パピプーーン: GTTM に基づく音楽要約システム

平田 圭二　　松田 周

NTT コミュニケーション科学基礎研究所　デジタル・アート・クリエーション

hirata@brl.ntt.co.jp　　shu@dacreation.com

あらまし

本予稿は現在開発中の音楽要約システム「パピプーーン」について述べる．パピプーーンは，傷の付いたレコード盤上での針飛びで生じる速聴きのような方法で要約を行う．ただし，その針飛びは任意の場所で生じるのではなく，楽曲の区切り個所において生じる．まず，generative theory of tonal music (GTTM) のタイムスパン簡約と演繹オブジェクト指向データベース (deductive object-oriented database, DOOD) に基づく手法でポリフォニーを表現する．次に，最小上界 (least upper bound, *lub*) という操作を定義する．*lub* は類似度を判定する際に重要な役割を果たす．パピプーーンの前処理の段階では，ユーザは専用ツール TS-Editor を用いてタイムスパン簡約に基づく課題曲分析を行う．オンライン処理の段階では，ユーザは主システム Summarizer を用いて要約を行う．Summarizer は楽曲の部分どうしの類似度を調べて課題曲の構造を発見する．ユーザが要約に際し削除すべき楽曲部分を同定すると，Summarizer はそこを削除し残り部分を接続する．パピプーーンは，ユーザとのインタラクションを通じて，課題曲全体の雰囲気を反映した質の高い要約を生成する．

Papipuun: Music Summarization System based on GTTM

Keiji Hirata　　Shu Matsuda

NTT Communication Science Laboratories　　Digital Art Creation

Abstract

This paper presents a music summarization system called ”Papipuun” that we are developing. Papipuun performs quick listening in a manner similar to a stylus skipping on a scratched record, but the skipping occurs correctly at punctuations of musical phrases, not arbitrarily. First, we developed a method for representing polyphony based on time-span reduction in the generative theory of tonal music (GTTM) and the deductive object-oriented database (DOOD). The operation, least upper bound, plays an important role in similarity checking of polyphonies represented in our method. Next, in a preprocessing phase, a user analyzes a set piece by the time-span reduction, using a dedicated tool, called TS-Editor. For a real time phase, the user interacts with the main system, Summarizer, to perform music summarization. Summarizer discovers a piece structure by similarity checking. When the user identifies the fragments to be skipped, Summarizer deletes them and concatenates the rest. Papipuun can produce the music summarization of good quality, reflecting the atmosphere of an entire piece through interaction with the user.

1 はじめに

音楽要約は，音楽システムの構成法を見直すきっかけとなるという意味において重要であり，新しい応用を切り拓くという意味で興味深いタスクである．

これまで多くの音楽システムは，作曲，編曲，演奏等の標準的なタスク実現のために開発されてきた．これらタスクは高次でかつ粒度が大きく，音楽の非専門家にとっては，何か窺い知れない未知の経験，技能，知識，才能等で満ち溢れているように感じられている．ここで我々は，作曲，編曲，演奏等のタスクがより低次で細かい粒度のタスク (例えば要約や検索) から構成される点に注目する．音楽システムがユーザに中 ∼ 小粒度のタスク (例えば要約や検索) を提供できば，ユーザは音楽的に細かいことを考慮せずにその中 ∼ 小粒度のタスクを組合せて高次のタスクをデザインできるだろう．このようなミドルウェアレベルを導入した音楽システムの構成法によって，非専門家でも音楽情報処理技術の恩恵を受けることが期待される．

近年，インターネットや Web 技術が普及しており，様々な知識や情報がこれらの上に分散的に蓄積されている．音楽要約や検索といったミドルウェアレベルのタスクとインターネット/Web 技術を組合せると，知的なカラオケ，ケータイの着信音，インタラクティブな音楽システムといった新しい応用が考えられる．これらはいずれコンテンツ産業を先導する技術として期待される．

音楽要約というタスクは，楽曲中で最も目立っている部分あるいは代表する部分を (自動的に) 見出すことである．音楽要約の手法は，これまで数えるほどしか提案されていないが，大きくオーディオ信号主体の方式と記号主体の方式の 2 通りに分けられる．前者の研究事例として Logan と Chu のシステム [11] がある．彼らは，楽曲中で最も目立って記憶に残る部分 (キーフレーズと呼ぶ) とは最も繰り返されている部分であると仮定し，抽出されたキーフレーズを

もって要約とするシステムを開発している．しかし，キーフレーズは聴取者の嗜好や感性に大きく左右されるので，この仮定は常に正しいとは限らない．さらに，Logan と Chu のシステムは，楽曲全体の雰囲気を反映したような要約を生成することができない．一方，記号主体の方式は，譜面レベルで表現された楽曲 (例えば SMF) を対象とし，オーディオ信号主体の方式より音楽理論の利用が容易である．Hurron[8] は記号主体の方式の 1 つである．

我々は，楽曲全体の雰囲気を反映したような要約の生成を目指す．このような要約を実現するためには複雑な楽曲構成を分析する必要があり，そのためには音楽理論が提供する音楽知識が必要となるので，必然的に記号主体の方式を採用する．

本稿では，現在開発中のプロトタイプシステム「パピプーーン」について述べる．パピプーーンは針飛びで生じる速聴きのような方法で要約を行う．ただし，その針飛びは任意の場所で生じるのではなく，楽曲の区切り個所において音楽的に正しく生じる．現在のパピプーーンの入力は，トルコ行進曲 (モーツァルト作曲) や Let It Be (ビートルズ) のピアノソロ編曲版等のピアノ曲に限定している (つまり，一種類の楽器による format 0 の SMF)．この制限は，音楽要約において解決すべき課題を全て含んでいるという意味において，一般性を損なっていない．その課題とは，ポリフォニーの表現，旋律の類似度判定を用いた楽曲構造の発見，針飛びすべき部分の同定，要約として残すべき部分の接続，である．

本稿の構成は次の通りである．続く第 2 章でポリフォニーの表現法について述べ，第 3 章で最小上界に基づく旋律類似度判定法を提案し，第 4 章でインタラクティブな音楽要約法を提案しプロトタイプシステムについて述べる．最後に第 5 章で結論を述べと今後の課題に触れる．

2 楽曲の表現

本章で導入する楽曲表現法の目的は，タイムスパン簡約に関するポリフォニー間の関係を記述して，それをポリフォニーの類似度判定に用いることである．

我々はまず，generative theory of tonal music (GTTM) [10] のタイムスパン簡約と演繹オブジェクト指向データベース (DOOD) [14, 9] に基づきポリフォニーを表現するデータ構造を設計した．GTTM は，計算機への実装に関して最も有望な音楽理論である．DOOD は理論的な基礎が確立された知識表現法の 1 つであり，従って形式的に取り扱い易い[1]．DOOD は素性構造 [1] とほぼ同じ機能を持つ．DOOD の包摂関係は一般に，具体–抽象の関係 (いわゆる `is_a` 関係) を表現するために用いられるので，GTTM において DOOD の包摂関係に対応するものはタイムスパン簡約であると規定するのが最も自然であると考える [5, 7]．リハーモナイザ [4, 2]，編曲システム [3]，演奏生成システム [6] も同じく GTTM と DOOD の枠組のもとで開発されている．

[1]DOOD は元々国際会議あるいは研究分野の名称として用いられていたが，本稿では知識表現手法の名称として用いる．

2.1 タイムスパン簡約を用いた旋律の抽象化と具体化

GTTM におけるタイムスパン木は二進木である．本稿では，重要な方の枝を primary 枝，そうでない方を secondary 枝と呼ぶ．図 1 左側に，簡単な旋律とそのタイムスパン木を示す．図中，primary 枝及び secondary 枝で支配される時間幅 (タイムスパン) (⟷ で示す) は，head と呼ばれる 1 つの音/和音で代表される (ここでは C4)．本楽曲表現法は，上述のタイムスパン木

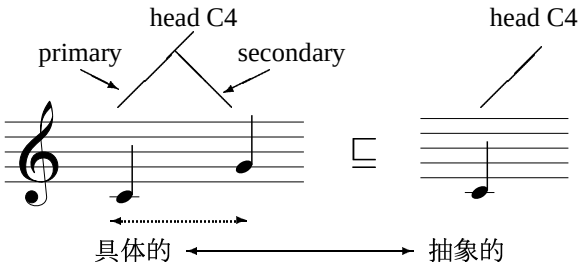

Figure 1: 旋律どうしの包摂関係

とポリフォニーに含まれる各音の時間情報を表現することができる．時間情報には，各音の発音時刻や音価だけでなく，注目している音とその周囲の音との時間関係も含まれる (図 1 では時間情報を表示していない)．

図 1 右側の旋律は C4 音のみから成り，同図左側の旋律より，タイムスパン簡約に関して抽象的と見なせる (左側の旋律は右側の旋律より具体的である)．この旋律の抽象–具体関係を包摂関係 (半順序) の一種と見なし，“$\sqsubseteq$” という記号で表現する．GTTM 分析を自動もしくは手動で行って，タイムスパン木の形，head の値，時間構造を決定しその分析結果を本楽曲表現法によって表現すると，タイムスパン簡約の抽象–具体関係が DOOD の包摂関係として自動的に表現される．例えば，図 1 の 2 つの旋律を本楽曲表現法によってあるデータ構造として表現すると，この 2 つの旋律間に包摂関係が成立つことが自動的に判定できる．もし旋律に異なる分析結果を与えると，タイムスパン木の形，head の値，時間構造も変わり，異なる包摂関係が成立する．

もし厳密に GTTM のタイムスパン簡約を解釈するなら，図右側は二分音符の C4 になるべきである．しかし，自動化を優先させたので，タイムスパン簡約を行った後の音は簡約前の音と同じ発音時刻と音価を持たせることとした [7]．

Lerdahl と Jackendoff [10] によると head の値には 4 通りの設定法がある．図 1 での head 値の設定法は “ordinary” であり，図 2 (アルベルティ・バス) での head 値の設定法は “fusion” である．head 値の設定法も分析された旋律の解釈に依存する．本楽曲表現法では，タイムスパン簡約の時に注目している音の発音時刻と音価を保存するので，周囲の重要でない音

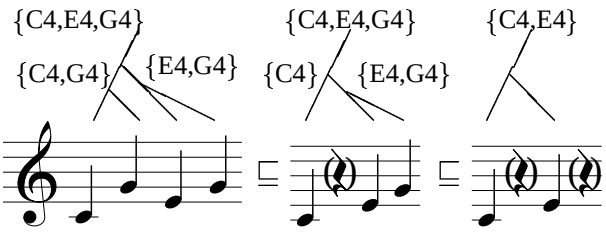

Figure 2: アルベルティ・バスの包摂関係

を削除すると，その音によって支配されていた時間幅は同じ長さの休符で置き換えられたように見える．そのため図中の休符には括弧を付けた．これは SMF における「休符」のようなものと言える．

以下，本稿では譜面のト音記号は省略する．

2.2 順序への簡約

本楽曲表現法の時間構造が依拠する直観は，発音時刻の時間差を抽象化すると発音順序になる，というものである．音高列としては同じだが発音時刻や音価は異なる 2 つの旋律を考える (図 3)．これら 2 つの

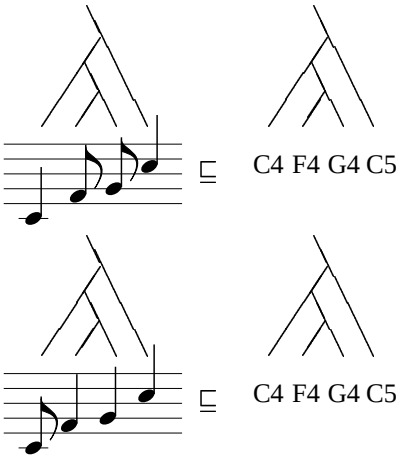

Figure 3: 異なる旋律の同一な音高列への簡約

旋律を聴取すると，我々は何か共通したものを認識するであろう．それは，図中 2 つの旋律に共通する “C4 F4 G4 C5” という同一の音高列である．このような観察から，ある旋律を簡約すると元の旋律と同じ順序の音の列が得られると考えて良いだろう．ここで，元の旋律とそれを抽象化した音の列 (時間的に簡約された旋律) は，同じタイムスパン木を共有していることに注意されたい．さらに，本楽曲表現法は部分的に順序に簡約された旋律も表現でき，そのような旋律と他の抽象的/具体的な旋律との間にも包摂関係が成り立つかどうかを決められる．本楽曲表現法では図 3 の包摂関係が成立するので，タイムスパン簡約の時間的順序的な側面に関して暗黙的に仮定されている事柄を陽に表現できるという意味において，旋律の形式化に成功したと言えるであろう．

2.3 ポリフォニーの定義

GTTM が扱う旋律は，理論的な理由でホモフォニーに限定されている．しかし，高い実用性のためには，ポリフォニーを (及びそれが持つ意味も) 扱えなければならないと考える．よって，我々はポリフォニーも扱えるよう GTTM のタイムスパン簡約を拡張する．基本的にポリフォニーとは，時には和声として響き合う独立で複数の旋律が複雑に絡み合って織り成すテクスチャのことである．しかし，我々の要約という目的に照らすと，より形式的な定義が必要である．

ホモフォニーは下位の旋律が時間的に直列に連なったもの (juxtaposition) として定義され，単旋律のように解釈される．次にポリフォニーを帰納的に定義する．まず，ホモフォニーはポリフォニーである．そして，時間的な重畳を許す 2 つのポリフォニーを直下の部分木として持つタイムスパン木もポリフォニーである．つまり我々の手法では，時間的な重畳を許す 2 つのポリフォニーの間に，タイムスパン簡約としての順序を付与する．図 4 では，ポリフォニーと拡張されたタイムスパン簡約の例を示す．図左側の各四角形は，元の楽曲をホモフォニーになるまで分解していった時に得られるホモフォニーを表す．図右側では，各ホモフォニーの head だけが，簡約前 (図左側) と同じ発音時刻と音価のまま残っている．ここで得られたホモフォニーへの分解と対応するタイムスパン木は，例題のポリフォニーを分析する 1 つの妥当なパターンに過ぎない．もし異なる分析パターンが得られた場合は，別の分解とタイムスパン木が得られる．

3 類似度の判定

包摂関係を用いて 2 つのポリフォニー間の類似度を判定するアルゴリズムを構築する．その中心的な演算は最小上界 (least upper bound, *lub*) である．直感的に，*lub* は 2 つのポリフォニーの最大共通部分を計算する．通常，$lub(x, y)$ は $\min(\{z | x \sqsubseteq z \wedge y \sqsubseteq z\})$ と定義される．全ての x, y について，$x \sqsubseteq lub(x, y)$ 及び $y \sqsubseteq lub(x, y)$ が成立つ．*lub* の数学的な意味は良く知られている．もし 2 つの共通部分が無いような全く異なる旋律が与えられると，*lub* の結果は，空つまり最も抽象的で情報量が少い要素になる ($\top$ と書く)．$\top$ は音として聴くことができないので，常に *lub* の結果を聴くことはできない．例えば，*lub* 計算結果に含まれるある音について，その発音時刻と音価の情報は確定しているが音高の値が $\top$ の場合，その音を実際の音として鳴らすことはできない．

我々の類似度の判定法とは *lub* を用いて 2 つの旋律の最大共通部分を計算することである．その 2 つの旋律に共通部分が多ければ多いほど，よりお互い類似していると判定される．Plaza [12] では，素性構造 [1] は事例の記述に適しており，*lub* は最類似の事例

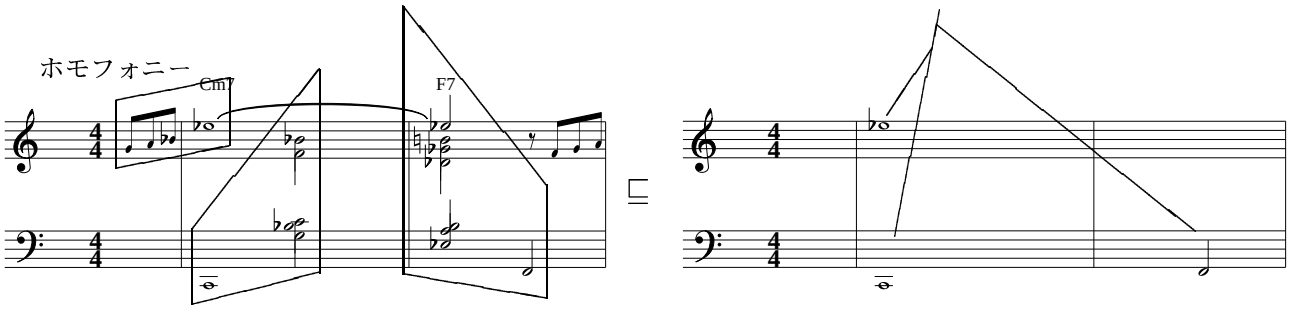

Figure 4: ポリフォニーのタイムスパン簡約

検索に適していると主張している (ただし Plaza の楽曲表現法は，我々のそれとは本質的に異なっている).

3.1 *lub* に基づく類似度

図 5 は，C4 G4 と C4 という単純な 2 つの旋律の *lub* 計算の例である．これら 2 つの旋律のタイムスパン木は示してあるが，時間構造は省略した．ここでは「旋律 C4 G4 ⊑ 旋律 C4」という包摂関係が成立しており *lub* の結果は旋律 C4 である．

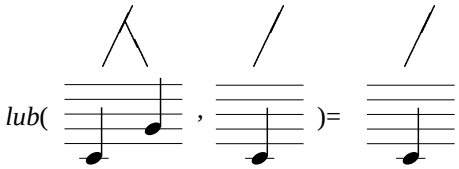

Figure 5: *lub* の簡単な例

図 6 は少し複雑な旋律の例である．2 つの旋律の中央の音は各々 D4, F4 であり，これらは音高やタイムスパン木に関して照合しない (従って時間構造に関しても照合しない) ので，*lub* の結果にはこれら D4, F4 の音は含まれない．

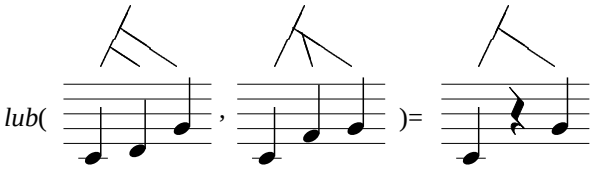

Figure 6: *lub* による最大共通部分の抽出

図 7 は，音高は同じだが音価が異なるような例である (図 3 に同じ)．この *lub* の結果は，音価の情報が確定しない C4, F4, G4 が音列を作り，最後が四分音符の C5 であるような旋律である．つまり，結果の抽象的な旋律は，入力の 2 つの旋律から得られる旋律の内，最も共通部分が多い旋律となっている．C4, F4, G4 は，音高は確定しているが，音価が不確定という意味で不完全である．

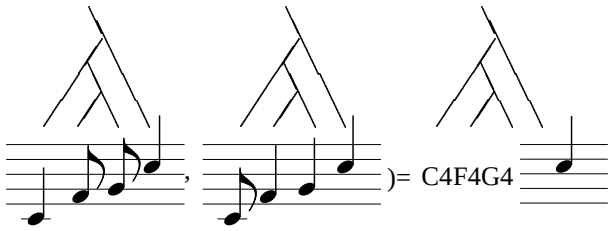

Figure 7: 抽象的な音と順序の情報

次に，ともに C4 を含む D4 C4 という旋律と C4 G4 という旋律を考える (図 8)．本楽曲表現法では時間情報も表現しているので，旋律の開始時刻を適切に揃えてから *lub* を計算する．図 8 では，C4 直前の小節線を揃えてから *lub* を計算している．本楽曲表現法で導入した時間構造は，auftakt を持つ旋律でも正しく取り扱える．

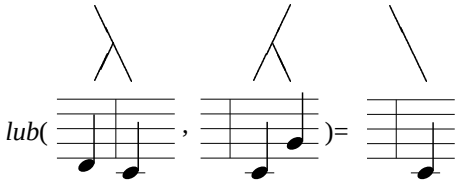

Figure 8: 時間的な整列

対照的に図 9 は，前例と同じ音列であるが開始時刻が異なる旋律の *lub* の結果を示している．開始時刻が異なるので結果は ⊤ である．

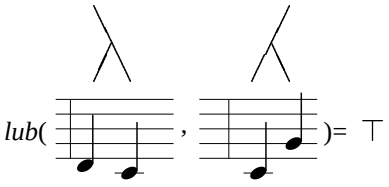

Figure 9: 整合性のない旋律の *lub*

ここまでの例では，説明を簡単にするため単旋律のみ扱ったが，本楽曲表現法の *lub* はポリフォニーにも同様に適用可能である．

3.2 類似度の尺度

lub の動作を直感的に述べる．*lub* は入力の 2 つの旋律の最大共通部分を計算するので，計算結果が入力と等しい場合は *lub* を計算したことによる情報の損失が無く，その 2 つの旋律は等価である (最も類似している) と考えられる．逆に計算結果が $\top$ の場合は，入力の旋律の情報が全て失われたということであり，入力の 2 つの旋律は無関係である (全く類似していない) と考えられる．以上の観察より，我々は *lub* 計算によって失われた情報の量をもって 2 つの旋律の類似度を計測することとする．失われる情報には，タイムスパン木に関するものと時間構造に関するものがあり，これらは我々が形式化したポリフォニーの 2 つの側面に対応している．

本稿で提案する類似度の尺度を記述するため，まず数学的な記法を導入する．P をポリフォニーとして，$|P|_N$ を P に含まれる音の総数，$|P|_A$ を P の note オブジェクトに含まれる属性の総数，$|P|_T$ を P の時間オブジェクトに含まれる属性の総数とする．1 つの音 (note オブジェクト) は 2 つの属性 (音高/音価と発音時刻) を持つので，well-formed ($\top$ を含まないよう) な P に関して $|P|_A = |P|_N \times 2$ である．同様に，1 つの時間オブジェクト (temporal オブジェクト) は 4 つの属性 (先行音，後続音，注目音，時間差分) を持つので，well-formed な P に関して $|P|_T = |P|_N \times 4$ である．

ここで，失われる情報を定量化する R_N, R_A, R_T という 3 つの類似度の尺度を導入する．P, Q をポリフォニーとして，

$$R_\$(P,Q) = \frac{|\mathit{lub}(P,Q)|_\$}{\max(|P|_\$, |Q|_\$)}$$

と定義する ($\$ = N, A, T$).

R_N と R_A はタイムスパン木に関連しており，R_T は時間構造に関連している．R_N は，不完全な音でも 1 つの音と見なして，音のレベルでのタイムスパン木の類似度を表している．R_A は，属性のレベルでのタイムスパン木の類似度を表しており，*lub* の結果に含まれる全ての音について音高と音価の属性がどの程度確定しているかを示す．同様に，R_T は，属性のレベルでの時間構造の類似度を表しており，*lub* の結果に含まれる全ての音の発音時間について上述の時間オブジェクトの 4 つの属性がどの程度確定しているかを示す．ただし，音の属性，時間構造の属性は全て同じ重みで定量化されている．

次に，これら類似度尺度が実際にどのような値を取るのかを調べる．$P = Q$ の場合は $R_N = R_A = R_T = 1.0$ である．逆に $\mathit{lub}(P,Q) = \top$ の場合は $R_N = R_A = R_T = 0.0$ である．興味深い場合として $P \sqsubseteq Q$ の場合を考える．$|P|_\$ \geq |Q|_\$$ ($ = N, A, T) なので，$R_\$ = |Q|_\$/|P|_\$$ となる．さらに，ここでは数値を算出する詳細な手順については説明しないが，図 6 の例では $R_N = 2/3, R_A = 2/3, R_T = 5/9$ となり，図 7 の例では $R_N = 1.0, R_A = 5/8, R_T = 10/13$ となり，図 8 の例では $R_N = 1/2, R_A = 1/2, R_T = 1/5$ となる．本楽曲表現法では，ポリフォニーの類似度尺度をこのように多面的に計算することが理解できよう．

従来の楽曲表現法や旋律の類似度判定法では，旋律の表層的な構造しか着目していなかった [13]．旋律の深層構造まで考慮するには簡約 (reduction) の概念を採り入れるのが有効である．Selfridge-Field は「旋律の検索や比較に適した簡約は，楽曲分析のために導入された Narmour の暗意-実現モデルの簡約ほど精密である必要はないだろう」と述べている [13]. Selfridge-Field は GTTM における簡約については触れていないものの，本稿で提案した我々の楽曲表現法は，Selfridge-Field の要請に対する 1 つの答えであると考えても良いだろう．

4 インタラクティブな音楽要約

第 1 章で述べたように，一般に音楽要約システムは次の動作を行う: (1) 類似度判定による楽曲構造の発見，(2) 残すあるいは削除する楽曲部分の同定，(3) 要約として残す楽曲部分の接続，である．プロトタイプシステム「パピプーーン」の操作も，基本的にはこの手順に従う (図 10)．前処理の段階では，ユーザはまず SMF の課題曲にタイムスパン簡約分析を行い，専用ツール TS-Editor を用いて対応するタイムスパン木の情報を入力する (4.1 節)．オンライン処理の段階では，ユーザは主システム Summarizer を用いて要約を行う．まず Summarizer は楽曲部分のタイムスパン木の *lub* を計算し，類似した楽曲部分を探し出し，それらをまとめてウィンドウ上に表示する (4.2 節)．ポリフォニーの類似度判定に関してユーザは幾つかのパラメータを与える．その内の 3 つは R_N, R_A, R_T の閾値として用いられる．ユーザは，Summarizer の GUI を用いてこれらパラメータを繰り返し調節し，類似した楽曲部分を確定していく (4.3 節)．

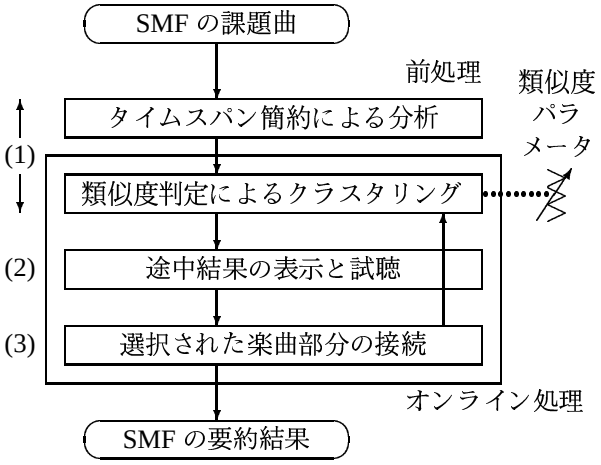

Figure 10: パピプーーンのシステム動作

ユーザが要約に際し削除すべき楽曲部分を同定すると，Summarizer はそれらを削除した残り部分を接続し，ユーザに試聴させる．その (途中) 結果がユー

ザの好みに合致すると要約は終了する．もしそうでない場合，類似度判定のパラメータを調節する段階に戻る．TS-Editor と Summarizer はともに Java で実装されている．

我々のシステム設計の方針は，音楽理論に裏打ちされている処理は自動化し，そうでない処理は手動 (ユーザとのインタラクション) で動作させるというものである．パピプーンの場合，ポリフォニーの表現法は GTTM が裏打ちしているが，要約の方法 (類似度の尺度の使い分けや要約として残す楽曲部分の選択等) はどんな音楽理論も裏打ちしていない．よって，類似した楽曲部分をまとめる処理は自動化し，それ以外の処理は手動とする．手動による処理の一部に対して，種々のヒューリスティクスを適用することも可能であるが，その結果得られた要約が常にユーザの好みに合うわけではないし，むしろユーザに不満を抱かせる場合もあるだろう．

4.1 TS-Editor による前処理

ユーザは，専用ツール TS-Editor を用いて課題曲のタイムスパン木を入力する．TS-Editor への入力は課題曲の SMF ファイルであり，出力は，入力の SMF ファイルを XML に変換したものと対応するタイムスパン木の情報 (XML ファイル) である．図 11 は動作中の TS-Editor のウィンドウであり，ピアノロール形式で表示された楽曲とタイムスパン木が見える (楽曲は “トルコ行進曲” の冒頭 7 小節である)．タイムス

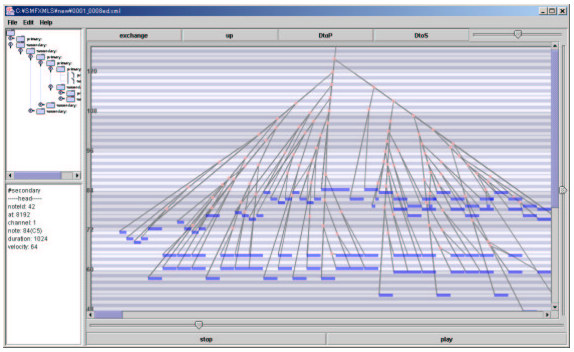

Figure 11: TS-Editor 上のピアノロール形式のポリフォニーとそのタイムスパン木

パン木の中間ノードの選択を容易にするために，上左隅のサブウィンドウには，同じタイムスパン木が Windows フォルダ形式で表示されている．入力の操作は大きく 2 種類に分けられる．1 つはタイムスパン木の入力であり，もう 1 つは時間構造の入力である．

まず TS-Editor が SMF ファイルを読み込むと，すべての音を発音時刻順にソートし，デフォールトのタイムスパン木 (次に発音時刻が早い音が左 secondary 枝となる) が付与される (図 12a)．図中，長方形がピアノロール形式における 1 つの音を表す．例えば今，ユーザは図 12a のタイムスパン木から図 12b のようなタイムスパン木を得たいとしよう．すると，ユーザは図 12a の矢印 ↘ で指示されているノードを選択し

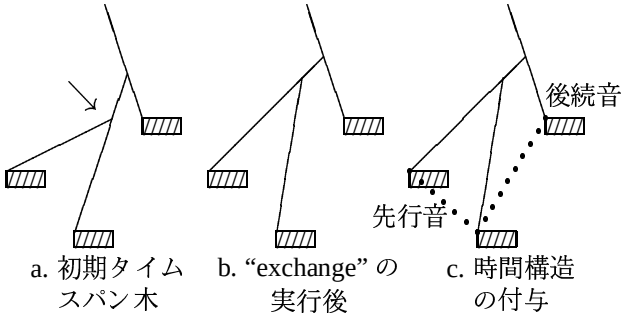

Figure 12: タイムスパン木の編集と時間構造の付与

“exchange” コマンドを発行する．TS-Editor は，タイムスパン木を編集するために，exchange を含む 4 つのコマンドを提供している．また，2 つ以上の音を 1 つの和音にグルーピングするコマンドも提供している．タイムスパン木を入力すると，次にユーザは時間構造の情報を入力する．図 12c は，今着目している中央の音がその先行音と後続音に挟まれている様子を示している．このような時間関係を表現するために，ユーザは中央の音から先行音と後続音の各々に点線を引く．

図 13 は，図 11 と同じ楽曲部分に対して，全ての音に時間構造を付与し終わった所を示している[2]．TS-Editor はタイムスパン木と時間構造を入力する効

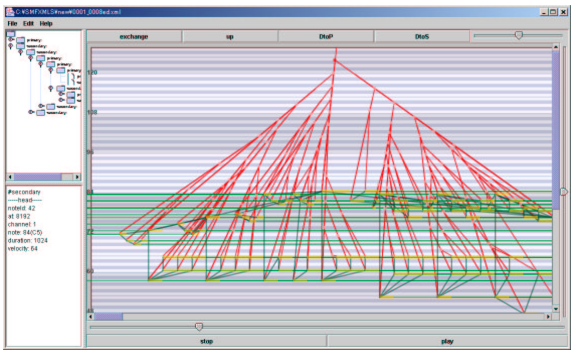

Figure 13: タイムスパン木と時間構造が付与されたポリフォニー

率を大幅に改善した．1 小節あたり平均 12.8 個の音符が含まれるトルコ行進曲を TS-Editor を用いて入力する場合，平均 3 時間で 4 小節分が入力できた．

4.2 類似した楽曲部分のクラスタリング

図 14 は，Summarizer が TS-Editor の出力ファイルを読み込んだ直後の，つまり Summarizer 起動直後のウィンドウである．TS-Editor 同様に，トルコ行進曲の楽譜全体がピアノロール形式で表示されている．楽譜の標準的な記法に従えば，楽曲の構成は AAB や ABA のように表現される．Summarizer の GUI はそ

[2] 研究報告の紙面上では残念ながらカラーで示すことができないが，実際の TS-Editor ウィンドウ上では，タイムスパン木の枝は赤で表し，時間構造の点線は緑で表し区別を付けている．

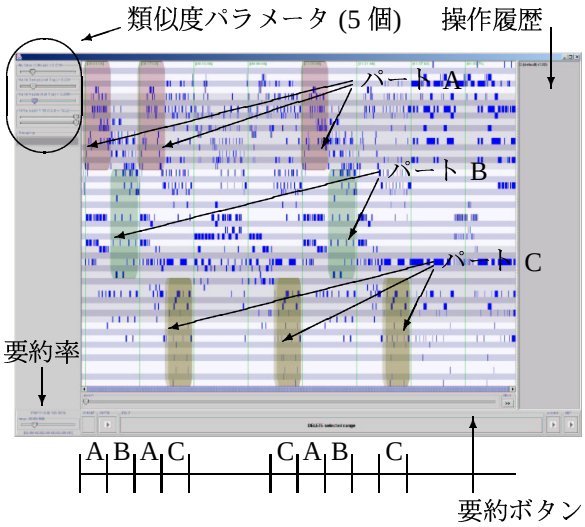

Figure 14: 起動直後の Summarizer の GUI 画面

の代わり，ピアノロール譜面上の色付きの短冊で類似楽曲部分を示す．上左隅には，類似度判定のパラメータを調節する 5 つのスライダがある．上から 3 つのパラメータは，各々 R_N, R_T, R_A の閾値であり，以下 T_N, T_T, T_A と呼ぶ ($T_\$$ = 0.0〜1.0 ただし \$ = N, T, A)．その次の 2 つのパラメータは，類似度判定を行なう楽曲部分のサイズ S とそのサイズのマージン M である．Summarizer 内部において，類似度判定の対象となる楽曲の各部分はタイムスパン木全体の中の対応する部分木で表現されている．

今，楽曲の部分 P と Q が与えられたとする．Summarizer は，$R_N(P,Q) > T_N \wedge R_T(P,Q) > T_T \wedge R_A(P,Q) > T_A$ の時に P と Q は類似していると判定する．楽曲部分のサイズ S は 2, 4, 8, 16 拍と変化させることができ，マージン M は 0% から 20% まで変化させることができる．例えば，S が 4 拍で M が 10% の時，類似度判定の対象となる楽曲部分のサイズは 3.6 拍 (4 × 0.9) から 4.4 拍 (4 × 1.1) の間に限られる．ここで 1 拍の長さは，入力 SMF ファイルの Tempo メタイベントから得られる．

類似度判定のためのパラメータが設定されると，Summarizer は，S と M の条件を満たす全てのタイムスパン部分木どうしの *lub* を計算し，その結果に基づき類似度判定を行う．もし P と Q が類似し，Q と R も類似している場合は，P，Q，R 全てが類似していると推論し，1 つのクラスタとする (例えばパート A という識別子と GUI 上の赤の短冊を与える)．図 14 において，各短冊の横幅は対応するタイムスパン木の時間幅を意味しており，12.8 拍 (16 × 0.8) から 19.2 拍 (16 × 1.2) である．一方，短冊の縦幅には特に音楽的な意味は無く，図 14 では 3 つのクラスタが発見されたので，表示を見易くするためにピアノロール譜面の縦幅を 3 等分しただけである．短冊が置かれていない区間は，そこに類似した部分が楽曲の他の区間で発見されなかったことを意味している．図中，参考のため，GUI ウィンドウの下にはクラスタリングの結果を標準的な記法で示した (A B A C ···)．

4.3 ユーザとのインタラクション

オンライン処理では，Summarizer の GUI を用いてユーザとのインタラクションを行う．図 14 の状態から始まり，図 15 に従って続く．ステップ 4 でユーザ

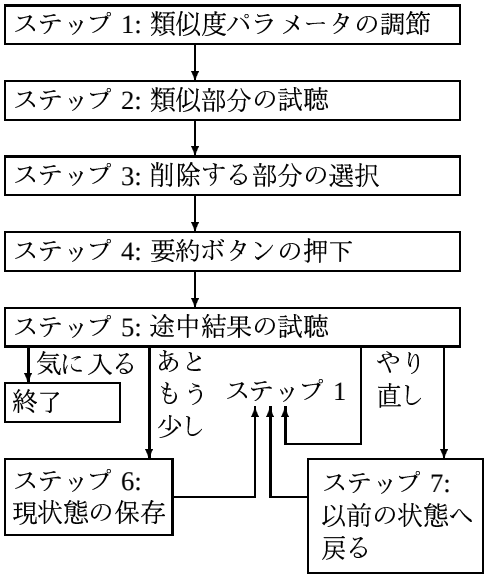

Figure 15: ユーザとのインタラクションによるオンライン処理

が要約ボタンを押下すると，パピプーンはステップ 3 で選択された楽曲部分を削除し，残った部分を接続し，要約率の値を更新する．この時，削除した部分を表示するために縦の黒い帯が現れる (図 16)．図

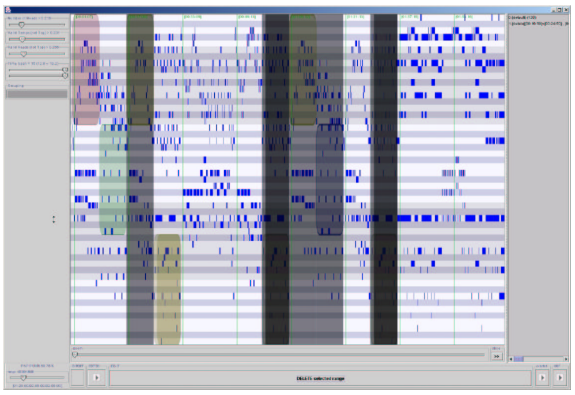

Figure 16: ある楽曲部分を削除した状態

中，削除されているのは，パート A の 2 番目と 3 番目，パート B の 2 番目，パート C の 2 番目と 3 番目であり，要約率は 68.75% である．

ステップ 5 でユーザは中間結果を試聴し，再生速度を 0.5〜2.0 倍に変化させて要約率の微調整を行うことができる (再生速度 1.0 が標準)．ステップ 6 では，後戻り処理のために，その時の要約の状態と類似度判定のパラメータを保存することができる．すると，その時点での状態が操作履歴のサブウィンド

ウの最下行に追加される．ユーザは途中要約を SMF ファイルとして残すこともできる．

さらに図 16 の状態から，ユーザがサイズ S を 16, 4, 2 と変化させて要約の処理を続けたとする．すると例えば，図 17 のような状態に到達することができる (要約率は 41.31%)．ここで操作履歴サブウィンドウの行数が増えていることに注意されたい．

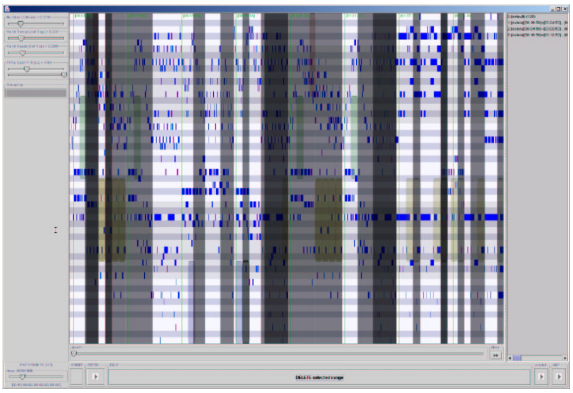

Figure 17: さらに要約を進めた状態

5 まとめ

現在のところ，被験者を使った正式な聴取実験はまだ実施していないものの，パピプーンを試用した数名からは好意的な感想を得ている．しかし，要約結果の質はシステム操作 (パラメータの調整や削除部分の指定) に関するユーザの熟達度や音楽センスに負う所が大であり，パピプーンの適切な評価は今後の課題であろう．

第 1 章で触れたように，音楽要約の中心課題には (a) ポリフォニーの表現，(b) 旋律の類似度判定を用いた楽曲構造の発見，(c) 削除すべき楽曲部分の同定，(d) 要約として残すべき部分の接続，があった．この内 (a) に関しては，他の音楽理論を本楽曲表現法の枠組に取り込むことで，表現の精度と効率を改善することができよう．(b) のパラメータ調節は，現在，ユーザとのインタラクションに基づき手動で行っているが，この調節の自動化は今度の課題である．(c) に関して，現在の GUI では，ユーザは削除する部分しか指定できないが，残す部分も指定できる方が自然であろう．削除する部分，残す部分を同定する処理の自動化についても今度の課題である．(d) に関して，現在の Summarizer は残った楽曲部分を単に接続するだけなので，auftakt 等により接続部分が不自然に聴こえる個所がある．よって，音楽的に自然な接続法が必要であろう．

上述したように TS-Editor は課題曲の入力の手間を大幅に改善しているが，さらなる改善も必要であろう．タイムスパン木に対する機械的あるいは定型的な操作 (列) を支援することが考えられる．さらに，同様の改善は Summarizer の GUI についても期待される．

最後に，パピプーンが採用している要約法以外の方法も考えられる．例えば，タイムスパン簡約により音の数を減らした旋律を作成し，その再生速度を上げる (速送り，fast-forward) という方法である．このような手法をパピプーンに組込むことも今後の課題であろう．

謝辞: 課題曲のデータ入力に関して，大館孝樹氏 (デジタル・アート・クリエーション) の多大なる協力を得ました．パピプーン開発の初期段階における青柳龍也助教授 (津田塾大学) との議論は大変有意義なものでした．

参考文献

[1] Carpenter, B., The Logic of Typed Feature Structures, Cambridge University Press (1992).

[2] Hirata, K., and Aoyagi, T., Musically Intelligent Agent for Composition and Interactive Performance, In Proceedings of ICMC 1999, pp.167-170.

[3] 平田, 青柳, バービーブン: 音符レベルでユーザ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 音楽情報科学研究会 研究報告 2000-MUS-37, pp.17–23 (2000).

[4] 平田, 青柳, パーピープン: ジャズ和音を生成する創作支援ツール, 情報処理学会論文誌, Vol.42, No.3 (2001).

[5] 平田, 青柳. 音楽理論 GTTM に基づく多声音楽の表現手法と基本演算. 情報処理学会論文誌, Vol.43, No.2, pp.277-286 (2002).

[6] Hirata, K., and Hiraga, R., Next Generation Performance Rendering - Exploiting Controllability, In Proceedings ICMC 2000, pp.360-363.

[7] 平田, 平賀, GTTM に基づく音楽表現手法再考, 情報処理学会 音楽情報科学研究会 研究報告 2002-MUS-45, pp.1–7 (2002).

[8] Huron, D., Perceptual and Cognitive Applications in Music Information Retrieval, In Proceedings of ISMIR 2000.

[9] Kifer, M., Lausen, G., and Wu, J., Logical Foundations of Object-Oriented and Frame-Based Languages, Journal of ACM 42, 3 (1995).

[10] Lerdahl, F., and Jackendoff, R., Generative Theory of Tonal Music, The MIT Press (1983).

[11] Logan, B., and Chu, S., Music Summarization using Key Phrases, In Proceedings of ICASSP 2000.

[12] Plaza, E., Cases as terms: A feature term approach to the structured representation of cases, Lecture Notes in Artificial Intelligence Vol.1000, pp.265-276, Springer-Verlag (1995).

[13] Selfridge-Field, E., Conceptual and Representational Issues in Melodic Comparison, Computing in Musicology 11, pp.3-64 (1998).

[14] Yokota, K., Towards an Integrated Knowledge-Base Management System: Overview of R&D on Databases and Knowledge Bases in the FGCS Project, In Proceedings of International Conference on Fifth Generation Computer Systems 1992, Institute for New Generation Computer Technology, pp.89-112.