

Mixtract：ユーザの意図に応える演奏表現デザイン支援環境

橋田光代^{†1,†2} 田中駿二^{†1} 片寄晴弘^{†1,†2}

演奏の表情付けは、指定された音の並びに対して音量やテンポ、アーティキュレーションに変化を与え、音楽を生き活きとしたものとして実体化する作業である。近年、音楽コンテンツデザインの中での重要性が増してきている。本稿では、フレージングに焦点を当てた演奏の表情付けを支援するシステム Mixtract と、保科の音楽解釈理論に準拠した Mixtract 上での演奏デザイン手法について述べる。保科の音楽解釈理論の運用にあたり、Mixtract では、フレーズ構造解析支援、フレーズ中での頂点音の推定・提示機能を提供した。これらの機能と方法論の利用により、演奏の表情付けデザインの効率性が高まること、また、ユーザの演奏表現に関する思考が客観的なものへと移行することが確認された。

Mixtract: an Directable Design Supporting Environment for Music Expression

MITSUYO HASHIDA,^{†1,†2} SHUNJI TANAKA^{†1}
and HARUHIRO KATAYOSE^{†1,†2}

Performance rendering is processing to embody musical idea in concrete sound, giving expression to tempo, dynamics and articulation to each note. Recently its importance has been increasing in terms of multimedia content production. This paper introduces a performance design environment called Mixtract, that assists the users to design "phrasing," and a performance design guideline called Hoshina-Mixtract method based on Hoshina's music interpretation method, for Mixtract. Mixtract provides its users a function for phrase analysis and a function to analyze importance of each note in a phrase group. We verified the validity of the proposed system and the method in improving productivity of music performance design and letting users' musical thought from a subject one to an objective one.

†1 関西学院大学理工学研究科 / ヒューマンメディア研究センター

Research Center for Human & Media, Kwansai Gakuin University

†2 科学技術振興機構 CREST「デジタルメディア領域」CrestMuse プロジェクト

1. はじめに

演奏の表情付けは、指定された音の並びに対して音量やテンポ、アーティキュレーション^{*1}に変化を与え、音楽を生き活きとしたものとして実体化する作業である。言うまでもなくこの作業は人間の知性や感性のあらわれとして実施されるものであり、その自動化研究は、自動作曲編曲、耳のモデリングと並んで、音楽系人工知能研究における3大主要課題の一つとして取り組まれてきた¹⁾。最近では、ゲームやアニメ、一般ユーザによるコンテンツ制作の隆盛にも関連して、演奏表情付けの重要性が大きく高まっている²⁾。

演奏表情付けにおける計算機による支援に関しては、上記の自動化技術の他に、ユーザ主導のデザインをアシストするというインタフェース系の研究がある。自動化技術は生産性向上という視点において大きなメリットを持つ一方で、必ずしもユーザが求めるデザインにたどり着くのは容易ではないという問題がある。

我々は、フレージングに焦点を当てたユーザ主導の演奏デザインにおいて、煩雑な作業の部分を自動化技術で代替するという方式の演奏デザインシステム Mixtract³⁾を提案・実装してきた。フレージングはいわゆる発声、呼吸に関連した音楽表現であり、一般の音楽ユーザにとっても直感的にくみし易い音楽表現である。しかし、演奏生成に関わる方法論が一切規定されない状態で、デザインを実施していくのは容易なことではない。そこで、我々は、音楽解釈学、特に、プラスバンドの演奏表現と指導において日本で最も尊敬されている保科洋氏の音楽解釈理論⁴⁾を定式化し、Mixtractによる演奏デザインに活用できるようにした。以下、第2章でフレージングと Mixtract の概要について述べ、第3、4章でフレージングを支援するための各機能の詳細について述べる。第5章で、保科の演奏デザインの具体的な方法論について述べるとともに提案手法の有効性について議論する。

2. Mixtract 概要

2.1 フレージングと Mixtract

どのような音楽においても、ある時刻における音とそれに前後する音との間になんらかの関係を持ったまとまり（フレーズ）が形成され、複数のフレーズがさらに関連付けられて、階層的なフレーズ構造が形成される（図1）。この階層的フレーズ構造をいかに聴取者へ伝

JST/CREST CrestMuse Project

*1 音や音のつながりに影響をもたらす音の形の表現。スラー、スタッカート、アクセントなどの記号やその表現を指す。

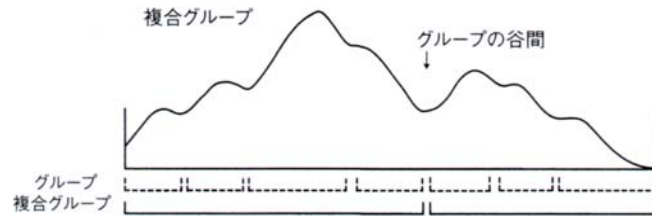


図1 フレーズング(文献4)より引用). 横軸が時間軸. 縦軸が音量やテンポの表現である. フレーズ(この図においてはグループ)には階層構造であり, その階層が音量やテンポとして表現される.

達するかが音楽表現の主題であるといっても過言ではない⁵⁾. ところで, 作曲者が用意した楽譜(音符の並びや演奏記号)から解釈されるフレーズ構造は必ずしも一つとは限らず, いくつかの可能な解釈がある. 演奏家や指揮者は, 可能なフレーズ構造の解釈の中から一つを定め, その構造が伝わるように演奏表現として具現化する. また, フレーズ構造の演奏表現も必ずしも一つではなく, テンポ表現, ダイナミクス, アーティキュレーションの組み合わせによって演奏の個性が表出される.

演奏デザインシステムの構成においては, 上記のような演奏表現上の自由度を確保することは重要である. しかし, それだけでは, 演奏デザインの環境として十分ではない. 1) ユーザの音楽的な試行を促す環境として機能すること, 2) ユーザの「こだわり」を確保した上で補完的な自動処理が用意されていること, の2点が満たされることが求められる. Mixtract は, 以上のような考えに基づいて構成された演奏デザイン環境である.

Mixtract の概略を図2に示す. ユーザは, エディタ, もしくは, MusicXML のインポート機能を利用して楽譜情報を入力する. 次に, フレーズ構造を指定し, 各フレーズに対する表情カーブ(テンポ, ダイナミクス, アーティキュレーション)を与え, その足し合わせにより演奏表情の概略のデザインを行う. 必要に応じ, 適宜, 各音の開始時刻, 終了時刻, 音長等の調整を実施する. エディティングの任意の段階で試聴が可能で, 表情カーブのエディティングにあわせてピアノロールのプレビューが更新される. ユーザはこれらのユーザフィードバックを手がかりに演奏デザインを進める.

Mixtract は, ユーザ主導でフレーズ構造を定めるべきという立場を取っているが, ユーザデザインを支援することを目的として, 認知的音楽理論を応用したフレーズ構造分析機能を用意している. 以下にその概略を述べる.

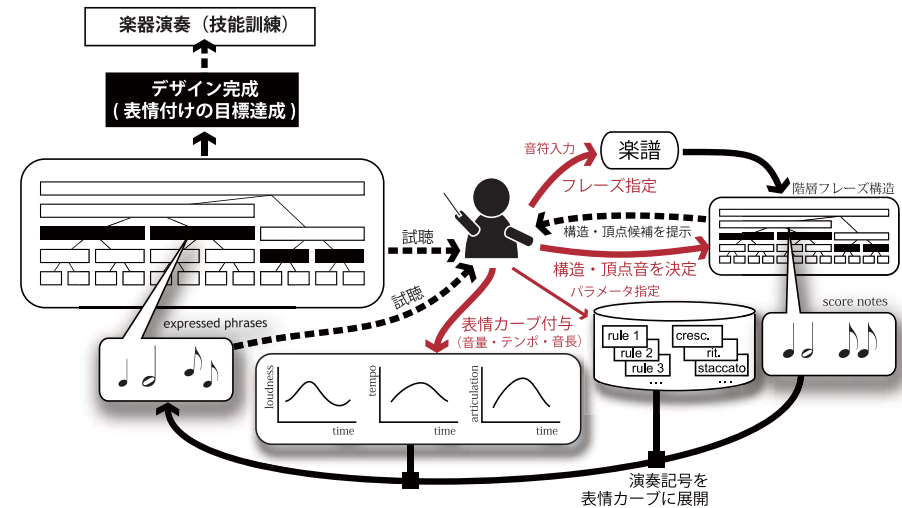


図2 Mixtract 概略

2.2 フレーズ構造解析支援

ユーザ主導のフレーズ構造解析において, ユーザが階層的フレーズ構造すべてを記述しなければならないとなると極めて煩雑な作業を強いことになる. そこで, フレーズ構造解析支援として, ユーザが部分的に指定したフレーズ以外の構造については GTTM (A Generative Theory of Tonal Music⁶⁾) を応用して解析する機構を用意した⁷⁾. GTTM そのものは, 音楽構造の多義性に対処できるものではないが, ユーザとのインタラクションプロセスを前提とした処理の構成により, 効率的, かつ, ユーザの負担が少ないフレーズ構造解析が可能となる.

2.3 フレーズ中の頂点の推定

フレーズの典型的な表現としては, (1) 音量やテンポを山を描くように変化させる表現, (2) フレーズの開始音にアクセントを与える表現などが知られている⁸⁾. Mixtract では, このような前提知識を用い, 試聴を繰り返すことで表情カーブのエディティングを行うが, 演奏生成に関わる方法論が一切規定されない状態で, デザインを実施していくのは容易なことではない.

Mixtract では, 表情カーブのデザインを支援するものとして, 保科の提唱した音楽解釈理論⁴⁾を定式化し, フレーズを構成する音符が持つエネルギー値を推定, 表示する機能を用意している. 保科は, 楽譜の解析から得られるフレーズにおいて最もエネルギー値が高い音符

を、頂点もしくは重心と定義し、これらの音のエネルギー値を音量やテンポ等の演奏表現上での制御に置き換えていくことで演奏が生成できると説明している。頂点の演奏表現法は一意ではないが、頂点を中心とした表情カーブを描いて行くスタンスを取ることで、音楽的な破綻を避けた演奏のデザインが可能となる。

3. 演奏表情エディットと表情の合成

Mixtract では、各フレーズ構造に対する表情カーブを用いた演奏デザインインタフェースと、各音の開始時刻、終了時刻、音量を操作するエディタが用意されている。以下、表情カーブを利用した演奏デザインとその表情合成方法について述べる。

3.1 表情カーブのエディティング

フレーズ構造が決定されると、Mixtract は各階層のフレーズの表情カーブとして、テンポカーブとダイナミクスカーブ、アーティキュレーションカーブを用意する。ユーザが各フレーズをダブルクリックすると、これらのカーブエディタが出現する。この様子を図3に示す。ユーザは、出現したデフォルト表情カーブを自由曲線で修正することによって演奏のデザインを実施する。

3.2 テンポカーブと発音・消音時間の計算

テンポカーブ自体は、指数表現、すなわち、変化無しの場合 0、テンポが倍になる場合は 1、テンポが半分になる場合は -1 として与えられる。楽曲全体でのテンポカーブは、次式によって計算される。

$$Tempo(t) = \sum_{k=1}^n w_k \cdot GroupTempo_k(t) \quad (1)$$

ここで、 t は楽譜時間、 $GroupTempo_k(t)$ は、楽譜上での時刻 t をその範囲に持つフレーズのテンポカーブ、 w_k はそのフレーズの重みである。各音の発音・消音時刻については、 $Tempo(t)$ の積分値によって定められる。つまり、楽譜上の時刻としてそれぞれ、 t_0 、 t_1 に配置されたイベント間の経過時間 $Time(t_0, t_1)$ は、テンポカーブが指数で表記されていること、テンポが時間進行の逆数になることから、次式によって計算される。この操作を繰り返して行くことにより、各音の発音・消音時刻が決定される。

$$Time(t_0, t_1) = \sum_{t=t_0}^{t=t_1} DeltaDuration \cdot 2^{-Tempo(t)} \quad (2)$$

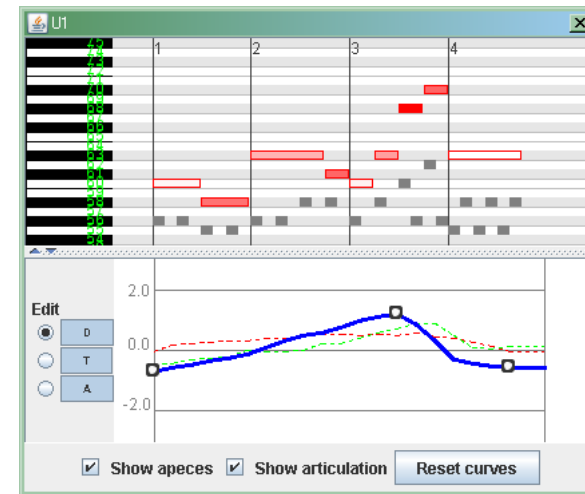


図3 表情カーブとそのエディタ。図下部において、D (ダイナミクス: 青)、T (テンポ: 緑)、A (アーティキュレーション: 赤) の各ボタンを ON にすると、対応する表情カーブがエディタに点線表示される。Edit ラジオボタンで選択された表情カーブは太線で示され、マウス操作によりこの画面上で直接編集できる。ピアノロール部において、各音符に対する頂点らしさは赤色の濃淡で示され、赤色が濃いほど、頂点である可能性が高くなる(第4章)。

ここで、 $DeltaDuration$ は、曲全体の bpm と式(2)での積分区間の分解能によって規定される値である。このような計算を実施するメリットとして、指揮システム等において実時間のスケジューラを構成する際に、拍以下の単位で、だんだん速くする、あるいは、遅くする等の処理がシンプルに実装できる点がある。

3.3 ダイナミクスカーブと音量の計算

音量の計算についても、上記の発音・消音と同じく、楽譜上での時刻 t をその範囲にもつフレーズのダイナミクスカーブの足し合わせによって計算される。

$$Dyn(t) = \sum_{k=1}^n w_k \cdot GroupDyn_k(t) \quad (3)$$

ダイナミクスカーブは MIDI における velocity を基準として設定されることから、楽譜上での時刻 t の音符の velocity は次式によって計算される。

$$Vel(t) = StdVel + Dyn(t) \quad (4)$$

ここで、*StdVel* はデフォルトの velocity 設定値である。

3.4 アーティキュレーションカーブと消音時刻の計算

アーティキュレーションとは、音や音のつながりに関する表現である。現時点での Mixtract では、ピアノのような打鍵楽器を対象としている。この制約下では、アーティキュレーションの表現の主要な制御対象は各音符のノートオフのタイミングとなる。

i 番目の音符のノートオフのタイミング $N_{Off}(i)$ は、*i* + 1 番目の音符の開始時刻を参照する形で、次式によって設定される。

$$N_{Off}(i) = N_{StartTime}(i+1) + StdBeatTime \cdot GroupArtcltn(N_{StartTime}(i+1)) \quad (5)$$

ここで、 $N_{StartTime}(i)$ 、 $N_{TimeValue}(i)$ は、それぞれ、*i* 番目の音符の開始時刻と音価、*StdBeatTime* は基準拍の長さ、*GroupArtcltn(t)* は、当該の音符が含まれるフレーズのアーティキュレーションカーブの時刻 *t* における比率である。

3.5 演奏記号の表現

楽譜情報としてクレッシェンドやリタルダンド、スタカート等の演奏記号が付加されている場合への対応として、Mixtract には、それらをルール処理によって表情カーブに展開する機能が用意されている。演奏記号が適用されると、その演奏記号に対応する新たなフレーズが形成され、対応する表情カーブのデフォルト値が設定される。ユーザは、必要に応じ、エディタを用いて表情カーブを修正することができる。

3.6 和音の演奏表情とポリフォニーについて

Mixtract 現バージョンでのデータの内部表現は、シングルパートすなわち一本の五線譜にすべての音符を配置するような形式を採用している。和音を含む旋律の表情付けは可能であるが、厳密な意味で言えば、別の時間（テンポ）進行で音が配置される音楽、つまり、ポリフォニーの演奏デザインは対象外となっている。この部分の実装は今後の課題である。

4. 頂点解析支援機構

Mixtract では、保科理論に即したフレージング手法を支援するものとして、頂点（重心）の自動解析機構を提供している。保科による頂点の解析例を図 4 に示す。

文献 4) では、(1) 音高が高いもの、つまり、音形輪郭における頂点音、(2) 音価が長いもの、(3) 各音の緊張 - 弛緩構造における緊張音、(4) 和声などによって、頂点らしさが定まってくると主張している。また、フレーズの最終音は頂点にならないというような原則も与えられているが、計算可能モデルの定式化はなされていない。

ここでは、これらの音形輪郭や和音進行の原則をルールの条件節として使用できるよう類

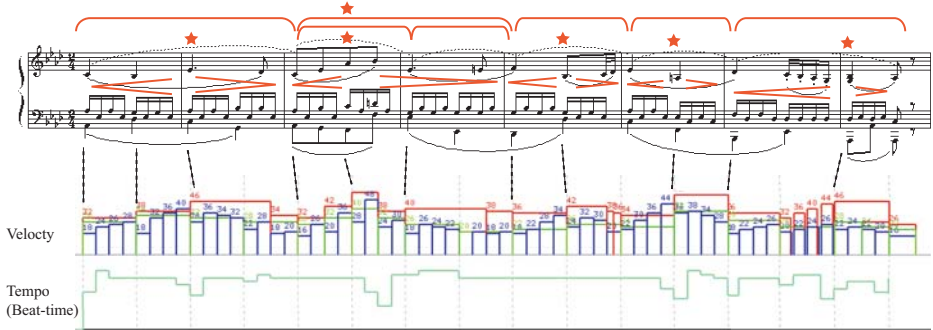


図 4 ピアノソナタ「悲愴」第 2 楽章の演奏データとその表現意図。頂点が星型マークで示されている。

型化し、さらに、臨時記号等に関するルールを新たに用意した⁹⁾。各音の頂点らしさの計算は、条件に合致した際、各音の頂点らしさのポイントを加算する処理によって実施される。図 5 に使用されるルールおよびパラメータを示す。ドミナント・モーションなど音楽的に意味を持つ音の並びの表現にも対応するため、各ルールにおいては、該当する音の他に、先行音や後続音に対してもポイントが設定されている。なお、各ルールのポイントについては、保科が示した頂点解析例ほか、数曲の譜例による予備調査の結果を満たすよう経験的に設定した。この処理によって計算される各音の頂点らしさの計算結果は、ピアノロール上の各音符に対する描画色の濃淡情報として表示される（図 3 参照）。

5. Mixtract を用いた演奏デザインと評価

5.1 演奏デザインメソッド

Mixtract は、フレージングを主体とした演奏デザインのフレームワークを提供している。Mixtract 自体はフレームワークであるため、フレーズの階層をいくつにすればよいのか、上位、下位階層のいずれから演奏デザインを実施するのか、音量やテンポカーブを自由曲線とするのか直線で近似するのか等、演奏のデザイン法は規定していない。自由度の確保は使い手のメリットとなる一方で、演奏デザイン初心者にとっては、どのように演奏デザインに取り組めばよいかが戸惑ってしまう要因にもなる。そこで、演奏デザインメソッドの一例として、保科の提唱する音楽解釈理論を Mixtract 環境下で実行できるよう、保科-Mixtract メソッドの定式化を行った。

保科-Mixtract メソッドにおいて、ユーザはまず、概ね 1 ~ 数小節分の長さを目安に、一

分類	Rule	Nnが各Ruleにおいて以下に該当するときの評価ポイント(w)			
		先行音	対象音	後続音	
音価	隣接する2音の第1音		後続音より短い // 長い	0 1	
	同一音価が連続する音群の第1音			1 0	
音高	隣接する2音の第1音		後続音より低い // 高い	0 1	
	同一音価が連続する音群の第2音以降			1 0	
進行到達音 (隣接する4音の第3音)	1) 上行-上行-下行			0	
	2) 下行-上行-下行			2	
	3) 上行-下行-上行			1	
	4) 下行-下行-上行			0	
	5) 1)または2)で基本音長が0.25秒以下			1	
旋律的 緊張-弛緩	倚音		音価が後続音より長い // 後続音と同じ // 後続音より短い	3 2 1	
			後続音が反行する	2	
			順次進行の途中	1	
			後続音へ跳躍進行する	2	
			// 順次進行する	1	
			予備音が休符	1	
	掛留音		予備音にタイあり // なし	3 2	
	持続音		予備音にタイあり // なし	2 1	
	刺繍音		上行して下行	2	
	経過音		下行して上行 上行形 下行形	1 2 1	
その他変化音 (臨時記号)			1		
音種 (和声)	和音		I I6 I46 II IV V V7 VI	0 1 3 6 1 2 3 1	
	借用和音		変化音である // ではない	1 2	
	先行和音から変化			1	
	カデンツ (隣接する2音の第1音)		V(7) - I V(7) - I6 V(7) - VI	4 2 2	
				0 3 1	
	音域		先行音より広がる // 狭まる	1 -1	
	旋律密度		重音	1	
	グループ	開始音			1
		終了音			-1
		最長音		グループにバウンド分割を含む // 含まない	1 2
最高音			グループの平均音価より長く、0.25秒以上 その他	2 1	
グループ中最大の 上行跳躍音			グループの平均音価より長く、0.25秒以上 その他	0 2	
その他	アーティキュレーション		アクセント テヌート 非裝飾音 スタッカート	1 1 1 1	
	スラー (演奏時間1秒程度以下)		開始音 終了音	1 -1	
	休符		強拍にあり、後続音が弱拍である 弱拍にあり、後続音が強拍である	0 1	
				1 0	

図 5 頂点の計算のためのルールおよびパラメータ

列に連なるフレーズ群 (以下、プライマリフレーズライン) を形成していく。楽譜情報のみを手がかりとしてフレーズ境界を判断することは必ずしも容易ではないが、楽曲の試聴機能を利用することで、初心者でもほぼ直感的にこの作業を実施することができる。続いて、プライマリフレーズラインに属するフレーズに対して、頂点情報を参考にしつつ、表情カーブを編集し、必要に応じて、各音の開始時刻、終了時刻、音量を設定する。プライマリフレーズラインの上位構造については、その構造がまとまりをもって聞かれる程度に控えめの表情カーブを付与しておく。保科-Mixtract メソッドの具体的なステップを以下に示す。

保科-Mixtract メソッド

Step1 プライマリフレーズラインの指定

- (1) ユーザによる初期設定
- (2) 簡易プレイバック機能を利用したプライマリフレーズラインの確定

Step2 プライマリフレーズライン中フレーズに対する頂点決定

- (1) 信頼度付き頂点候補の提示
- (2) (ユーザによる頂点の確定)

Step3 頂点情報を利用したプライマリフレーズラインの演奏デザイン

- (1) (上記の解析に基づいた表情カーブのデフォルト値の自動設定*1)
- (2) 表情カーブのエディティング
- (3) (演奏記号の表現展開のマクロオペレータ処理)

Step4 プライマリフレーズラインの上位構造に対するまとまり感の付与

- (1) (頂点に基づいてダイナミクスカーブ、テンポカーブ山型表現のデフォルト値自動設定)
- (2) 頂点位置とダイナミクスカーブ、テンポカーブレンジの調整

Step5 (各音の開始時刻、終了時刻、音量の調整)

なお、括弧 () で記した事項は必要がなければ割愛可能な処理である。また、Step4において、先に確定させたプライマリフレーズラインよりも上位構造の演奏デザインを重視したくなった場合は、当初設定したプライマリフレーズラインの重み係数を下げ、上位構造をあ

*1 1) 頂点音に対してのシンプルな山型のダイナミクスカーブとテンポカーブ、2) 各音の頂点らしさに応じたダイナミクスカーブとテンポカーブ、の二つから選択することができる

らためてプライマリフレーズラインとして再設定してステップを継続させてもよい。以上の一連の過程は、必要に応じて、適宜、前のステップに戻り、処理を繰り返すことも認められる。

5.2 検 討

1分間程度の中級者向けのピアノ曲に対して表情付けを行う場合、対象音符はおおむね500音程度あり、それぞれに演奏表情をつけないといけない。Mixtractを用いた場合、ユーザが指定するフレーズは10数個、それらの中から、ユーザが指定したフレーズのテンポとダイナミクスカーブを与えるだけで表情付けが行われる。演奏デザインの効率化という視点において、Mixtractが有効であることがわかる。

提案システムと方法論がユーザの演奏表現に関する思考にどのような影響を与えるかを把握することを目的として、ピアノ専攻で音楽修士号を取得した現役の高校教諭を教師役として、小～中学生7名を対象にフレーズ教育のワークショップを実施した。生徒からは「表情カーブの深さを動かして音が変わるのが面白かった」「家に置いてじっくり表現を考えていたい」「今回教えてもらったことを活かして部活にも集中していきたい」というような声が聞かれた。教師からは、「思い返せばフレーズを概念としてしっかり生徒に教えたことがなかったように思う。その意味で自分自身も勉強になったし、音楽の教え方そのものを再考させられた」という感想が聞かれた。

音楽経験者は、暗黙知としてフレーズの表現法を理解しているが、テンポや音量の表現を具体的にどのように組み合わせることでその表現ができあがっているのかを言語化できないことが多い。Mixtractの利用により、フレーズに関する暗黙知を形式化、類型化できる期待がある。今後、ワークショップを定期的開催し、音楽教育分野への実利用を進めるのに併せて、演奏デザインメソッドの改訂を進めて行く計画である。

3.6節で述べたように、現時点でのMixtractはシングルパート用として実装されており、マルチパート対応になっていない。また、フレーズ構造解析支援および頂点解析機構において、和声解析は、現時点では、自動処理が実現されておらず、手作業によって与える仕様となっている。今後の課題として、これらへの対応が求められる。

6. おわりに

本稿では、フレーズに焦点を当てた演奏表情付けを支援するシステムMixtractのデザインコンセプトを述べ、フレーズ中での頂点音の推定・提示機能について説明を行った。

Mixtract自体は、演奏デザインのフレームワークであり、さまざまな演奏デザインの適用

が可能である。具体的な演奏デザインメソッドとして、保科の音楽解釈理論に準拠した保科-Mixtractメソッドを示した。

Mixtractの各機能と保科-Mixtractメソッドの利用により、演奏表情付けデザインの効率性が高まること、また、ユーザの演奏表現に関する思考が客観的なものへと移行することが確認された。Mixtractは、構造分析と表現のデザインの支援の統合環境を提供しており、ユーザは、自分自身の手で編集と試聴を繰り返すことで、対象楽譜に対する表情付けのアイデアを練り込んだり、新しい解釈の発想を促したりできる。楽曲分析や演奏表現の教育ツールとして利用が見込まれる。今後は、音の配置機能(狭義作曲機能)をサポートして、教育への展開に力を入れていきたいと考えている。

謝辞 本研究は、科学技術振興機構 戦略的創造研究推進事業 CREST「デジタルメディア領域」CrestMuseプロジェクトの支援を受けて実施されました。本研究を実施するにあたり、指導を賜った保科洋先生に感謝します。

参 考 文 献

- 1) 片寄晴弘：音楽における自動処理と Directability, オペレーションズ・リサーチ, 音楽とOR特集, Vol.54, No.2, pp.540-545 (2000).
- 2) 橋田光代, 北原鉄朗, 鈴木健嗣, 片寄晴弘, 平田圭二：演奏表情付けコンテスト EC-Rencon 開催報告, 情報処理学会研究報告 音楽情報科学 2009-MUS-83, No.1 (2009).
- 3) Hashida, M., and Katayose, H.: Mixtract: A Directable Musical Expression System, *Affective Computing and Intelligent Interaction (ACII)* (2009).
- 4) 保科 洋：生きた音楽表現へのアプローチ:エネルギー思考に基づく演奏解釈法, 音楽之友社 (1998).
- 5) 小澤征爾, 堤 剛, 前橋汀子, 安田謙一郎, 山崎伸子(編): 斉藤秀雄講義録, 白水社 (1999).
- 6) Lerdahl and Jackendoff: *A Generative Theory of Tonal Music*, MIT Press (1983).
- 7) 田中駿二, 橋田光代, 片寄晴弘：演奏表情付けシステムのためのユーザ主導型音楽構造解析, *Entertainment Computing(EC)2009* (2009).
- 8) 橋田光代, 長田典子, 河原英紀, 片寄晴弘：複数旋律音楽に対する演奏表情付けモデルの構築, 情報処理学会論文誌, Vol.48, No.1, pp.248-257 (2007).
- 9) 橋田光代, 片寄晴弘, 野池賢二, 保科 洋, 河原英紀：音楽聴取に関する一検討:グループと頂点の推定, FIT2004 情報科学技術レターズ, Vol.3, No.3, pp.145-148 (2004).