージの中, 本研究に関連する項目について文献 [1] の中から簡潔に述べておく.

- MIDI メッセージの種類

  まず, MIDI メッセージの分類について述べる. MIDI メッセージは図 3.1 のように大

きく分けて2つのメッセージに分けられる. MIDIでは 最大16パートをコントロー

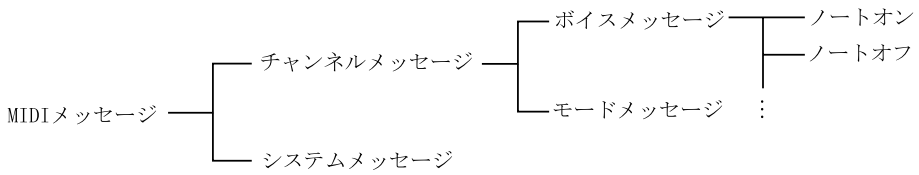

図 3.1 MIDI メッセージの分類.

ルするためにMIDIチャンネルという概念があり, チャンネルメッセージは, ある特定のMIDIチャンネルに対して発行されるメッセージである. システムメッセージはMIDIチャンネルに関係なく全パートに発行されるメッセージである. チャンネルメッセージはさらにボイスメッセージとモードメッセージに分けられる, このうちモードメッセージはMIDIモードと呼ばれるモードを切り替えたりするために用いられるメッセージであるが, 本研究との関連性がないためここでは説明を省略する. ボイスメッセージはノートオン, ノートオフ (音を出す, 止める) などがある. 以下ではそのノートオン, ノートオフについて説明する.

- ノートオンメッセージ

  ノートオンメッセージは3.2のように3バイトのデータで表される. はじめの1

メッセージ	ノートオン(note on): チャンネルボイスメッセージ
フォーマット	3Byteのデータで表される. 1バイト目:ステータス　2バイト目:データ　3バイト目:データ 9 | n　　ノートナンバー　　ベロシティ n　　:チャンネル1～16(0ｈ～Fh) ノートナンバー　:0～127(0～7Fh) ベロシティ　　:0～127(0～7Fh)
動作	発音(ベロシティ0の時は消音)

図 3.2 ノートオンメッセージ

Byteはステータスバイトである. ステータスバイトの9n の9 はこのメッセージがノートオンメッセージであることを表し, n には1から16までのチャンネル番

号が入る. 2バイトで表されるデータバイトの1バイト目は音高を表すノートナンバーである. ノート番号は音高を数値で表したもので, ピアノの中央のC(ド)を番号60として半音ごとに1つづつ増減していき, 一番高い音が127, 一番低い音が0となる (図3.3).(これより音高番号60のCを図3.3のようにC4と表すこととし, 他の音も同様に, 音高番号52のEならE3などのように表す, また, C4から半音上がった音をC♯4(またはD♭4) と表すこととする.)2バイト目のベロシティとは鍵盤を押す速さのことで, つまりその音の強さを表したパラメータである. ベロシティは最大で127, 最小で0であるが, 0のときは消音を表し, 次に述べるノートオフメッセージと同じ効果を表す.

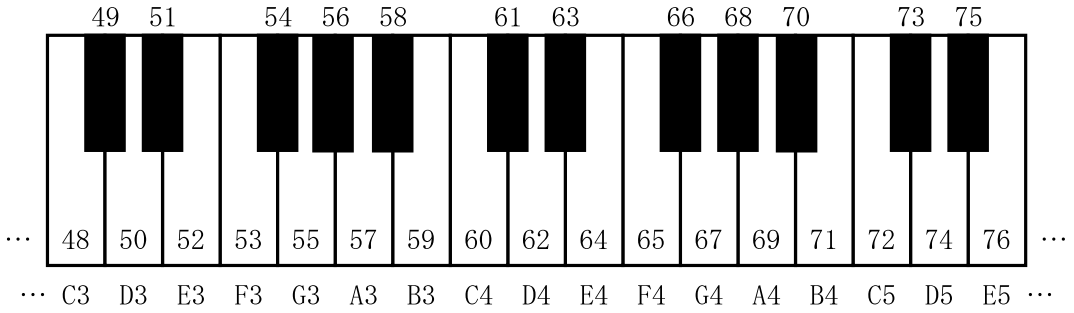

図 3.3 ピアノの鍵盤とSMFでの音高表記の関係

- ノートオフメッセージ

  ノートオフメッセージは発音された音を止めるためのメッセージである, ステー

メッセージ	ノートオフ(note off): チャンネルボイスメッセージ
フォーマット	3Byteのデータで表される. 1バイト目:ステータス　2バイト目:データ　3バイト目:データ 8 | n　　ノートナンバー　　ベロシティ n　　:チャンネル1～16(0h～Fh) ノートナンバー　:0～127(0～7Fh) ベロシティ　　:0～127(0～7Fh)
動作	発音停止

図 3.4 ノートオフメッセージ

タスバイトの 8n の 8 はノートオフメッセージであることを示す. データバイトの 1 バイト目はすでにノートオンメッセージによって発音されているノートナンバーを表す. ノートオフベロシティはノートオンベロシティとは反対に音を消す速さを表すが, このパラメータはあまり使われない.

- デルタタイム
  デルタタイムはこれまでに述べた MIDI メッセージとは少し異なる. デルタタイムはすべてのイベントに付随する可変長データで, ノートオンメッセージに付随すると音がなっている長さを表す.

- MIDI フォーマット
  本研究で用いる SMF は MIDI フォーマット 0 のものを用いる, MIDI フォーマットとは, 複数のトラックを扱えるかどうかを表すものでフォーマット 0 は単一トラックのみのフォーマット, フォーマット 1 は複数トラックが扱えるフォーマットである, さらにフォーマット 2 というものもあるがこれはドラムマシンのためのフォーマットであり普段はあまり使われることはない. 前述の MIDI チャンネルとトラックは別物である, 簡潔に説明を述べるとトラックは同じチャンネル内で複数存在することができ (MIDI フォーマット 1 以上の場合), 同じ音色のパートを複数に分割したものである. つまりトラックはオーケストラでいう一人の人間のようなもので, MIDI チャンネルは同じ楽器を持ったグループということになる.

これまでに述べたように SMF は音がなる, 音が止まるというデータを時間順に記述しただけのデータである. これはたとえるならばピアノロールのようなものであり, 楽譜から人間が読み取れるデータと SMF を比べた際に, 例えば調の情報はメタインベントというデータとしてファイル内に記述することは可能であるが小節線や 5 線譜のような情報を持っていない. SMF を解析データとして扱うときにはこのような点に留意しなければならない.

3.2 GTTM のグルーピング構造分析が持つ問題点

GTTM は元々, 計算機上に実装されることを目的とした理論ではないため, その自動化に際してはたくさんの問題点がある. ここではグルーピング構造分析に見られる問題

点について述べる.

3.2.1 規則の優先度についての問題

GPR の各規則は箇条書きされているものの各規則間の優先度については言及されていない. これは例えば GPR1 と GPR5 に見られる, この問題は図 3.5 のように等しい長さの二つの部分と単音のグルーピングの競合が起こった場合どちらを優先するのか, という問題である.

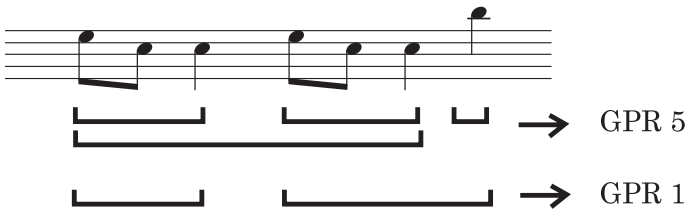

図 3.5 規則の競合

3.2.2 用語定義の曖昧性

GTTM の規則の中には, 曖昧と思われる表現が見られ, これが計算機上の自動化を実現する際に大きな障害となっている. ここではグルーピング構造分析においてそのような曖昧な表現が使われている個所について述べる.

- GPR2, 3 について.

  GPR2, 3 は連続する 4 つの音符からある条件が成り立つときに 2 つ目と 3 つ目の音符の間にグルーピングの切目がある. というルールであるが, 連続した 4 つの音符はどのように見つけるのかという問題がある. これは特に入力データ形式を SMF にする場合大きな問題となる. 旋律が 1 つしかないような楽曲の場合, 時間順にデータ上の音符データたどっていけば良いが, ポリフォニーからなるような楽曲の場合は時間順に拾った 4 つの音符列が GPR2, 3 で評価するべきかどうかという問題が生じる (図 3.6 参照).

- GPR4 について.

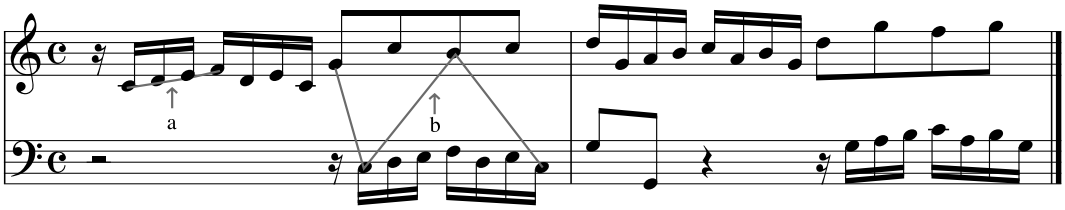

図 3.6 連続する 4 つの音符の選択 (a: 正しい例, b: 間違った例)

GPR4 は GPR2, 3 の分析結果を元にその効果が比較的明白なところは大きなレベルでもグループの境界がそこにある. と言う規則であるが, ここで “比較的明白” という表現について, どのように GPR2, 3 が適用されているのが効果が明白とみなされるのか, が問題となる.

- GPR5 について.

  GPR5 はグループの分割が長さの等しい 2 つまたはそれ以上の部分からなるようにグルーピングすることを優先するという規則である. この規則についての問題点は GTTM の 50 頁で述べられている (図 3.7).

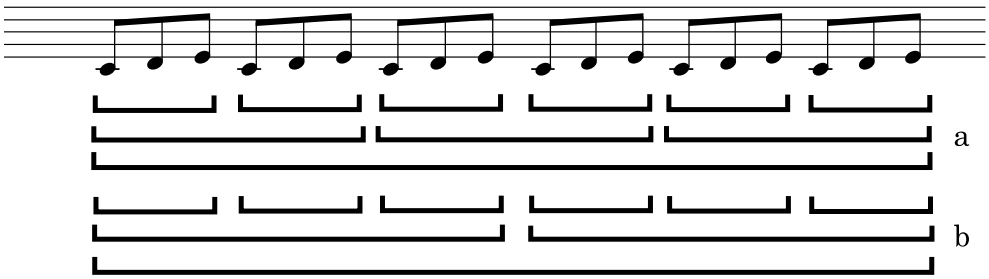

図 3.7 GPR5 の解候補

図 3.7 において, 中間のグルーピングが a のように 2 つにするか, b のように 3 つにするかで曖昧性が生じる. GTTM では拍節的な情報, または調性的な情報により解決すると述べていて, GPR5 単体で処理するならば, 3 つをグルーピングする b よりも 2 つをグルーピングする a を選択した方が安定するだろうと述べているが, この理由だけでは曖昧性の問題を解決したとは考えにくい.

- GPR6 について.

GPR6 はグループ間で並行した部分があれば並行性のあるグルーピングを行うという規則であるが, ここでは並行性の定義が曖昧となる. 図 3.8 のように反行 (Inversion), 逆行 (Retrograde), 反行＋逆行 (Retrograde Inversion) という構造や, 3.9 のようにオリジナルの音符列と比較して, ある音符列の途中で音符が削除されているような場合 (deletion), ある音符列とある音符 列の途中に音符が挿入されている場合 (insertion), 途中の音符の性質が変わっている場合 (replacement) などが起こる. 音楽ではたびたびこのような構造が見られ, どの程度まで並行性のある部分と見なすかが問題となる.

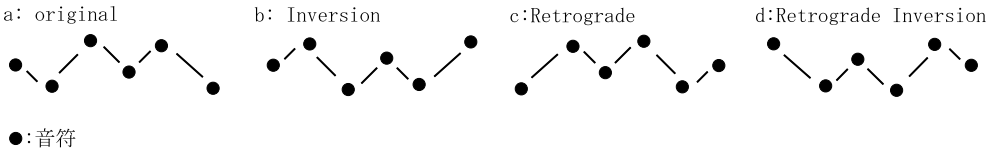

図 3.8 並行な音符列と見なされる候補 1

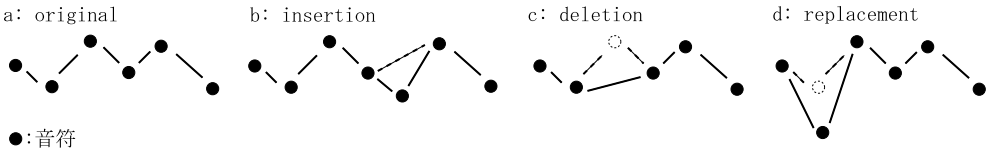

図 3.9 平行な音符列と見なされる候補 2

3.3 先行研究のアプローチと問題点

井田 [8] 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, 以下のようであった.

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.

井田は GPR2, 3 の実装のためのヒューリスティクスを “暗黙的パラメータの明示化”

として下のように定義した. これは文献 [2] の中で 20 個例題を選び, 決定したものであると述べられているものである.

- DifferentTimeIntervalGPR2b : 1.3
  これは GPR2a において, (4 つの音符を n_1, n_2, n_3, n_4 として)n_1 のオンセット時間と n_2 のオンセット時間の間および, n_3 のオンセット時間と n_4 のオンセット時間の間よりも, n_2 のオンセット時間と, n_3 のオンセット時間の間が 1.3 倍以上長い時に適用されるということを表したパラメータである.

- SameVelocityNoteGPR3b : 1.1
  これは GPR3b において, n_2, n_3 間の音の強さの差が, n_1, n_2 間と n_3, n_4 間の音の強さの差より 1.1 倍以上大きい時に GPR3b を適用するということを表したパラメータである.

- SameDurationNoteGPR3d : 1.1
  これは GPR3d において, n_2, n_3 の音長の差が, n_1, n_2 と n_3, n_4 の音長の差より 1.1 倍長い時に GPR3d を適用するということを表したパラメータである.

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.

- GPR4, 5, 6 が未実装.
  GPR2, 3 でグルーピングの切目を分析した後, それらの切目を判定することと, 上位のグルーピングを行うための規則群である GPR4, 5, 6 が 未実装となっている. そのためグルーピング構造解析が不十分である.

- GPR7 が未実装.
  拍節構造解析と, タイムスパン簡約の自動化がで完全にできていないため, GPR7 の実装が困難となっている, そのためグルーピング構造解析のための情報が不足している.

3.4 グルーピング構造分析のアプローチ

本節では, GPR の各規則について, 本研究ではどのようなアプローチをとって実装したかを述べる.

3.4.1 GPR2, 3 実装のためのアプローチ

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.
本研究では声部切りわけ処理を含めた 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. 以下では (1)-(3) の内容について GPR2, 3 の定式化とともに詳しく述べる.

3.4.2 述語 κ_S , $change$ 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ぶ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〜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.

3.4.3 GPR2,3 の定式化

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.

$\exists n_2, n_3$ if $\forall n_1 n_4 \quad \kappa_S(n_1, n_2, n_3, n_4) \cap change(n_1, n_2, n_3, n_4)$
$then \quad n_2, n_3$の間をグループ境界とする.

κ_S の実装には 2 つの手法が存在する. 1 つ目の手法は最初に考えられるだけ隣接音接続のリンクを作っておき, 条件にしたがって隣接音接続を切断していく手法であり, 2 つ目の手法は逆に隣接音接続をゼロから結んでいく手法である. これら 2 つの手法は互いに等価である. 従って, 前者と後者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 [8] の手法は後者であり, 本分析システムでは前者の手法を採用している.

3.4.4 GPR4 実装のためのアプローチ

GPR4 は本研究では GPR2, 3 の結果を利用してその強さを数値化し上位のグルーピングの切目を作り出すルールであると解釈して実装する. 本研究ではまず 6 つある GPR2, 3 のルールが何個適用されているかを単純に数値化し, その適用数に応じて階層化したグループ構造をつくリ出す, その後設定されたヒューリスティクス値によりどのレベルのグルーピングを GPR4 を適用したグルーピング構造にするか決定する. この処理で作られたグルーピングの切目は下位のどのレベルでも必ずグルーピングの切目として認識される. 図 3.10 は GPR2, 3 がこのように適用されているとき, GPR4 の閾値を 3 に設

定した例である. GPR4 で作り出したグルーピングより下位のグルーピング構造の分析については, GPR6 などで行う.

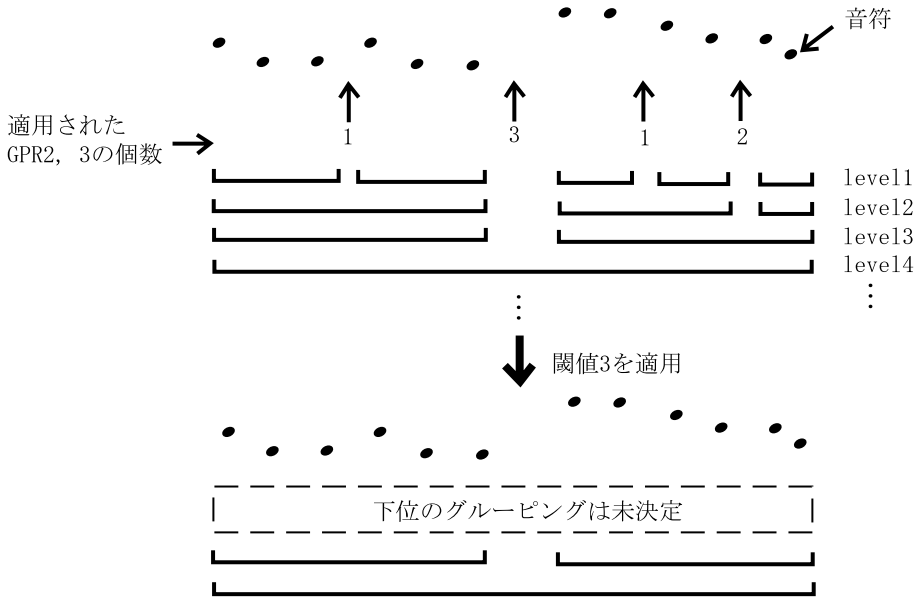

図 3.10 GPR4 のアプローチ

3.4.5 GPR6 実装のためのアプローチ

本研究において GPR6 は 2.5.2 節で述べた手法である DP Matching を用い実装する. パターンマッチングアルゴリズムは他にも様々なものが選択可能であるが, 例えば KMP 法などのストリングマッチングアルゴリズムはこの場合適用しにくい. これは楽譜上の音符のデータには音高と音長という 2 つのパラメータがあるためである. そのため, 音符のコストを正しく設定することで計算された総コストでの比較を意味のあるものにすることができる DP Matching を採用する. さらに 2.5.2 節で紹介した Lloyd A. Smith ら [10] によるマッチング法を音高がずれても対応できるようなアルゴリズムに改良する. これは楽曲中には図 3.11 のように形が似ていても音高 が違うというパターンが頻繁に出現するためである. 具体的にはコスト c を以下のように定める. 音符列 n, m の i 番目の音符を n_i, m_i, n_i の音高を p_{ni}, 音長を t_{ni} とし, 以下の計算式

$$c = |p_{ni} - p_{mj} - (p_{n1} - p_{m1})| + \frac{|t_{ni} - t_{mj}|}{2}$$

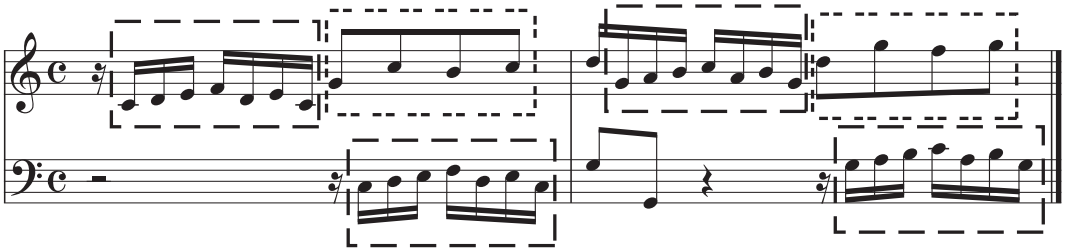

図 3.11 形が似ていて音高が違う例

で求める．本研究では旋律切り分けプログラムと, GPR4 の結果を元に並行と見なされる部分を見つけ出す．ここではすべての並行な部分を見つけ出すために音符 1 つのレベルから全探索を行う．使用したパラメータに関しては 4.6 で述べる．また, GPR6 の結果階層化した構造が現れる可能性がある (図 3.12)．このような時はそのまま階層化した構造として見なすこととする．さらにその階層化した構造が中間部分で競合を起こしていることがある (図 3.13)．この場合はこの時点で競合を解決せず, 拍節構造分析の結果により解決する．

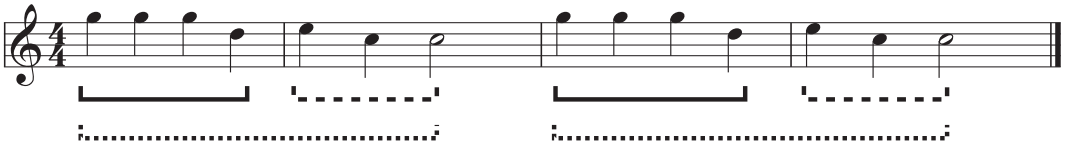

図 3.12 GPR6 の階層化例

3.4.6 グルーピング構造木作成

GPR4 と GPR6 の分析結果を用いてグルーピング構造木を作る．グルーピング構造木を作る処理はトップダウンかつボトムアップに行う．トップダウン方向は GPR4 の結果から求める．最上位レベルのグループは κ_s にて切り分けられた声部とする．ボトムアップ方向は GPR6 で平行な部分と認識されたグループを用い, GPR1, 5 を適用して上位方向へグルーピングしていく, GPR1 は単音のグループになっているグループを探し, 左右どちらかにあるグループと接合する．この時, GPR1-6 内で解決できないような選択がある場合このグループは単音のグループのまま処理する (つまり GPR1 を適用しない),

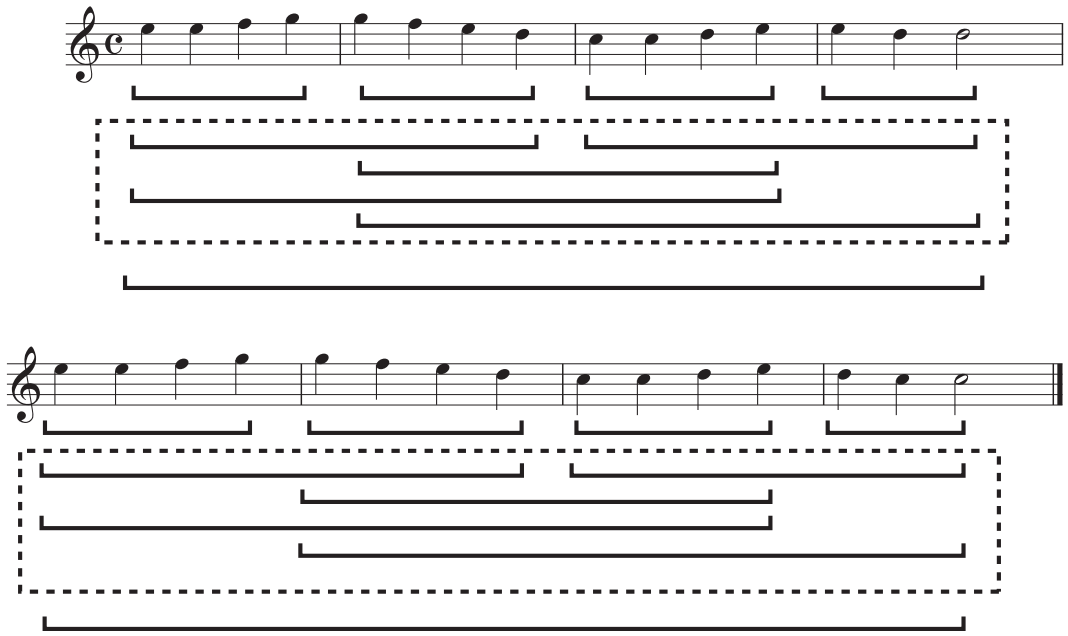

図 3.13 GPR6 の競合例

GPR5 は時間的な長さが等しい, または等しいと見なされる 2 つ以上のグループを見つけグルーピングする. GPR6 の結果は階層化した構造が現れることがあるのでこのような構造が表れた時はそのまま用いる. 競合を起こしている階層 GPR1-6 の中で解決できるものだけを解決する. 以上 2 方向からのグルーピングを組み合わせてグルーピング構造木を作成する.

第 4 章

実装と実験

4.1 システムの概要図

本研究では前章までに述べた各 GPR に対するアプローチを図 4.1 のような手順で組み合わせることにより, グルーピング構造分析の自動化を実現した. 4.2 節以降で各モジュールについて解説する.

4.2 前処理

本節ではグルーピング構造分析の自動化アルゴリズムとして本質的ではない部分の前処理について解説する.

4.2.1 xsmf

本研究ではSMF データを入力データとして用いているが, SMF はバイナリ形式のデータであるためそのデータを解析するプログラムを新たに作成する必要がある. しかしこの手間を省くため, 本研究では SMF をより扱いやすい XML 形式に変換する xsmf[1] というプログラムを採用した.

[1] 青柳 龍也 助教授 (津田塾大学) 作成

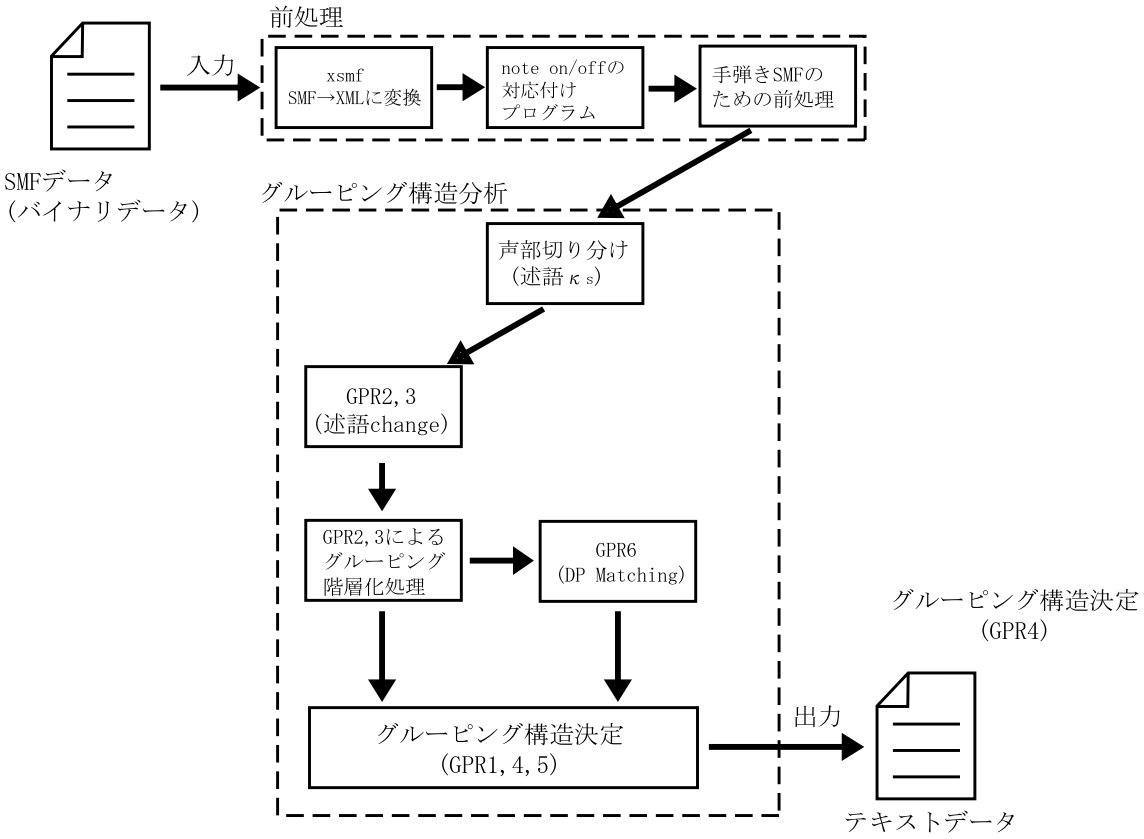

図 4.1 システム概要図

4.2.2 note on, note off の対応付けを行った note 情報を作り出すプログラム

さらに前処理として, XML 化された SMF データから noteon, noteoff の MIDI メッセージを対応させ音符の情報として XML タグ化するプログラムを適用する. このプログラムは井田 [8] により作成された. このプログラムは SMF の中でデルタタイムとして表現されている音符の音長を音符の音長値として duration というタグの中で表現できるようにしたもので, このプログラムにより, note というタグの中で, notenumber(音高), velocity(音の強さ), onset(オンセット時間, 音が鳴り始める時間), duration(音の長さ), noteID(音符の ID) を要素として持たせることができ, 音符を XML の文法に従って 1 つの楽譜上の単位として扱えるようになる.

4.2.3 手弾き SMF のための前処理

前処理の最後として, 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る (図 4.2).

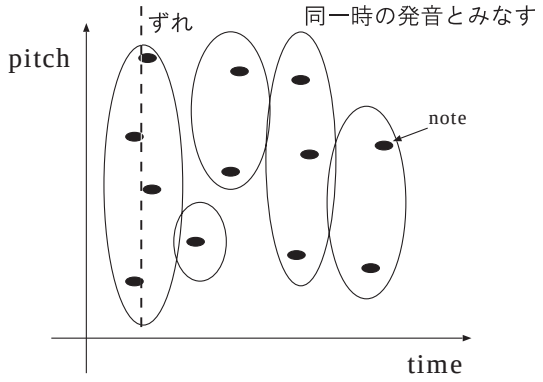

図 4.2 手弾き SMF のための前処理

4.3 κ_S の実装

3.4.3 節で述べた 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 κ_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 (図 4.3).

- 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 (図 4.3).

- 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.

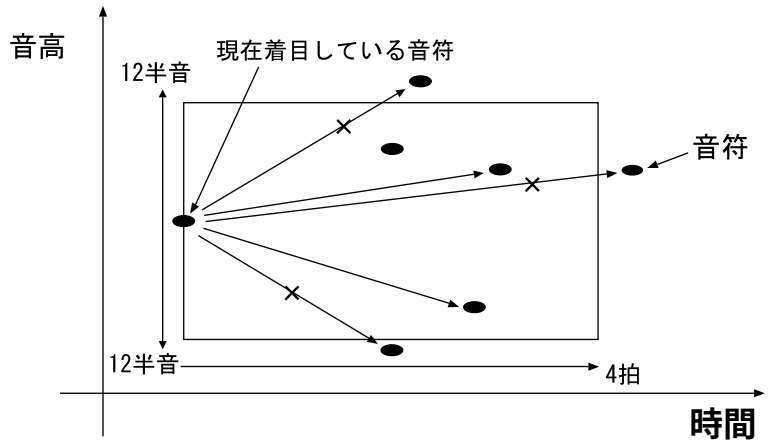

図 4.3 12 半音, 4 拍以内の音符をつなげる

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 4.4 a), また C, C♯ , D, D ♯のような時にも C から D や, C から $D\sharp$ のような隣接音接続は成立しないと仮定する (図 4.4 b).

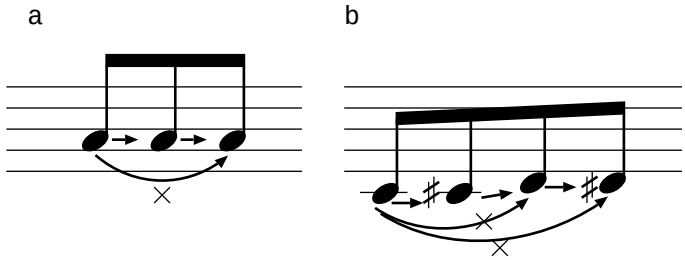

図 4.4 音高に関する隣接音接続判定

- 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)×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 4.5). 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.

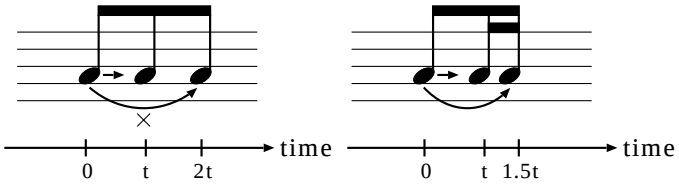

図 4.5 時間による隣接音接続判定

- 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さないと判定する (図 4.6).

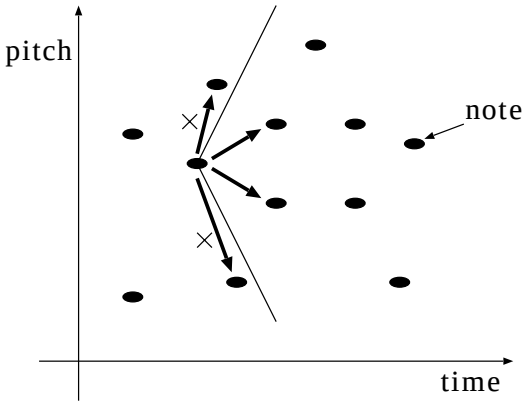

図 4.6 楔形の選択範囲の適用

- 音程/時間差比の導入.

  例えば, $...-C\sharp 4 - B\flat 3 - C4 - ...$ のようなメロディの隣接音接続には曖昧さがある. この問題に対して音程/時間差比の導入をする. 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 4.7). 現在はここまでに使ってきたヒューリスティクスをもとに 2 股以上のリンクが残る音符を検出し, 16 分音符の長さを時間の単位, 1 半音を音の高さの単位として, 以下の式のような計算式で長さを判定し一番短いリンクをの残す処理を行っている.

- 再結合.

  これまでのヒューリスティクスにより本来なら同じ声部や, 旋律であると見なされ

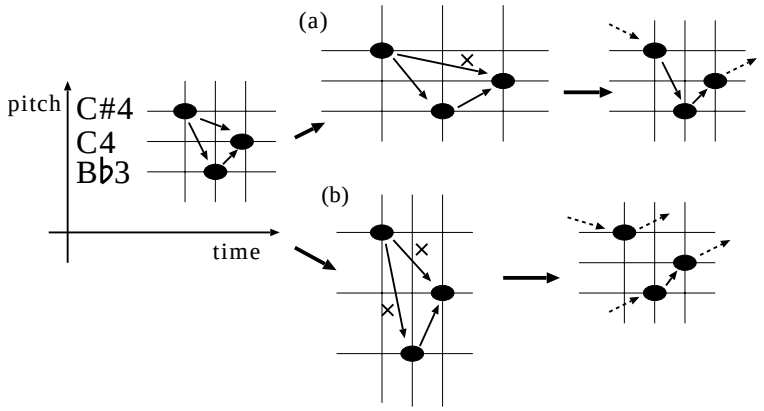

図 4.7 音程/時間差比の導入

る音符同士のリンクがとぎれてしまうことがある. そこでこのように途切れたリンクを見つけ出し, 再結合するヒューリスティクスを導入する.

4.4 GPR4のパラメータ

GPR4 の実装では 3.4.4 節で述べたアプローチの閾値 “3” を用いて実装した. この数値は 4.7 節で述べる GPR2, 3 の適用結果から, この値が適当であると考えたためである.

4.5 GPR5のパラメータ

GPR5 では比較する音符列同士のはじめの音のオンセット時間から最後の音のオフセット時間までの時間をそれぞれ T_A, T_B とすると T_B の長さが T_A の 0.8 倍から 1.2 倍の時, T_A, T_B を同じ長さであると見なし, そのような 2 つ以上のグループをグルーピングする.

4.6 GPR6のパラメータ

GPR6 の実装では, 以下のようなパラメータを用いた.

- 半音差につき 1 のコスト.
- 音長差の半分のコスト.

これらのパラメータはLloyd A. Smithら[10]で紹介されているパラメータをそのまま適用したものである.

- DP Matchingにより比較する音符列の最小の要素数を3とする.
  これはあまり少なすぎる音符列同士を比較しても意味がない, という仮定から2音以下のマッチングは行わないようにするというものである. 3という数値に特別な根拠は無い.

4.7 実験

本グルーピング構造分析システムのために用意したSMFは一般的なシーケンスソフト2種類を用いて作成した. 実験結果は声部切り分け処理後GPR2, 3適用後までと, 上位のグルーピング規則GPR4, 6適用後グルーピング構造木作成までに分けて行った. 以下に結果と考察を述べる.

4.7.1 声部切り分け処理とGPR2, 3の結果

本分析プログラムの内, 声部切り分けプログラムとGPR2, 3を適用するプログラムを実行した結果を以下に述べる. モノフォニーの楽曲として W. A. Mozart 作曲, 交響曲第40番ト短調K. 550 第一楽章 (はじめの2小節, 図4.8, RWC-MDB-C-2001-No. 2 [4]) を, ポリフォニーの楽曲として, J. S. Bach作曲Invention No. 1(はじめの7小節, 図4.9) に適用した結果を示す. またポリフォニーの曲で複雑に声部が絡み合う例として, J. S. Bach作曲 “O Haipt voll Blut und Wunden”(はじめの3小節, 図4.10) を解析した結果を示す. また, シーケンスソフトの違いにより解析結果がどのように異なるのかを調べるため, 別のシーケンスソフトを用いて作ったSMF (W. A. Mozart 作曲, 交響曲 第40番ト短調 K. 550 第一楽章) を入力として本システムにより解析を行った結果を示す (図4.11).

図4.8, 4.9, 4.10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. また, 図中に示される矢印と “2a, 3a” という記号はプログラムによりGPR2, 3が適用された個所を示す.

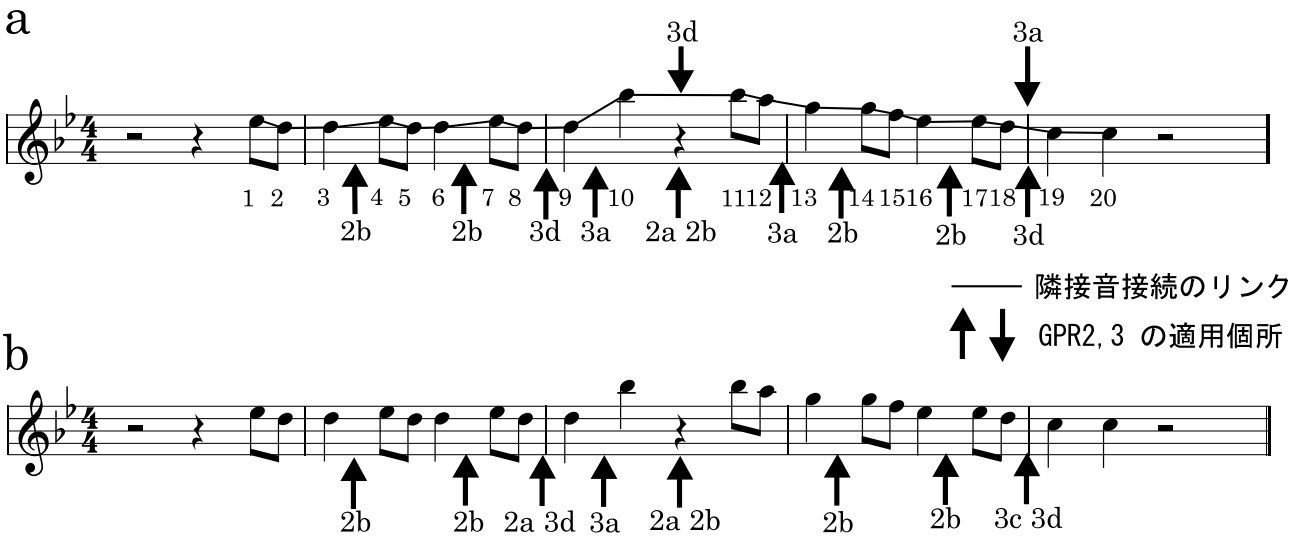

図 4.8“ 交響曲 第 40 番ト単調 K. 550 第一楽章 ”(a 本システムの分析結果, b 文献 [2], 47 頁)

図 4.8 は, 本分析プログラムの結果 a と文献 [2] での結果 b の違いを示している. 10 番目の音 (四分音符の B♭5) と 11 番目の音 (八分音符の B♭5) で b では適用されていない GPR 3d が適用されている, これは GPR2, 3 で適用されたヒューリスティクス値の設定のためであると考えられる. 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 4.10 について, 図中 a の部分で声部切り分けがうまく行っていないことが分かる. この楽曲はテナーとバスが 2 小節目のはじめの G3 で同じ音を歌う曲のため, 2 つの声部が G3 に集まるのが正解であるが, 2 つ以上の音符の隣接音接続を切り放すというヒューリスティクスのために D3-G3 と G3-A3 というリンクが切れてしまい, さらに再結合の処理のために D3-A3 のリンクが出来てしまったものである. このような声部が複雑に絡み合うような楽曲の場合, 分析に問題が残ることが分かった.

図 4.8 と図 4.11 との比較であるが, 図 4.11 では 3 と 4, 6 と 7, 13 と 14, 16 と 17 で GPR2a の適用がみられる. これはシーケンスソフトの特性で, デルタタイムの設定が異なってしまい, 別シーケンスソフトの方ではオフセット時間と隣接音のオンセット時間の間に自動的に休符を挿入してしまうためである.

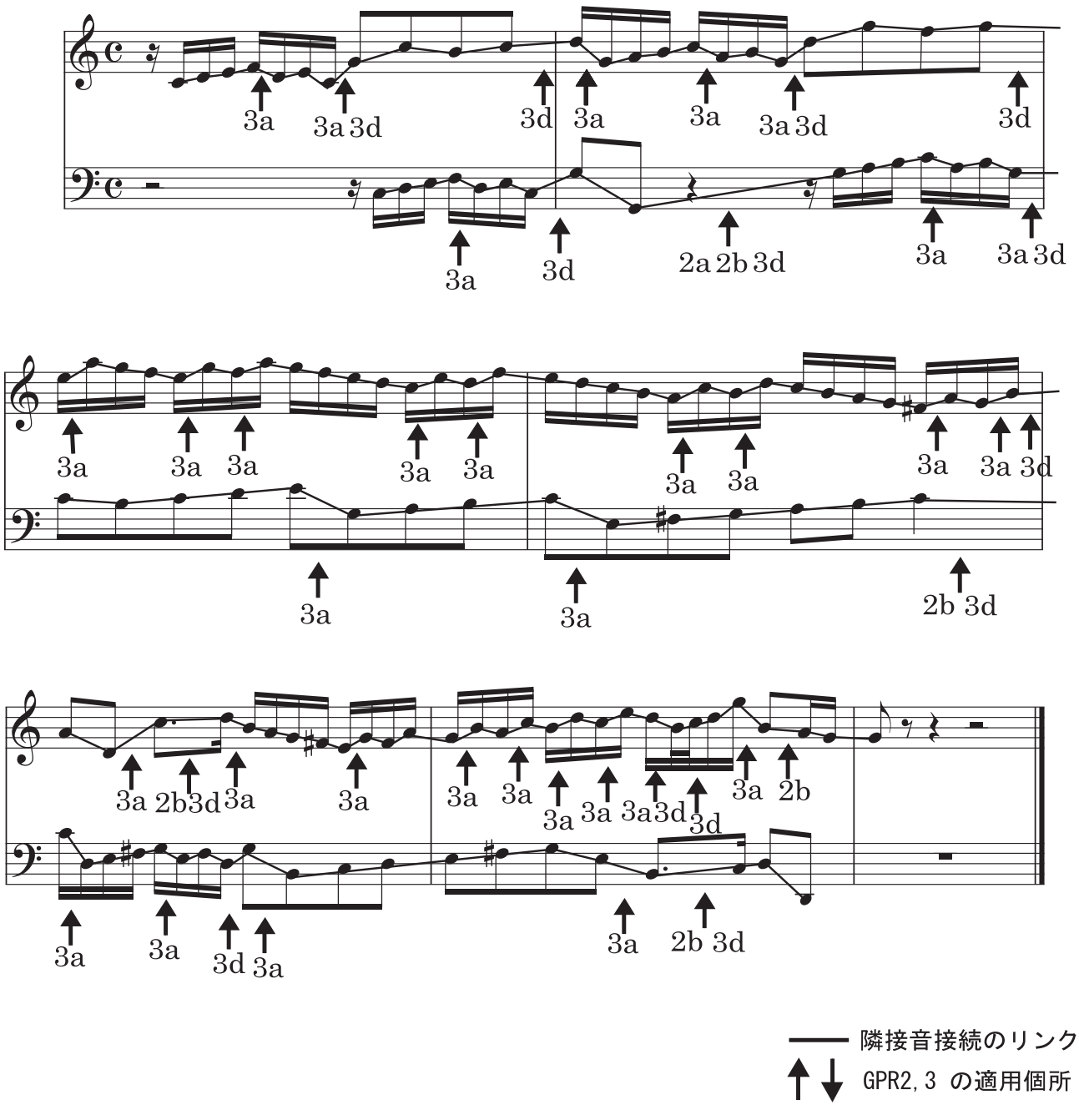

図 4.9“ Invention No.1 ”分析結果

4.7.2 上位のグルーピングのための規則実装の実験

次に上位のグルーピングのための規則である GPR4, 6 を適用し, グルーピング構造木作成プログラムを適用した結果できたグルーピング構造について述べる.

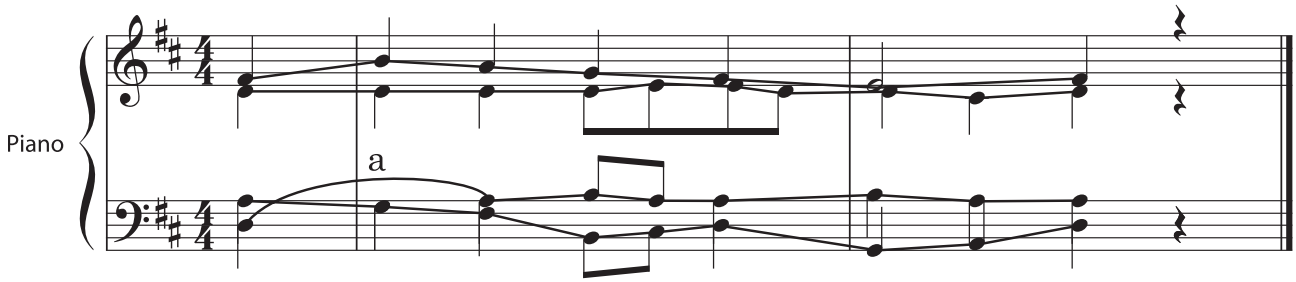

図 4.10 “O Haupt voll Blut und Wunden” 分析結果

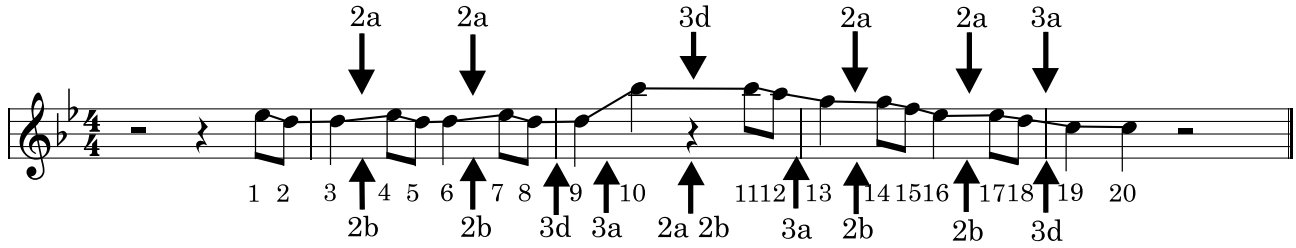

図 4.11“ 交響曲 第 40 番ト単調 K. 550 第一楽章 ”(別シーケンスソフトを用いた場合)

まず, モノフォニーの楽曲である W. A. Mozart 作曲, 交響曲 第 40 番ト短調 K. 550 第一楽章 (はじめの 2 小節, 図 4.12, RWC-MDB-C-2001-No. 2 [4]) の結果であるが, 図 4.12 中 i と ii は GPR6 で同じような作りであると判断されたグループである. iii は GPR5 にて同じ長さのグループ 2 つがグルーピングされた個所である. iv は GPR1 で単音のグループを左のグループへ接合して出来たグループである. 図中 b は GTTM 48 頁に書かれている図であるが, 本分析システムの出力結果である a と比べてみるとほぼ同じようなグルーピング構造ができあがっていることが分かる.

次に, ポリフォニーの楽曲である, J. S. Bach 作曲 Invention No. 1(はじめの 2 小節, 図 4.13) の出力結果について述べる.

図 4.13 中の i から vi は GPR6 によって同じ作りであると見なされたグループである. iv と v は競合しているが, GPR5 により iv と iii のグループが同じ長さであると見なされるため, v のグルーピングは消去される. i, ii と iv, iv, iii と vi は GPR6 により階層関係が作られている. vii は GPR4 によりグループ分けされた個所である. viii のグループが単音で残ってしまっているが, これは GPR1 が適用されなかったためである. この部分はすぐ右の i のグループに結合されるべきであるが, この部分を実現するには拍節構造分析

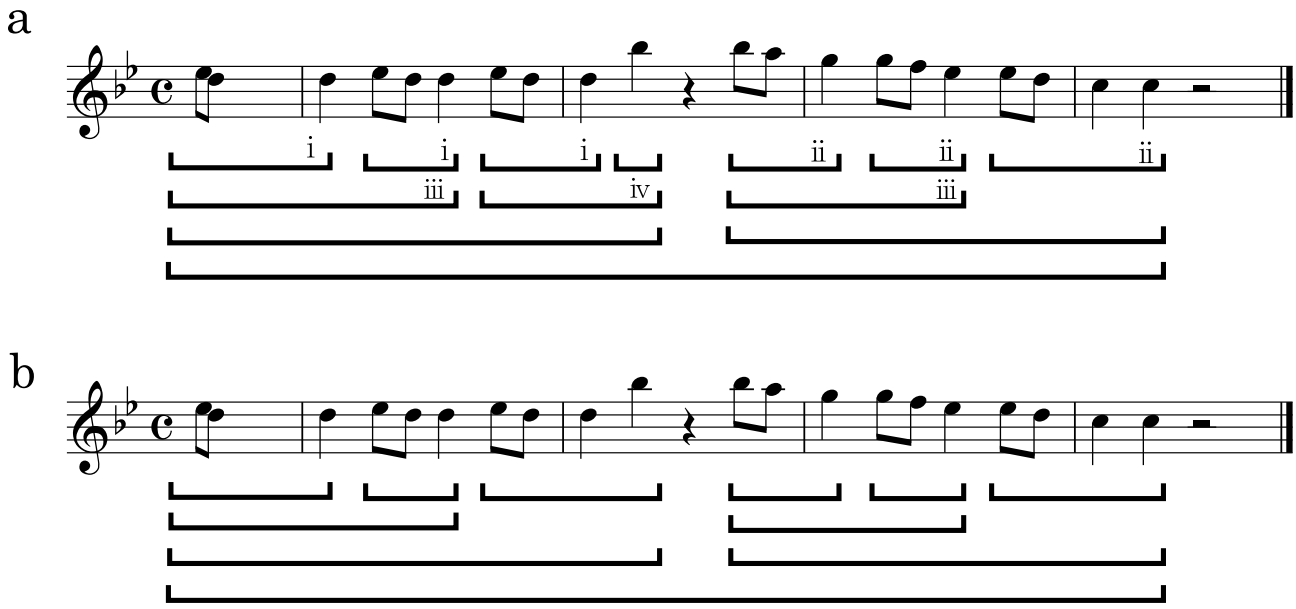

図 4.12“ 交響曲 第 40 番ト単調 K. 550 第一楽章 ”(a 本システムの分析結果, b 文献 [2], 48 頁)

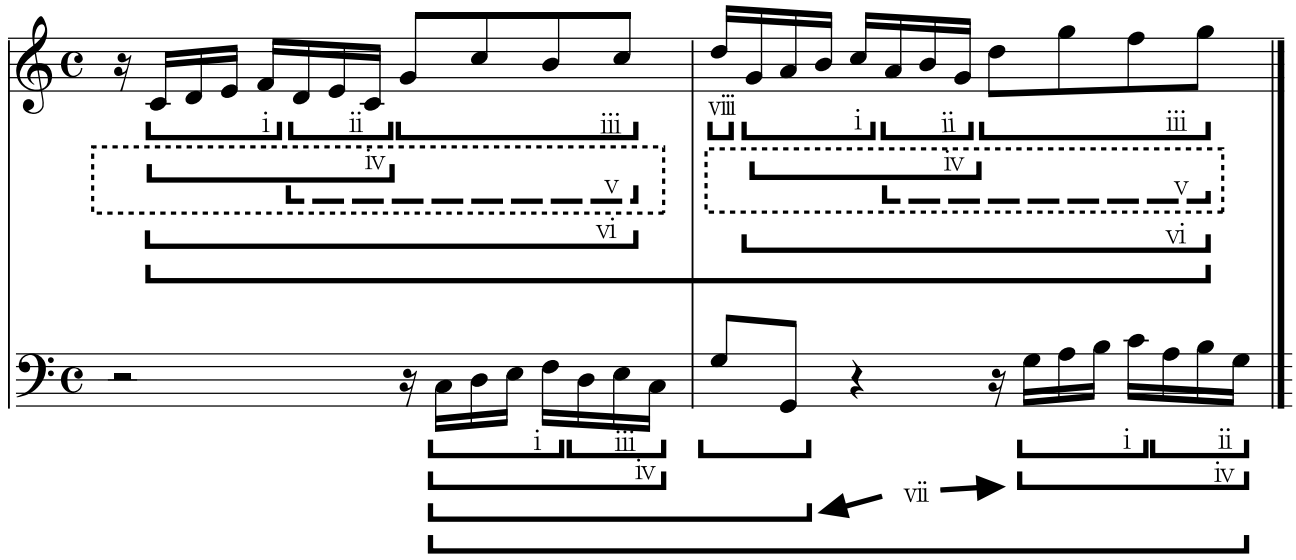

図 4.13“ Invention No.1 ”分析結果

の結果が必要になると考えられる.

第 5 章

まとめと今後の課題

5.1 本研究での成果

本稿で述べたことを以下にまとめる.

- グルーピング構造分析の自動化に関して, 新しい手法を提案した.
  先に挙げた問題点に関して GPR の適用順や曖昧性を持つ規則のパラメータ化により新しいグルーピング構造分析の手法を提案した.

- 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.
  声部切り分け処理によりポリフォニーの楽曲に対してモノフォニーの集合としてグルーピング構造を分析できるようにする方法を提案した.

- 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.

- GPR 2, 3 の定式化を行った.
  GPR2, 3 の定式化を通して, 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の切り分けを行うことができた.

- κ_S を実装した.
  ヒューリスティクスにより声部切り分け処理を実装した.

- GPR4 を実装した.
  閾値を設定することで, 曖昧な規則を数値化するという手法を提案した.

- GPR6 を DP Matching を用いて実装した.
  パターン認識アルゴリズムである DP atching を GTTM の規則に適用する手法を提案した.

5.2 今後の課題

これまでの研究で明らかになった, かつこれから解決しなければならない問題点は以下の通りである.

- 旋律切り分け処理の精度向上.
  これは図 4.10 の結果より明らかになった課題である. ポリフォニーからなる楽曲で, 音高が近いような個所が現れた場合声部の切り分けがうまく行かない可能性がある問題をどう解決するかということである. この問題の解決のためには κ_s のヒューリスティクス群の見直しが必要になると考えられる.

- GPR6 (DP Matching) の計算量の低減
  DP Matching を用いている, GPR6 は現在全探索アルゴリズムとっており非常に計算時間がかかってしまう. そこで, 計算時間を短縮できるアルゴリズムの適用が課題となる.

- GPR6 適用後の並行な音符列に対する競合の解決.
  これは, 図 3.13 のように, グルーピングの競合が起こった場合, 現状ではどちらのグルーピングを選択するのか, またはどちらも選択しないのか, を選択できない問題の解決のための理論の確率という課題である. この問題の解決には次で述べる拍節構造分析のグルーピング構造分析への応用が解決法の 1 つであると考えられる.

- 拍節構造分析結果のグルーピング構造分析結果への適用.
  これは, 図 5.1 のような構造をもつ楽曲のグルーピング構造を作るときに拍節的に安定しない構造を選んでしまう可能性があるという 問題点で, GPR7 より拍節構

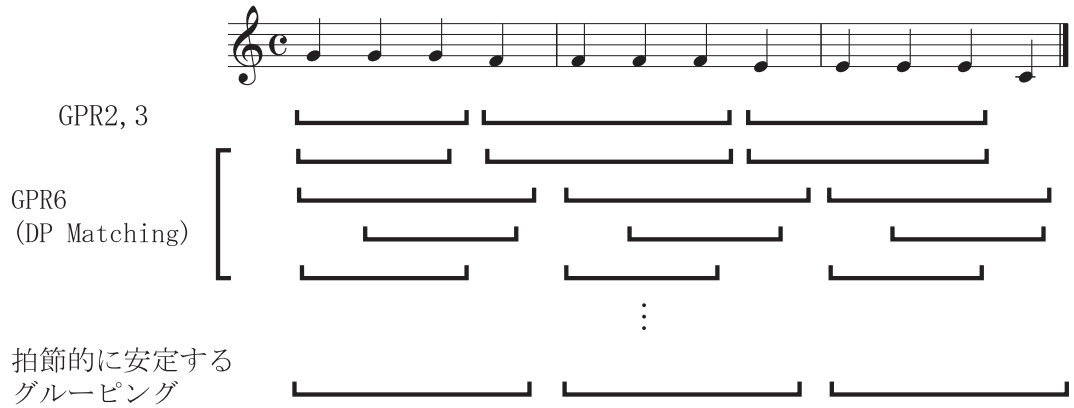

図 5.1 拍節構造分析の必要性

造分析のフィードバックが必要となる. これは実際に図 4.9 の 3, 4 小節目の下の声部で見られる問題である. この問題の解決のためには拍節構造分析の実装が課題となる.

- GPR 1 に残る問題の解決.
  GPR1 の問題の解決も今後の課題である. GPR1 は図 5.2 のように単音が選ばれてしまった場合どちらのグループに取り込むべきかが問題となる, これは拍節的に安定する構造を選ぶことが一つの解決法として挙げられる.

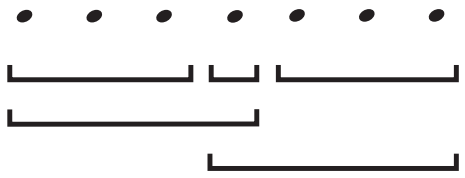

図 5.2 GPR1 の競合

- GPR4 の拡張.
  GPR2, 3 には規則間の優先度が無くそのため GPR4 の実装は GPR2, 3 の適用個数でしか判定することができない. これは正しいグルーピングを導かない可能性がある. そこで, GPR2, 3 の各規則に重み付けをし GPR4 の分析結果をより精度の高いものにする必要がある.

以上の課題を考え合わせて, 図 5.3 が正しいグルーピング構造分析手順だと思われる. このシステムを作り出すのが今後の課題となる.

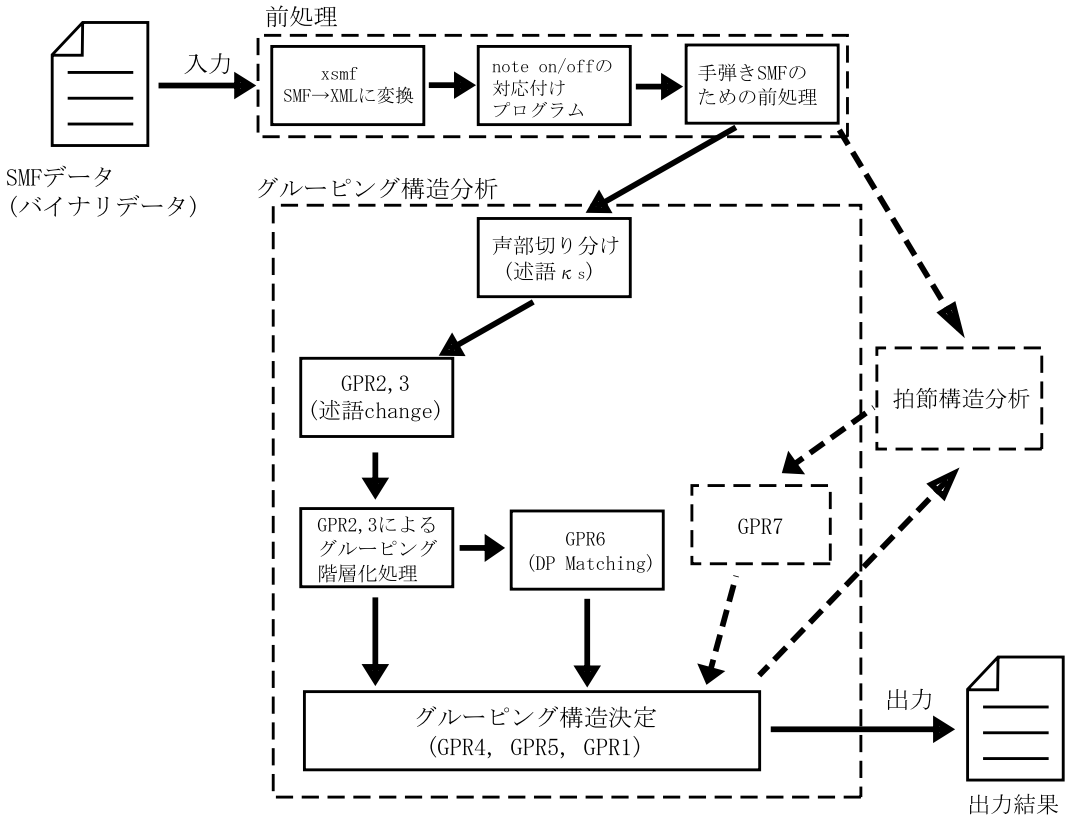

図 5.3 課題となるグルーピング構造分析システム

謝辞

研究の全般にわたって助言かつ指導をしてくださった平田圭二氏 (NTT コミュニケーション科学基礎研究所) に深く感謝致します.

また, 音楽知研究会において助言をいただいた佐藤健先生 (国立情報学研究所) に感謝致します.

最後に, 研究にご協力をいただいた鳥澤健太郎助教授, 永田裕一助手をはじめとした東条, 鳥澤研究室のメンバーに感謝致します.

参考文献

[1] Curtis Roads, コンピュータ音楽 - 歴史 テクノロジー アート -, 東京電機大学出版局, 2001.

[2] Fred Lerdahl, Ray Jackendoff, A Generative Theory of Tonal Music, The MIT Press, 1983.

[3] David Cope, Experiments in Musical Intelligence,

[4] 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.

[5] 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:ポピュラー音楽と著作権切れ音楽データベース, 情報処理学会 音楽情報処理研究会 研究報告, No. 42, pp. 35- 42, 2001.

[6] 平賀譲, 音楽認知研究の諸問題, 情報処理学会, 音楽情報科学研究会 研究報告, No. 6, pp. 15 -22 , 1994.

[7] 平田圭二, 青柳龍也, バービーブン: 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.

[8] 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.

[9] 片寄晴弘, 竹内好宏, 演奏解釈の音楽理論とその応用について, 情報処理学会 音楽情報科学研究会 研究報告, No. 7 , pp. 15- 22, 1994.

[10] Lloyd A. Smith, Rodger J. McNab, Ian H. Witten, Sequence-Based Melodic Comparision: A Dynamic-Programing Approach, Melodic Similarity, Computing in Musicology 11, pp. 101- 117, The MIT Press, 1998.

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

[12] Peter Halasz, COMPUTER SIMULATION OF TIME-SPAN REDUCTION, International Conference of Music Perception and Cognition(ICMPC), pp. 267- 268, 1994.

[13] 竹内好宏, 音楽の構造解析とその応用, コンピュータと音楽の世界, 共立出版, pp.224-240,1999.

[14] Tim Crawford, Costas S. Iliopoulos, Rajeev Raman, String-Matching Techniques for Musical Similarity and Melodic Recognition, Melodic Similarity, Computing in Musicology 11, pp. 73- 100, The MIT Press, 1998.

[15] 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.