



第5回

計算論的音楽理論の応用

浜中雅俊 (筑波大学システム情報工学研究科)

平田圭二 (NTT コミュニケーション科学基礎研究所)

東条 敏 (北陸先端科学技術大学院大学)

音楽の意味

これまでの連載では、音楽に含まれる構造にはどのようなものがあるか、そしてそれらの構造を抽出するにはどのようにすればよいかについて情報处理的な視点から述べてきました。

音楽に含まれる構造とは、楽曲のまとまり感を表す階層的なグループ構造や簡約構造などのことです。具体的な例を考えてみましょう。ここに「ドレミドレミ」というメロディがあれば、「ドレミ」が2回繰り返されていると分かります。「ドレミ」で緊張が高まった後に「ファソラ」と続けば一般に、より緊張感が高まりますが、もし後に「ミレド」と続けば前半の緊張が弛緩するでしょう。繰り返しがあるということや、緊張が高まる部分と弛緩する部分があるということは、メロディのどこかに区切り・境界が存在するということです。すると、区切り・境界の強弱が階層的なグループ構造を生み出し、その階層に沿ってグループ中でより重要な音を残すことで簡約構造を抽出することができます。

音楽家に限らず音楽を作ったり聞いたりしている人々は、この音楽理論で規定されているような音楽的な構造を使って何らかの情報を表現したり伝えたりしています。では、どんな情報をやりとりしているのでしょうか？記号論が教えるところのコードを参照しながらメッセージを作るという枠組みにあてはめれば、音楽理論の規則がコードに相当し、楽曲がメッセージに相当するでしょう。このメッセージとしての楽曲が表現しているものを「音楽の意味」と呼びましょう^{☆1}。

音楽的な構造を使って何らかの情報を伝えることの1つに作曲と呼ばれる活動があります。人は、音楽的な技術や知識を駆使して湧いてくるアイデアを楽曲として表現する一方、制作途中の楽曲を聞き返して（鑑賞、

振り返り）アイデアを練り直したり修正したりします。このような活動を支援するあるいは代替する作曲ツールや作曲アルゴリズムの研究は、人の意図（まさに「音楽の意味」）という要素が入ってくるので、システム（ツールやアルゴリズム）を改良するための評価を実現するのは非常に難しくなります。

一般に、作り出された楽曲や演奏の善し悪し（あるいは好き嫌い）ではシステムの評価はできません。システムがどんなに素晴らしくても、入力されたデータやそれを使う人が不慣れですと、質の高い楽曲や演奏を作り出すことは難しいでしょう。たとえば、楽譜エディタのように五線譜の上に音符や音楽記号を1つ1つ乗せていくソフトがあれば、どんな名曲でも駄作でも思いのままに記述することができます。作品の出来は楽譜エディタを使う人次第です。

この問題の本質は、ユーザの意図をいかに表現し、システムに伝え、システムがそれを理解するかではないかと思います。ここでもし、グルーピング構造やタイムスパン簡約構造など、音楽の意味に対応する高次の音楽構造が操作できるシステムがあったとしたらどうなるでしょうか。システムは、そのような高次の音楽構造を通じてユーザの意図を知ることができるでしょう。初心者でも自分の意図を伝えやすくなる可能性があると思います。

筆者らは、これまで本連載で述べてきた計算論的音楽理論が「音楽の意味」を扱う武器になり得るのではないかと考えています。そこで最終回となる今回は、いくつかの基本的な音楽タスクを取り上げて、音楽の構造を操作する方法論を紹介してみたいと思います。

作曲や演奏といった高度な活動は、そのような基本的な音楽タスクが複雑に組み合わさったものですから、計算論的音楽理論はいつか（少し大袈裟に言えば）音楽の意味を創出し理解する方法論へと発展していくのではないかと期待しています。

☆1 音符の出現頻度やメロディの変化度からエントロピーを算出するような情報量的な「意味」は本稿では扱いません。また、楽しい、重々しいなど心理的な「意味」については音楽心理学という分野で研究されており、やはり本稿では扱いません。

楽曲分析システムの完全自動化

ではまず音楽タスクの話に入る前に、前回（第4

回)の最後に予告した計算機上で実行可能な音楽理論 exGTTM の全自動化について述べておきましょう¹⁾。exGTTM とは、音楽理論 GTTM²⁾では暗黙的だったり曖昧だった概念やパラメータを明示化したものです。exGTTM に正しい分析結果を出力させるためには、正しい分析結果(出力させたい音楽構造)を知っている人が、明示化されたパラメータ群(全部で46個)の値を上手く調整しなければなりません。パラメータ群の値調整はそれ自体非常に骨の折れる仕事なので自動化されることが強く望まれますが、そもそも人が正しい分析結果を知っていないとパラメータ群の値調整ができないというのは、さまざまな応用を考えると大きな不都合をもたらすことでしょう。実際に、この exGTTM 全自動化は、後に紹介する楽曲の類似性判定、メロディ予測、メロディ生成といった音楽タスクを実現するにあたって、必要不可欠な段階です。

exGTTM 全自動化の鍵はタイムスパン木の安定性です。安定性については後ほど詳述しますが、「正しい分析結果=最も安定したタイムスパン木」という仮定を置き、exGTTM が最も安定したタイムスパン木を出力するようパラメータ群の値を調整していくということで全自動化を達成しました。

【拍節構造の安定性とタイムスパン木の安定性】

たとえば、W. A. Mozart の K.331 第三楽章(いわゆるトルコ行進曲)の冒頭の旋律を分析してみると、2通りの解釈が考えられます(図-1)。図中、五線譜の上にある木構造はタイムスパン木と呼ばれる二分木で、隣接する2つの音符のうち、メロディを構成する音として重要な方を幹とし、そうでない方を枝とします。五線譜の下にある複数の点は拍節構造(リズム)、つまり時間的に等間隔であることと、どの拍にどのような強弱が付いているかということを表しています。強い拍ほど縦に多くの点(拍点)が並んでいます。さらにその下にはグルーピング構造が書かれています。メロディの区切り個所に挟まれた「」で示された部分が1つのグループを表しています。

ここで解釈(a)と(b)を比較してみましょう。グルーピング構造や拍節構造には違いがありませんが、タイムスパン木には少し違いがあることが分かります^{☆2}。解釈(a)では、拍節構造の拍点の多い音符がタイムスパン木の幹に対応し、拍点の少ない音符が枝に対応しています。ここで GTTM のタイムスパン木と拍節構造の安定性に関するルール TSRPR5 を紹介します。

☆2 違いが分かりやすいよう、第1音の枝を淡い色にしました。

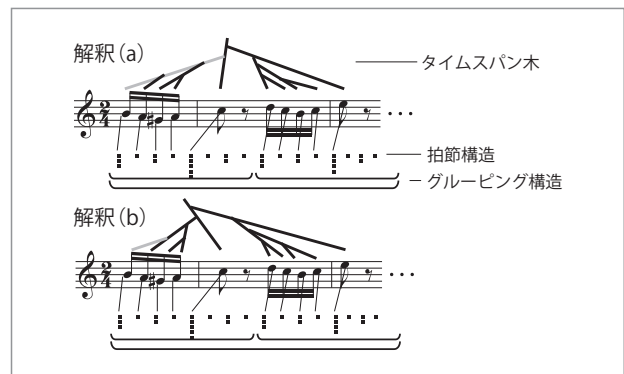


図-1 楽曲の2つの解釈

TSRPR5 : In choosing the head of time-spans T , prefer a choice that results in more stable choice of metrical structure.

(タイムスパン木のヘッド(幹)を選択するとき、より安定した拍節構造を得ることになる選択が好ましい)

ここは少し説明が必要です。等間隔で和音に変化する点や線型に変化するベースラインは音楽的に重要なので、一般に、そのような点に現れる音符にはタイムスパン木の幹が対応します。さらにそのような音符によっても「拍節」が生み出されるので、それらがリズムによる拍節構造(拍点の多い音符)と対応していることが望ましいと GTTM は主張します。安定した拍節構造というのは、タイムスパン木と良い対応関係にある拍節構造のことを意味しています。解釈(a)はこの TSRPR5 を満たしていると言えます。

一方、解釈(b)では、冒頭の2音を見ていただくとすぐ分かると思いますが、拍点の少ない方の音符がタイムスパン木の幹になっています。ここで GTTM に含まれる安定なタイムスパン木に関するルール GPR7 を紹介します。

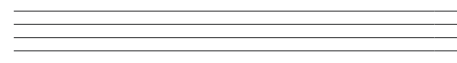
GPR7 : Prefer a grouping structure that results in more stable time-span and/or prolongational reductions.

(タイムスパン木やプロロンゲーション木がより安定するようなグルーピング構造が好ましい)

タイムスパン木の安定性については Tonal Pitch Space の節で詳しく触れますが、解釈(b)のタイムスパン木は(a)のそれより安定しており、解釈(b)はこの GPR7 を満たしていると言えます。

TSRPR5 も GPR7 も、後段のタイムスパン簡約の結果が前段のグルーピング構造分析や拍節構造分析に反映されるという意味で相互依存性を持っています。

このようにどのルールを優先するかでタイムスパン木が変わり解釈が変わってきます。前回も述べました



が、我々は GTTM を規範的に見る立場をとっています。GTTM は最終的な分析結果としてより安定したタイムスパン木の方が望ましい、つまり音楽的に正しいと主張しており、それゆえ「正しい分析結果＝最も安定したタイムスパン木」という仮定を置きました。したがって、拍節構造を反映して構成された安定したタイムスパン木 (a) より、タイムスパン木自体の安定性を優先した (b) の方が正しい解釈と考えられます。

【 Tonal Pitch Space 】

タイムスパン木の安定性の定義は、Tonal Pitch Space (TPS) という理論が与えています³⁾。一般に音どうしの間には、音階の中で近い音 (C と C# など) と調の機能において近い音 (C と G など) があります。するとこれら異なる尺度を組み合わせると音どうしの間に距離を定義することができます。この距離を pitch-space 上の距離と言います。同様に、和音間に対しても距離が定義でき、それは音間距離、重複する音の数、和音内での機能に対する重みづけ (第三音より第五音、第五音より根音への重みづけなど) に基づきます。さらには、トニック (主和音) どうしの距離をもって調間の距離も定義することができます。

あるピッチイベント (音や和音) が次のピッチイベントへ進行する場合を考えましょう。TPS は、「ピッチイベント間距離は pitch-space 上、最小限の値を持つよう計算されるべきである」という原則を設けています。すなわち、余計な和声の変化、転調などは考えないようにすべきだという考え方です。Tonal Pitch Space の著者 Fred Lerdahl はこの原則を物理現象の状態変位になぞらえ、時間、距離を最小にする経路で変位が起きるようなものとして説明しています (文献 3), p.74)。

より安定なタイムスパン木を構成する話に戻しましょう。もしある時間幅を持つメロディ部分が「局所的に調を構成している」と仮定できるとき、そのヘッド候補として複数のピッチイベントがあるとしましょう。このとき、余計な和声変化、転調をしないものほどより安定なヘッドであり望ましいと考えます。すなわちヘッドの選択では、もし局所的なトニック (主和音) があればそれを選び、もしなければ pitch-space 的にトニックからの距離がより小さいものを選びます。

このタイムスパン木の安定性に関するルールは、ある 1 つのヘッドの選択に関する制約を記述しているのですが、このルールはタイムスパン木の全ノードに対して同時に適用されます。そして、適用した結果、タイムスパン木全体として、各ノードにおける幹と枝の距離が最小になることが好ましいと言っているのです。各ノードにおける幹と枝の距離をタイムスパン木全体で集計しなけ

ればなりませんが、その点は曖昧なままです。

本稿では GPR7、TSRPR5 の定式化と実装についての詳細は省略しますが、ここでも前回ご紹介したように、ルール GPR7 と TSRPR5 が満たされる程度を各々 D_{GPR7} と D_{TSRPR5} というパラメータとして明示化しました¹⁾。こうして GPR7、TSRPR5 の定式化が済めば、あとは exGTTM が最も安定したタイムスパン木を出力するよう、つまり D_{GPR7} と D_{TSRPR5} の平均を最大化するようパラメータ群の値を調整することで全自動化を実現しました。GPR7 と TSRPR5 は相互依存的で少し厄介ですが、パラメータ値の調整を最適化問題の 1 つだと考えれば、今のシステムで採用している山登り法以外にもさまざまな解決法が考えられるでしょう。

メロディの類似性判定

音楽理論 exGTTM の全自動化により、音楽的に正しいタイムスパン木が得られるようになりました^{☆3}。これで、音楽の構造や意味をある程度計算機上で表現し操作できます。

まず最初に取り上げる音楽タスクはメロディの類似性判定です⁴⁾。2 つのメロディは似ているのかいないのか、どこが似ていてどこが似ていないのか。類似性の判定は、音楽に限らず、一般のメディア処理において最も基本的なタスクの 1 つと言えるでしょう。

メロディの類似性判定と聞いてすぐ思い起こされることに、平成 10 年から平成 16 年にかけて作曲家の小林 亜星氏と服部克久氏の間で争われたいわゆる「どこまでも行こう vs 記念樹裁判」があるでしょう⁵⁾。この裁判では、「どこまでも行こう」と「記念樹」という 2 曲のメロディ、和声、拍子・リズム等の構造が細かく取り上げられ、分析、比較され、同一性や類似性が議論されました。その判決文を読むと、まず両メロディの表層的な情報、たとえば含まれる音の個数、音高の一致する音、各音の長さ等を比較していますが、それだけではなく音楽の構造に関する言及も多数散見されます。たとえば、「全体のメロディは複数のフレーズ (グループ構造) からなる」、「繰り返し構造」、「導音^{☆4}を除いて抽象化すると同一」、「強拍に乗っている音が共通」、「旋律が山型を形成している (これは緊張—弛緩構造のことです)」等々。つまり、2 曲の類似性を客観的に判定するには、

☆3 まだ、最も安定したタイムスパン木が出力できない場合や最も安定したタイムスパン木のはずなのに正しい分析結果に思えないような場合が残っていますが、これらは今後の研究課題です。

☆4 半音上行して主音を導く音、つまり長音階、短音階の第 7 度音のこと。

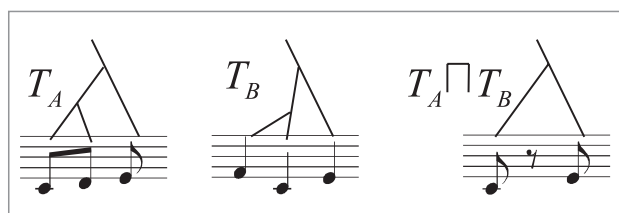


図-2 2つのタイムスパン木の *meet* 演算の例

そのような音楽の構造にまで踏み込む必要があったということでしょう。

本連載の読者の皆様ならもうお気づきのことと思いますが、ここに出てきた音楽用語はほとんど本連載ではお馴染みのものばかりです。そして、我々の計算論的音楽理論の枠組みでは、それらはグルーピング構造、拍節構造、タイムスパン木の構成要素として自然に取り込まれ、計算機の操作対象となっています。

タイムスパン木を利用した楽曲の類似性判定する方法を紹介します。まず筆者らは、タイムスパン木をデータ項 (term) として表現し、タイムスパン木間の包摂関係 (\sqsubseteq , 半順序関係) を機械的に計算する方式を提案しました⁶⁾。つまり、あるタイムスパン木 P をタイムスパン簡約して Q が得られるとき、 $Q \sqsubseteq P$ が成り立つようにデータ表現と包摂関係を定義しました。すると楽曲を元とするような束とその上の二項演算 *meet* (\sqcap , 交わり) と *join* (\sqcup , 結び) が定義できます。

図-2は演算の適用例です。図はメロディ「ドレミ」のタイムスパン木 T_A と、「ファドミ」のタイムスパン木 T_B の *meet* を計算し、右端の新しいメロディ「ドミ」と $T_A \sqcap T_B$ が得られる様子を示しています。この *meet* 演算が、2つのメロディの共通部分を抜き出したことが分かります。もし同じメロディ P どちらの *meet* なら答えも P になりますが、まったく異なるメロディの *meet* なら答えは \perp (空)、あるいはそれに近い項になるでしょう。たとえば、前掲判決文中の「導音を除いて抽象化すると同一」というのは、まさしく $\text{meet}(P, Q)$ によって表現される分析のことです。

このことから、2つのメロディ P, Q の $\text{meet}(P, Q)$ を計算して、 P あるいは Q から失われた情報の量を測ると P と Q の間の類似度 $R(P, Q)$ を表せるのではないかと考えられます⁷⁾。具体的には、

$$R(P, Q) = \frac{|\text{meet}(P, Q)|}{\max(|P|, |Q|)}$$

と定義すればよいでしょう。ここで $|P|$ は項 P の持っている情報量あるいは複雑さであり、たとえば、 P に含まれる音の個数 (木構造として葉の枚数)、木構造としての枝の本数、含まれる全音符の属性の総数などを尺度として用いることができます。たとえば、前掲判決文中

には「甲曲と乙曲は全体の約72%が同一音であり」という記述が出てくるのですが、これは尺度として音の個数を採用した場合に相当します。またこのときは、甲曲と乙曲のメロディ中の同一音を適切に発見し対応付けた上でその個数を数えなければなりません、 $\text{meet}(P, Q)$ の計算はそれを機械的に実現してしまいます。

記号的ではなく、楽曲の音楽音響信号を分析して類似性判定を行う手法もいくつか提案されています。これらは、メロディではなく音響信号としての楽曲の類似性を判定するもので、音響的特徴量の時間変化を比較します。たとえば文献8)の手法では、楽曲信号を短い区間に分割し、そのSTFTパワースペクトルを求め、12次元クロマベクトルを抽出し、そのクロマベクトル間の類似度を用いて楽曲信号の類似度を算出しています。

メロディ予測

次に取り上げる音楽タスクはメロディ予測です。これまで演奏されたメロディや音符列から次にどんな音が演奏されるか予測したり、演奏されるのが好ましい音の候補を挙げるというものです。

メロディの予測が一筋縄で行きそうにないことは、直感的にお分かりいただけると思います。通常、作曲家や演奏家は楽曲が単調にならないようにさまざまな工夫を施します。メロディを聴取する人の期待や予想を裏切ったり、引き延ばしたり、時には満足させたりして、楽曲に緊張と弛緩を与えます。つまり、簡単には予測できないような変化が付いているわけで、そのような変化を機械が予測するというのは、かなり難しい課題に思えます。また、演奏されるのが好ましい音の候補を挙げるという観点からは、付加する変化の程度を(ユーザの意図に従って)制御する必要があります。これもかなり難しい課題に思えます。

従来のメロディ予測手法の多くはコーパスデータの統計的学習手法を利用するものです。まずコーパスデータを何らかの方法で学習し楽曲の特徴を抽出しておきます。その特徴を頼りに今演奏されているメロディに類似したメロディをコーパスの中から探し出し、出現頻度のデータや事例から次の音を推論します⁹⁾。あるいは、あらかじめコーパスデータを用いた学習でモデルを作っておき、今演奏されているメロディをモデルに入力してメロディ予測を行います^{10), 11)}。このようにコーパスデータを学習するということは、ある特定のスタイルやジャンルに閉じた範囲では適切なメロディ予測が可能です。そのスタイルやジャンルから外れたメロディが入力されたときの予測まで保証するのは難しいでしょう。

そもそもコーパスデータを効果的に学習するために

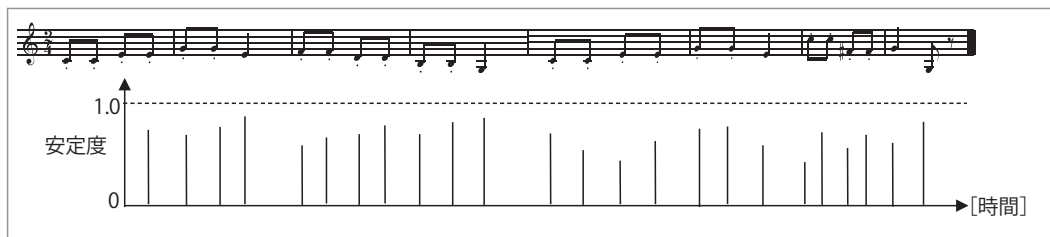


図-3 メロディとその安定度

は、音楽的に近いメロディが、項として、あるいはデータとしても近くなるように表現されていることが重要で、単に楽譜上の音高や時刻といった表層的な近さだけでは、音楽的な近さを十分にカバーすることができず、効果的な学習も望めません。そこで、メロディの類似性判定の章で出てきた $R(P, Q)$ を近さの尺度として使うことが考えられます。そうすれば、特定のスタイルやジャンルによらない学習が可能となるでしょう^{☆5}。

【タイムスパン木の安定度に基づくメロディ予測法】

さらに、exGTTM から得られるタイムスパン木をもっと直接的に利用したメロディ予測法も考えられます。今まで演奏したメロディの音列 S に、その次に弾く音の候補 n を加えて新しい音列 $S+n$ を作り、 $S+n$ のタイムスパン木とその安定度を計算することで、 n の妥当性を見積もることができます。ただし、上でも触れたように、メロディには通常、簡単には予測できないような変化が付いているので、安定なタイムスパン木をもたらし n が常に好ましいとは言いきれません。ここで実際のメロディ例において、安定度が実際にどのように変化しているのか見てみましょう (図-3)。楽曲はハイドンの Andante です。

図中、五線譜の下に音符ごとに算出した安定度が記してあります。この安定度は、exGTTM が最も安定したタイムスパン木を出力するために使われた指標であり、 D_{GPR7} と D_{TSRPR5} の平均のことです。その値域は 0～1 で、その値が大きいほどタイムスパン木は安定しています。GTTM のルールを適用するためには最低 4 音必要であるため、安定度は 4 音目以降から算出されています。メロディの調およびコード進行は Tonal Pitch Space に基づいて自動推定しています。図-3 をご覧いただければ分かるように、全体的に 2 小節ごとにフレーズが区切られていて、各フレーズ (5～6 小節目を除いて) の最初は期待を裏切るような音から始まり、トニック和音かドミナント和音の構成音で終了する (解決する) のを

反映して、安定度も低→高という傾向を示しています。7 小節目の「ドドファ#ファ#」は、この 8 小節の中で一番期待を裏切って緊張する部分であり、安定度も低くなっていますが、8 小節目で予測通り解決して安定度が高くなっています。前半 4 小節の「ドドミソソ…」の 2 音ずつ繰り返す部分では 1 音目より 2 音目の方が安定度が高くなっていて、つまり予測に合っていることが分かります^{☆6}。このように、実際のメロディでは、予測を裏切ったり期待を満足したりが繰り返され、それに合わせて安定度も低くなったり高くなったり変化し、安定度が常に最大になっているわけではないことがお分かりいただけると思います。

もし、安定度に基づくメロディ予測器を作ったとしたら、そこで予測されるメロディというのは、その時点までに演奏された音から得られる期待を素直に満足させるような音列になります。メロディの期待や予測を裏切るような音を正確に予測しようとするものではありません。そもそも、予測を裏切るように意図されている部分まで正確に予測しようとするのは、音楽的にあまり意味がないかもしれません。

【予測ピアノ】

筆者らは、タイムスパン木の安定度に基づくメロディ予測法を実際の音楽システムに応用してみました¹²⁾。即興演奏などを行っている演奏の初心者に、予測によって得られた後続音を提示することによって、演奏初心者の即興演奏を支援するシステムです (図-4)。「予測ピアノ」という名前を付けました。ピアノ天板に横 25 個、縦 32 個のフルカラー LED が配置されており、手元に近いほど近未来の予測音が、演奏者から遠くなるほど遠い未来の予測音が表示されます。予測音の候補が時間とともに縦スクロールしながら表示されるわけです。

予測音の計算は次のように行われます。次に弾かれる

☆5 GTTM が対象としている西洋調性音楽の範囲に限定されるものの、現在我々の周囲から聞こえてくる音楽の多くは該当しているでしょう。

☆6 ハ長調であるにもかかわらず、5 小節目の C の安定度が前の小節の G よりも低かったり、5～6 小節目の安定度が S 字カーブ状であったり、人の考える「安定度」と食い違っているように見える個所もあります。この主な原因として、人の考える安定度は「楽曲全体の中の『この音』の安定度」であるのに対し、exGTTM は「『この音』で曲が終わる」と仮定して安定度を計算していることが考えられます。

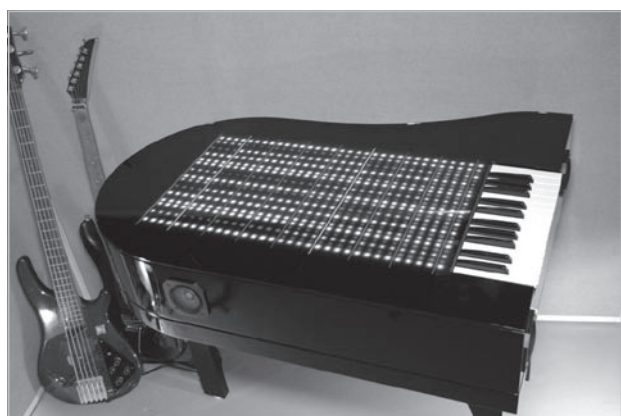


図-4 予測ピアノ

音候補の空間が音高方向に 25 点，時間方向に 32 点広がっているとします（25 は 2 オクターブ分の鍵盤に相当し，32 は 16 音符 2 小節分に相当します）。これまで弾かれた音列 S に対して，もし音候補の空間上のある 1 音 n を弾いたとして，新しい音列 $S+n$ のタイムスパン木の安定度をリアルタイムで計算します。音候補の空間には全部で 800 点（ 25×32 ）ありますのでその全点に関して安定度をリアルタイムで計算して，安定度の高い順に黄と赤で輝度を変えながら表示します（図-4）。

前に図-3 を観察して分かったように，常に安定度が最大の音が演奏されるわけではないので，後続の音候補としては，安定度の高い音を複数個黄色で表示するようにしました。予測ピアノの演奏者は，即興演奏中にどの音を弾こうか迷ったとき，期待を素直に満足する黄色い音を選んで弾くこともできますし，あえてそれ以外の予測を裏切る音を選ぶこともできます。

予測ピアノは，まだ筆者らが個人的に使用している段階です。比較的安定度の大きい音を選んで弾いていくと，一応，曲らしくなる音を弾くことができますが，無難であり面白みは感じられませんでした。他方，安定度の低い音を選んでもそれなりの曲に聞こえるときもあり，予測を裏切ったり，期待を満足したりすることをコントロールするのは大変難しいということを再認識しました。また，音楽初心者は，あまり余裕がないので手元を見ながら弾く傾向にあり，遠い未来の予測音まで気を配るのは難しそうでした。予測ピアノのユーザインタフェースにはまだまだ改良の余地があるようです。

メロディ生成

最後に取り上げる音楽タスクはメロディ生成です。exGTTM から得られるタイムスパン木を利用することで，音楽の構造や意味を計算機上で表現し操作できるようになったので，それを活かしたメロディ生成法を考え

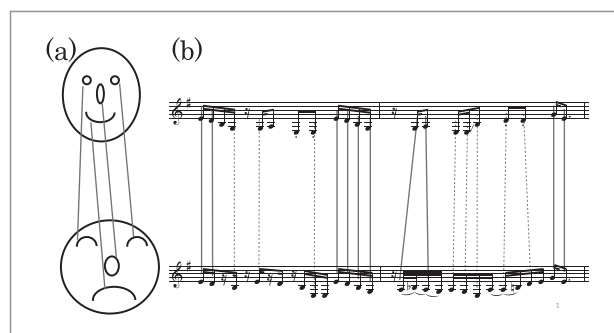


図-5 2つの画像とメロディの対応付けの例

てみたいと思います。

一般に，操作を容易にするために操作の抽象度を上げることと，細かいレベルで個々のユーザの意図を反映するために操作の抽象度を下げることがトレードオフの関係にあります。そのトレードオフ解消を目指す方法論の1つに，事例を用いて指示を出すものがあります。たとえばユーザがシステムに対して「メロディ A にメロディ B のニュアンスを付加せよ」と指示できれば，ユーザの意図を簡易かつ的確にシステムに伝達することができるでしょう^{13), 14)}。しかし，メロディ B のニュアンスだけを取り出し，それをメロディ A に適用するには，メロディ A やメロディ B の構造や意味を正しく理解していなければ実現は難しいのではないかと思います^{☆7}。そこでタイムスパン木の登場です。

このメロディ A にメロディ B のニュアンスを付加して少しずつメロディ B に近づけていく操作をここではモーフィングと呼びましょう。メロディ A とメロディ B の内挿を計算すると換言することもできます。では，一体どのようにすればメロディのモーフィングが実現できるのでしょうか。一般にモーフィングという処理は，2つの画像が与えられたときに，片方の画像からもう一方の画像へ滑らかに変化していくようにその間を補うための画像を作成することをいいます（図-5）。2次元の顔画像のモーフィングの場合，たとえば以下のような手順でモーフィング画像を生成することができます。（1）目や鼻など2つの画像の特徴点の対応付け，（2）各画像部品の形状，位置，色等の重み付け，（3）両画像の合成。この画像のモーフィング手順にならって，モーフィングされたメロディを作ってみましょう（図-6）¹⁶⁾。

(1) メロディの共通部分の対応付け：2つのメロディ A, B に対応するタイムスパン木 T_A, T_B が与えられると，類

☆7 対象がメロディではないのですが，楽曲 A に類似した楽曲 B を検索してきて，B の演奏表情付けを A に転写する SaxEx という音楽システムがあります¹⁵⁾。このシステムは，類似した楽曲の検索や演奏表情付けの適用の過程で，GTTM のタイムスパン木だけでなく，Narmour の IR モデル(本連載第2回を参照)も利用しています。



似性判定の章で紹介した *meet* によって $T_A \sqcap T_B$ としてその共通部分を求めます (図-6 上部)。共通部分を求めることで、メロディ A, B に含まれる共通の特徴的な音 (重要な音) が抽出されます。メロディの場合、画像と違って、対応付けを表現する項 $T_A \sqcap T_B$ が実際に得られるのが特徴です^{☆8}。

(2) 各メロディの部分簡約：メロディ A と B の共通部分を求めるということは、メロディの非共通部分を求めることと表裏一体であり、そのメロディの非共通部分には、それぞれ相手のメロディにはない特徴が表現されているはずです。この部分が相手のメロディに適用されるニュアンスに相当します。画像モーフィングにおける各画像部品の形状、位置、色等の重み付けに対応するのは、この非共通部分を増減させる処理です。*meet* の性質から $(T_A \sqcap T_B) \sqsubseteq T_A$ と $(T_A \sqcap T_B) \sqsubseteq T_B$ が成り立っていますので、非共通部分を増減させた項 T_A' は、 $(T_A \sqcap T_B) \sqsubseteq T_A' \sqsubseteq T_A$ のように表現できます (図-6 左側)。 T_B' も同様です (図-6 右側)。 T_A' , T_B' の具体的な計算方法は後述します。

(3) メロディの合成：最後のステップは、お互いの非共通部分を含む T_A' と T_B' の合成です。この合成は *join* 演

算により $T_A' \sqcup T_B'$ として得られます (図-6 下部)。合成する際、ステップ (1) で抽出された共通の特徴的な音どうしを対応させる必要がありますが、それは *join* 演算によって保証されます。ただし、タイムスパン木の形は同じだけれど音高だけが異なっているようなメロディどうしの *join* では、いずれかの音を含むすべての場合のメロディを出力するようにしました (図-6 にてメロディ C が複数出力されている)。

このように、あまりに機械的に計算できてしまうので、 T_A や T_B がメロディを表現したタイムスパン木であることを忘れてしまいそうになるくらいです。

【メロディの部分簡約の計算方法】

T_A を部分的に簡約した T_A' の具体的な計算方法を述べますが、その前にまずメロディの簡約について簡単に説明しておきたいと思います。図-7 は、メロディ D を簡約してメロディ E が得られ、さらに簡約してメロディ F が得られる例を示しています。メロディ D の五線譜の上にある木構造は、メロディ D をタイムスパン簡約した結果得られたタイムスパン木です。そして、タイムスパン木を横切る簡約レベル E がメロディ E に、簡約レベル F がメロディ F に対応しています。通常のタイムスパン木の簡約では、木の最も下位の部分木に対して一

☆8 強い画像において項 $T_A \sqcap T_B$ に対応するものを探すとしたら、それは平均顔のワイヤーフレームかもしれません。

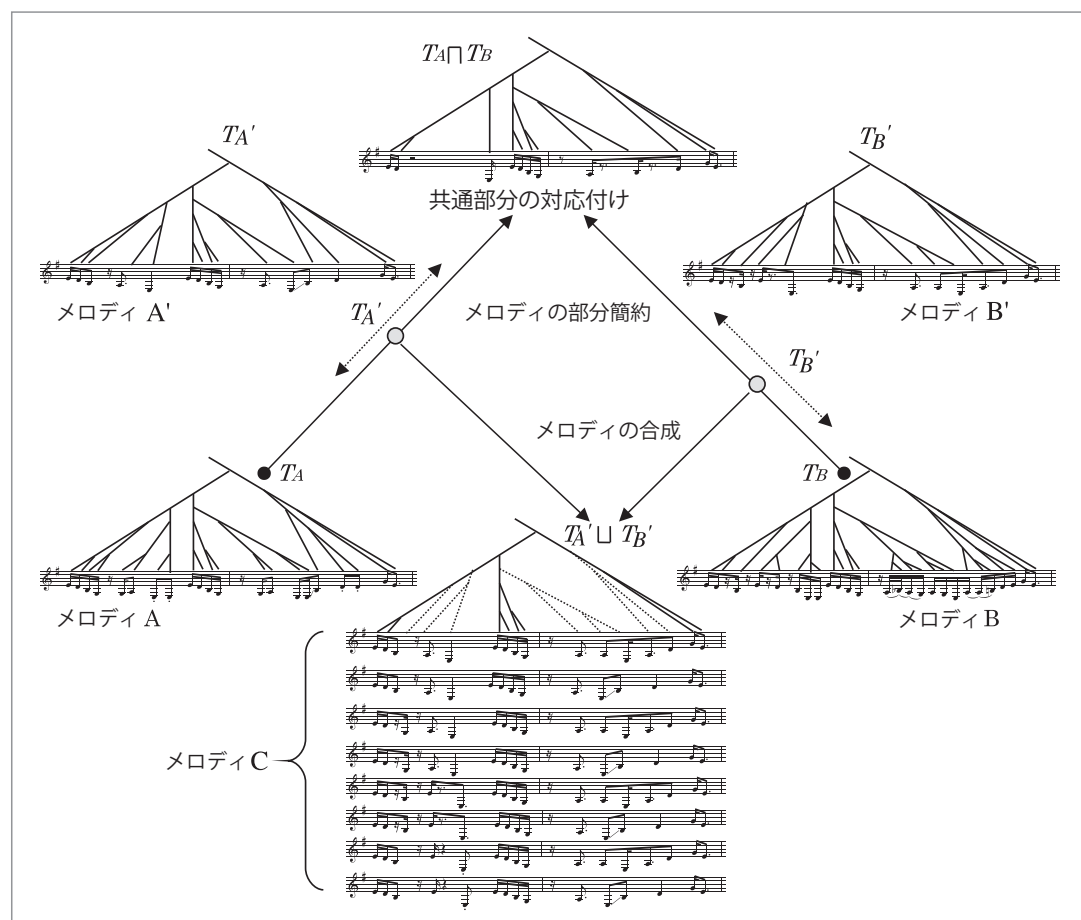


図-6 メロディモーフィングの概要

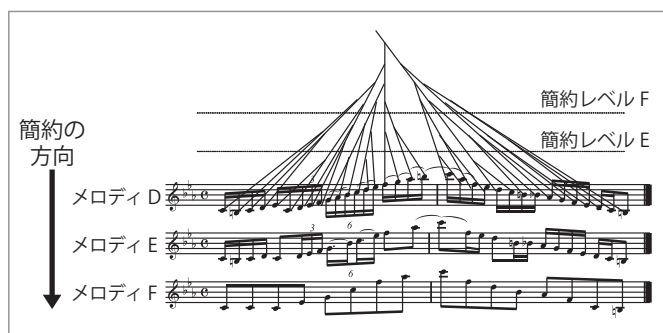


図-7 メロディの簡約

度に複数個の音符が捨象されます。図-7では、タイムスパン木Dの最も下位の構造に対して6個の音符が捨象されEが得られます。さらに次のレベルでは11個の音符が捨象されFが得られます。このとき $T_F \subseteq T_E \subseteq T_D$ という関係が成立しています。

メロディのモーフィングで用いられるメロディの部分簡約とは、音符を1つずつ捨象していく簡約法です(図-8)。捨象する音符は以下のようにして決定します。非共通部分のタイムスパン部分木に含まれる拍点の数が最小の音を選び、その音符を捨象します。拍点は GTTM の拍節構造分析により求めます。拍点の数が最小のものが複数ある場合には、楽曲の先頭に近いほうの音符を捨象します。図-8において、 L の値域は、1 からメロディの非共通部分に現れる音符の数までの間です。この部分簡約で得られたメロディ A' は、メロディ B にはないメロディ A のみを持つ特徴の一部を減衰させたメロディと考えることができます。

ここまで見てきたように、タイムスパン木に *meet* や *join* という演算を適用して新しいタイムスパン木を算出してきました。それら演算やタイムスパン木には常にきちんと音楽的な意味が付与されていた点にご留意ください。

計算論的音楽理論のこれから

1983年に音楽理論 GTTM が提案されてから今年で25年が経ちましたが、GTTM の計算機上への実装やその応用システムについて本格的に検討が始まったのは、わずか10年ほど前からです。したがって、計算論的音楽理論は今ようやく成果が出始めた段階と言えるでしょう。

連載最終回となる本稿では、計算論的音楽理論の応用として、音楽理論 GTTM に基づく類似性の判定、メロディ予測、メロディ生成の3つの音楽タスクを紹介しました。おそらく今後の10年間には、システム側が音楽的な構造を十分理解できるようになり、ユーザの意図をもって出力をコントロールできるようにもなるでしょう。

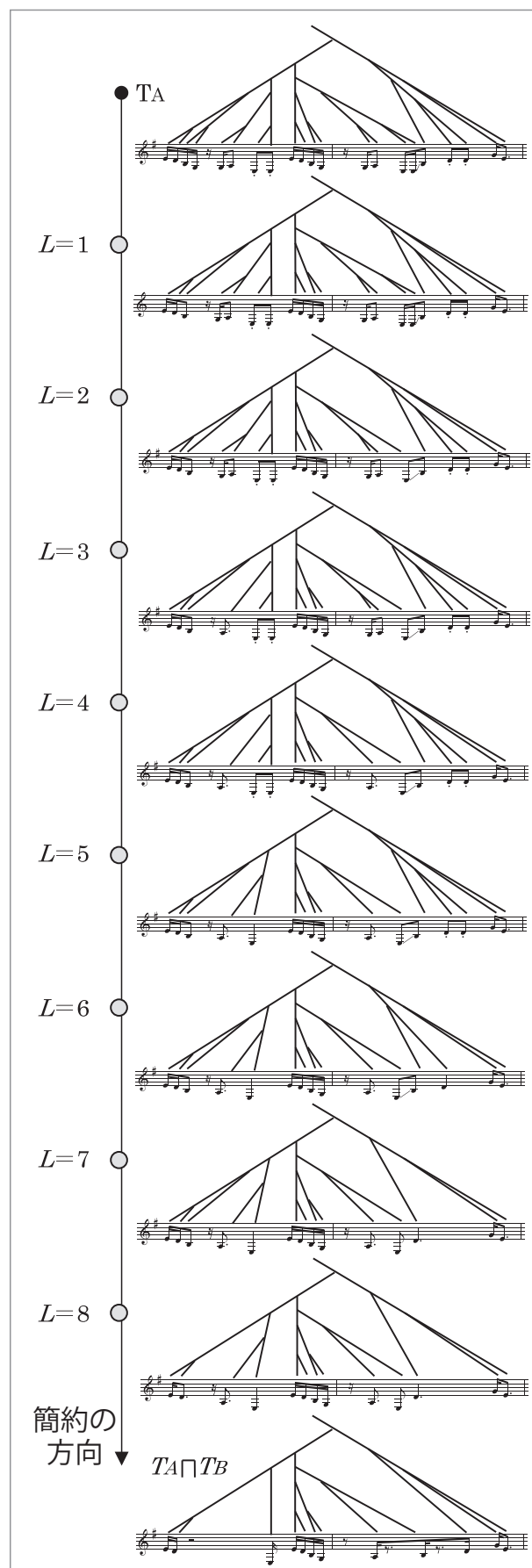


図-8 メロディの部分簡約

う。作曲、編曲をはじめとして、インタラクティブで創造的な即興演奏、演奏の表情付け、プレイリスト作成、



楽曲ライナーノーツの生成、動画、ジェスチャといった他メディアとの統合など、計算論的音楽理論を利用したさまざまな生成的、創造的な試みが行われることになるでしょう。

さて、今後の計算論的音楽理論の展開という点とまず第1に、エンドユーザのための音楽ツール開発という発想がくるでしょう。今回紹介した音楽タスクはその一例です。しかし、それだけではありません。

第2に、音楽自体への分析手段としての展開があります。本連載の第2回、4回で触れたように、音楽の構造を正しく計算の視点から見直すことにより、音楽学そのものの精緻化、音楽に対する理解の深化が期待されます。これまでの伝統的な音楽学には、論理的、数理的な表現や考え方が欠如していましたが、音楽には（も）ゲシュタルト（Gestalt）が存在するという原理から、記号論や情報学的な意味での構造や計算という世界を作り出せそうなのが分かってきました。特に、楽曲構造の安定性という概念を定式化し定量化した意義は計り知れません。これにより、たとえば楽曲の類似性の根拠、音楽の系統分類、特徴分類などが、主観を排した言葉で語られることになるわけです。楽曲に関して、順序や列、大小比較といった数論の概念とのアナロジーも生まれます。計算論的音楽理論の拡大は、計量音楽学（metrical musicology）とでもいうべき学際領域を創出する可能性さえあるのではないかと考えています。

そしてさらに第3に、メディア処理技術の理論的基盤の一翼となる展開もあるのではないのでしょうか。音楽は、人を取り巻く数多くのメディアの1つに過ぎません。しかし、音楽以外にもゲシュタルトの存在を仮定できるような非言語メディアは少なくありません。計算論的音楽理論は、もちろん音楽を対象として構築されたものですが、それを非言語メディアへ拡張することは、それほど不自然ではないように思えます。本連載の第3回では、音楽と言語はあくまで別物として、それらのアナロジーについて論じました。たしかに、非言語メディアによって表現され伝達される意味と言語によって表現され伝達される意味は、一般に相互に変換できないものです。ところが、人は言語や非言語メディアを自由自在に使い分けて、あるいは組み合わせでコミュニケーションを行っています。これはよく考えると非常に不思議で高度なことですが、非言語メディアの意味と言語の意味を統合すること、その上に技術体系を構築することには、大いに知的好奇心をそそられます。そしてこの試みによって、人の知性の本質に一步近付けるような気もしています。

参考文献

- 1) Hamanaka, M., Hirata, K. and Tojo, S. : FATTA : Full Automatic Time-span Tree Analyzer, Proceedings of the 2007 International

- Computer Music Conference (ICMC), Vol.1, pp.153-156 (2007).
- 2) Lerdaahl, F. and Jackendoff, R. : A Generative Theory of Tonal Music, The MIT Press (1983).
- 3) Lerdaahl, F. : Tonal Pitch Space, Oxford University Press (2001).
- 4) Hewlett, W. B. and Selfridge-Field, E. (Eds.) : Melodic Similarity : Concepts, Procedures, and Application, *Computing in Musicology*, Vol.11, The MIT press, Cambridge (1998).
- 5) どこまでも行こう vs 記念樹裁判文 : http://www.remus.dti.ne.jp/~astro/hanketsu/kousai_hanketsu_2.0.pdf 東京高等裁判所判決文(平成14年9月6日)。
- 6) Hirata, K. and Aoyagi, T. : Computational Music Representation Based on the Generative Theory of Tonal Music and the Deductive Object-Oriented Database, *Computer Music Journal*, 27(3), pp.73-89 (2003).
- 7) Hirata, K. and Matsuda, S. : Interactive Music Summarization Based on GTTM, Proc. of the 3rd International Symposium on Music Information Retrieval (ISMIR), pp.86-93 (2002).
- 8) 後藤真孝 : SmartMusicKIOSK : サビ出し機能付き音楽試聴機, 情報処理学会論文誌, Vol.44, No.11, pp.2737-2747 (Nov. 2003).
- 9) Cope, D. : Experiments in Musical Intelligence, A-R Editions, Inc. (1996).
- 10) Conklin, D. and Witten, I. : Multiple Viewpoint Systems for Music Prediction, *Journal of New Music Research*, 24(1) : 51-73 (1995).
- 11) Trivino-Rodriguez, J. L. and Morales-Bueno, R. : Using Multiattribute Prediction Suffix Graphs to Predict and Generate Music, *Computer Music Journal*, 25:3, pp.62-79 (Fall 2001).
- 12) Hamanaka, M., Hirata, K. and Tojo, S. : Melody Expectation Method Based on GTTM and TPS Proc. of the 9th International Symposium on Music Information Retrieval (ISMIR), pp.107-112 (2008).
- 13) CrestMuse プロジェクト : <http://www.crestmuse.jp/index-j.html>
- 14) Ribeiro, P., Pereira, F. C., Ferrand, M. and Cardoso, A. : Case-Based Melody Generation with MuzaCazUza, In G. Wiggins (ed.), Proc. of the AISB'01 Symposium on Artificial Intelligence and Creativity in Arts and Science, pp.67-74 (2001).
- 15) Arcos, J. L. and De Mantaras, R. L. : An Interactive Case-Based Reasoning Approach for Generating Expressive Music, *Applied Intelligence*, Vol.14, No.1, pp.115-129 (2001).
- 16) Hamanaka, M., Hirata, K. and Tojo, S. : Melody Morphing Method Based on GTTM, Proc. of the 2008 International Computer Music Conference (ICMC), pp.763-766 (2008).

(平成20年9月12日受付)

浜中雅俊（正会員） hamanaka@iit.tsukuba.ac.jp

2003年筑波大学大学院工学研究科博士課程修了。同年日本学術振興会特別研究員PD、さきがけ研究員（専任）などを経て2007年より筑波大学大学院システム情報工学研究科講師。音楽情報処理の研究に従事。博士（工学）。2001年山下記念研究賞、2001年SCI in Art 優秀論文賞、2003年筑波大学大学院優秀論文賞（博士課程長賞）、ICMC2005 Best Paper Award 受賞。

平田圭二（正会員） hirata@brl.ntt.co.jp

1987年東京大学大学院工学系研究科情報工学専門課程博士課程修了。工学博士。同年NTT基礎研究所入所。1990～93年（財）新世代コンピュータ技術開発機構（ICOT）。1999年よりNTTコミュニケーション科学基礎研究所。2001年本会論文賞、2003年山下記念研究賞。音楽情報処理に興味を持つ。ビデオコミュニケーションシステムt-Roomのプロジェクトに取り組む。

東条 敏（正会員） tojo@jaist.ac.jp

1981年東京大学工学部計数工学科卒業、1983年同大学院工学系研究科修了。1995年同大学院博士（工学）。1983～95年三菱総合研究所、1995年北陸先端科学技術大学院大学情報科学研究科助教授、2000年同大教授。自然言語の形式意味論および人工知能の論理の研究に従事。人工知能学会、ソフトウェア科学会、言語処理学会、認知科学会各会員。