

## エレキギターの表情付け支援システム 「Guitar-Case Maker」

水本直希<sup>†1</sup> 馬場隆<sup>†1</sup>  
北原鉄朗<sup>†2</sup> 片寄晴弘<sup>†1</sup>

近年、計算機上の音楽制作が一般的になったことから、音楽の演奏表情付け技術の需要が高まっている。しかし、既存研究ではピアノにおける演奏表情付けがほとんどであり、ギターやバイオリンといった、演奏者が発音後にピッチやダイナミクスを制御できる持続系楽器の演奏表情付けについてはあまり扱われてこなかった。本研究では、ポピュラー音楽において使用頻度の高いエレキギターを対象とし、MIDI データにおけるエレキギターの演奏表情付け支援システムの開発を目的とする。本稿では、システムで用いる事例ベースによる演奏表情付けアルゴリズムと動的計画法を用いた運指推定アルゴリズムの概要について報告する。

### Guitar-Case Maker: Performance Rendering Support System for Electric Guitar

NAOKI MIZUMOTO,<sup>†1</sup> TAKASHI BABA,<sup>†1</sup>  
TETSURO KITAHARA<sup>†2</sup> and HARUHIRO KATAYOSE<sup>†1</sup>

The recent increases of non-professional people who create their own musical pieces have demanded technologies for rendering expressive musical performances. Most of the existing studies for performance rendering, however, targeted the piano's performances, so they did not deal with sustained-tone instruments, the pitch and dynamics of which can be continuously changed. Focusing on the electric guitar as a sustained-tone instrument, we have been developing a performance rendering support system for electric guitar solo phrases. In this paper, we describe a case-based performance rendering algorithm and a guitar fingering estimation algorithm with dynamic programming.

### 1. はじめに

エレキギターは現代のポピュラー音楽において欠かせない存在である。そのため、DTM<sup>\*1</sup>を用いたアマチュアによる音楽制作においてもエレキギターの音色が使われることが少くない。しかし、エレキギターは自然な演奏を DTM により表現するのが困難な楽器として知られている。近年、大容量のソフトウェア音源の登場により本物と変わらないほどの音を生成できるようになった。それらは音源の質が良い点では評価できるが、あくまで人間らしい演奏生成への下地でしかなく、自然な演奏を自動あるいは半自動的に生成する計算機技術が望まれている。

エレキギターの演奏生成が難しいのは、エレキギターではチョーキング、ハンマリング・オンなどといった様々な奏法が使用され、それらをピッチの細かな連続的制御によって適切に表現し分ける必要があるからである。これまでに演奏の自動生成（演奏表情付けという）に関する様々な研究<sup>1)–8)</sup>が行われてきた。しかし、そのほとんど<sup>2)–6)</sup>がピアノを対象としており、エレキギターで多用される発音後のピッチの連続的制御は扱っていなかった。一方、発音後にピッチやダイナミクスを制御可能な楽器（持続系楽器という）を扱った研究として、サクスを対象とした SaxEx<sup>7)</sup>、バイオリンを対象とした清水らの研究<sup>8)</sup>が挙げられる。しかし、これらの研究で提案された技術がエレキギターに見られる多種多様な奏法に対して有効かどうかは検証されていない。また、エレキギターを対象とした研究として中西らの研究<sup>9)</sup>があるが、制御対象はベロシティであり、ピッチの連続的制御は扱っていなかった。

我々はエレキギターのソロフレーズを表す MIDI データに対してベロシティ情報とピッチバンド情報を付加することで自然なピッチ遷移を実現する演奏表情付け支援システム“Guitar-Case Maker”の開発に取り組んでいる。我々は、前報にて、様々な旋律にベロシティ情報とピッチバンド情報があらかじめ付与されたデータからなる事例データベースを用意し、入力された MIDI データに旋律の特徴に近いものをこのデータベースから探索し、探索された事例に付与されているベロシティ情報とピッチバンド情報を利用する演奏表情付けアルゴリズムを提案した<sup>10)</sup>。しかし、エレキギターの演奏において重要な要素である運指につい

<sup>†1</sup> 関西学院大学大学院 理工学研究科

Graduate School of Science and Technology, Kwansai Gakuin University

<sup>†2</sup> 日本大学 文理学部

College of Humanities and Sciences, Nihon University

\*1 DTM (Desktop Music): パソコンと電子楽器を MIDI ケーブルなどによって接続し演奏する音楽、あるいはその音楽制作行為の総称。

て重要な要素である運指については扱われていなかった。本報告では、演奏表情付けの質をより向上させるため、運指情報を MIDI データから推定し、演奏表情付けに導入する手法について取り扱う。

以下、2 章では演奏表情付けにおける課題とアプローチについて議論する。3 章では事例ベースによる演奏表情付けアルゴリズムについて述べる。4 章では運指最適化問題を多段決定問題として定式化し、動的計画法を用いた運指推定アルゴリズムについて述べる。5 章では、システムの実装および提案アルゴリズムの性能を検証するために行った予備実験について述べ、最後に 6 章でまとめと今後の課題について述べる。

## 2. 演奏表情付けにおける課題

エレキギターは撥弦楽器の一種であるが、他の撥弦楽器と比べて特徴的な事項として、撥弦して音が鳴っている状態でピッチを変化させる奏法が多用されることがあげられる。特に、いわゆる歪み系のエフェクトを使用したときは撥弦から長時間音が持続するため、このような奏法が頻繁に使用される。

撥弦後にピッチを変化させる奏法は数多くあり、代表的なものとしてチョーキングやハンマリング・オンが挙げられる。ギターソロにおいてこれらの特殊奏法が用いられる位置にはある程度の共通性があるが、ギタリストによる違いも存在し、その位置を決定する絶対的な条件は明らかではない。そのため、ピッチの時間的変化を適切に付与することで、特定のギタリストに対するその人らしさを表現できると考えられる。以上の議論に基づき、本章では、本研究での問題設定や課題などを論じる。

### 2.1 問題設定

演奏表情付けが行われていない MIDI データを入力とし、ギタリストの演奏特徴が反映された MIDI データを出力とする。本システムの演奏表情付けの対象は単旋律であるギターソロのみとし、ピッチバンドとペロシティの 2 つを扱うこととする。

### 2.2 演奏表情付けのアプローチと課題

演奏表情付けを行う際に用いられるアプローチとして、大きく jPop-E<sup>11)</sup> のようなルールベースと Kagurame<sup>12)</sup> のような事例ベースの 2 つに分けられる。ルールベースは、音楽に関する知識や概念をルールという形で記述し、そのルールに基づいて音楽を生成する。例えば、フレーズの終わりの部分で減速するというようなルールを与えると、フレーズの存在を強調した演奏が生成される。この手法では楽曲の特徴と演奏表情との関連付けなどを行うことによって演奏表情を作り出す。メリットとして、専門家が注意深くルールを設計す

れば、品質の高い演奏を生成できる。しかし、エレキギターのピッチ制御は、ピアノ演奏に比べて学術的な分析や理論化がなされておらず、ギター演奏の専門家が持つ暗黙的な知識をルールとして書き下すことは、必ずしも容易ではない。一方、事例ベースはあらかじめ用意された演奏事例の事例を利用して演奏を生成する方法である。この方法では、利用する演奏事例集を切り替えることで異なった演奏表現が生成される。ユーザが好む演奏事例集を複数パターン用意しておくことで、期待した演奏表情を得やすいというメリットがある。

事例ベースによる手法のフレームワークとして、入力楽譜の 1 フレーズのような処理単位毎に探索条件に合致する事例を見つけ出して、その事例の特徴を直接反映させる「インスタンススペース」と、確率モデルのような何らかのモデルを想定し、事例からそのモデルパラメータを推定して、そのモデルを元に演奏を生成する「モデルベース」の 2 つが考えられる。前者は探索条件に合致する事例が見つからない場合に何も演奏表情が付与されないデータスパースネス問題に陥る可能性がある。後者には、精度よくモデルパラメータを推定するにあたり大量のデータが必要となり、出現頻度の低い特徴的な演奏が統計的推定によって埋もれてしまうという問題もある。

本研究では、ユーザが好むギタリストの演奏事例を大量に用意するのは困難であることからインスタンススペースの方法を採用する。次節でインスタンススペースにおける課題について議論する。

### 2.3 インスタンススペースにおける課題

演奏表情付けを行うアルゴリズムにインスタンススペースを用いる際、データスパースネス問題の解決が大きな課題となる。本研究では、この課題を含む以下の 3 つの事項を解決すべき課題として設定する。

#### 課題 1 データスパースネス問題

探索条件に合致する事例が見つからない場合に、何も演奏表情が付与されない可能性があるという問題

#### 課題 2 フレットの曖昧性の問題

ギターの運指は複数の可能性が存在し、押弦位置が一意に定まらないという問題

#### 課題 3 類似尺度の個人依存性の問題

楽曲の演奏表情に対する類似尺度が、個人間で異なるという問題

課題 1 のデータスパースネス問題は設定した探索条件が厳し過ぎる際に発生する。最も探索条件に符合した事例の探索を目的とする場合の探索条件は、より具体的、すなわち、厳

しいものとするのが望ましい。しかし、厳しい探索条件を課すことは、類似事例発見の可能性を低くしてしまう要因となる。どの程度の厳密性を条件に設けるかは扱う問題によって異なり、適切に設定することは難しい。本研究では、探索条件の厳密性は楽曲、さらには、音符毎にも異なるという前提に立ち、事例が見つからない場合に探索条件を段階的に緩くしていくという方策での類似事例の探索を実施する。具体的には、対象楽曲のある音符に対する類似事例を事例データベースから探索する際、その音符の近傍の音符との特徴が近い事例を探す。その際、近傍の音符数の値を大きくすれば旋律を考慮した類似事例探索となるが、それと同時に、類似事例が見つからない場合も多くなる。そこで、類似事例が見つからない場合は近傍の音符数を減らし、特徴が近い事例が多く探索できるようにする。これによりデータスパースネス問題を回避することが可能となる。

課題2のフレットの曖昧性の問題は、同一の旋律に対してギターの運指が複数存在するために起こる。例えば、全く同じ旋律を、ネックに近いフレットとピックアップに近いフレットの両方で演奏することが可能である。フレットでの演奏位置が異なれば、音色や弦のしなり方が異なり、また、奏法における身体的な制約も異なったものとなる。ギタリストによるソロ演奏は、当然、これらの点を考慮して実施されている。運指が共通であれば、異なる旋律であっても奏法にも共通の特徴が生じると考えられる。本研究では、これらの事項を考慮し、MIDIデータには記述されていない運指情報を動的計画法により推定し、事例探索時の一特徴量として利用する。

課題3の類似尺度の個人依存性の問題は、演奏表情に対するとらえ方や演奏表情のどの特徴に着目するかが個人によって異なるという問題である。類似事例探索のための類似尺度をある特定のものに固定してしまうと、ユーザの演奏表情に対するとらえ方を反映した表情付けを行うことができない。本研究では、類似事例探索のための類似関数および各特徴量の重み値を、ユーザがGUI上で編集できるようにする。これにより、自分のとらえ方にあった類似尺度を試行錯誤しながら探すことができ、自分の意図通りの演奏表情付けがしやすくなると期待される。

### 3. 演奏表情付けアルゴリズムの設計

本章では、前章の議論に基づいて設計した演奏表情付けアルゴリズムについて述べる。

#### 3.1 アルゴリズムの概要

本手法では、ギターソロの単旋律の楽譜を入力とし、それにピッチの揺らぎ(ピッチベンド)を付与したMIDIデータを出力する。個々の音符を処理単位とし、音符ごとに条件を満

たす事例を事例データベースから見付け出し、その事例に付与されているピッチベンドを対象楽曲に付与する。具体的には、対象楽曲のある1音符を $N_t$ とすると、各 $t$ に対して、対象音符 $N_t$ の近傍 $N_{t-k}, \dots, N_{t+k}$ の特徴が近い旋律断片を探し出す。その際、扱う特徴量としてギターソロの単旋律の楽譜に記述されている情報と後述する運指推定アルゴリズム(詳細は4章)を用いて推定した運指情報を用いる。対象音符を含む旋律断片と最も符合する旋律断片を得た後、それに付与されているピッチベンド情報を対象楽曲に付与することで演奏表情付けを行う。特徴が近い旋律断片が見つからなかった場合は考慮する近傍の範囲 $k$ を小さくし、複数見つかった場合は $k$ の値を大きくして、事例が1つだけ見つかるまで事例探索を繰り返す。

#### 3.2 特徴抽出

対象楽曲の各音符 $N_t$ に対して、

- 近傍 $k+1$ 個の音符 $N_{t-k}, \dots, N_{t+k}$ の相対音高、ベロシティ、音長、運指情報(ただし、相対音高は $N_t$ の音高(ノートナンバー)に対する相対値、運指情報は次章で述べる運指推定アルゴリズムによって推定された演奏に用いられるギターの弦番号)
- ピッチベンドの時系列データ
- 用いられている奏法名のタグ情報

を抽出する。

#### 3.3 類似度の定義

入力楽曲中の音符列 $N_{t-k}, \dots, N_{t+k}$ と特徴が近い音符列を探索するため、音符列間の類似性を評価する数値である類似ランクを定義する。類似ランク $R_t$ は、相対音高に対する類似度 $L_1$ 、音長に対する類似度 $L_2$ 、運指情報に対する類似度の $L_3$ の重み付き線型和

$$R_t = \sum_{n=1}^3 w_n L_n$$

で定義する。ここで、重み係数 $w_1, w_2, w_3$ は類似事例探索の際に各特徴量をどの程度重視するかを表すパラメータである。例えば $(w_1, w_2, w_3) = (1.0, 0.0, 1.0)$ という設定にすると、相対音高と運指情報を同程度に重要視し、音長を無視した類似事例探索を行うことができる。

事例データベースから選ばれた音(選択音という)の相対音高 $P_t$ に対する類似度 $L_1$ は、抽出対象音の相対音高 $P'_t$ と $k+1$ 個の近傍音の相対音高、類似関数 $f(x)$ (初期設定では $f(x) = \text{sech}(x)$ )、関数制御定数 $\alpha_1$ を用いて、

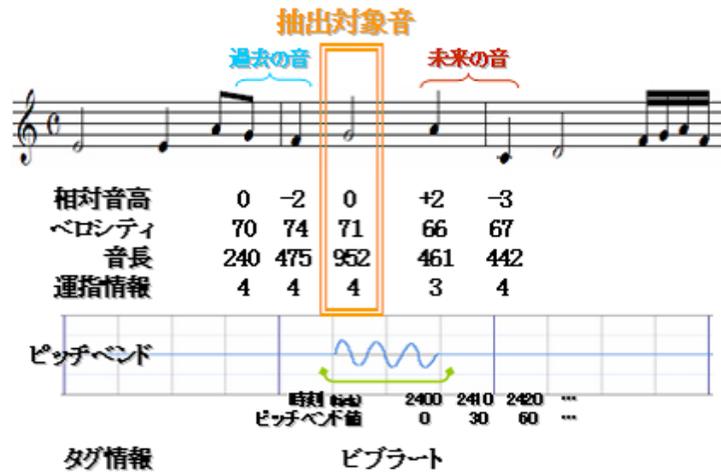


図1 特徴抽出の例

$$L_1 = \prod_{n=-k}^k f\left(\frac{P_{t+n} - P'_{t+n}}{\alpha_1}\right)$$

と表される。

同様に、選択音の音長  $G_t$  に対する類似度  $L_2$  は、抽出対象音の音長  $G'_t$  と  $k+1$  個の近傍音の音長、前述の類似関数  $f(x)$ 、関数制御定数  $\alpha_2$  を用いて、

$$L_2 = \prod_{n=-k}^k f\left(\frac{G_{t+n} - G'_{t+n}}{\alpha_2}\right)$$

と表される。

同様に、選択音の運指情報  $F_t$  に対する類似度  $L_3$  は、抽出対象音の運指情報  $F'_t$  と  $k+1$  個の近傍音の運指情報、前述の類似関数  $f(x)$ 、関数制御定数  $\alpha_3$  を用いて、

$$L_3 = \prod_{n=-k}^k f\left(\frac{F_{t+n} - F'_{t+n}}{\alpha_3}\right)$$

と表される。

現在の実装では  $(\alpha_1, \alpha_2, \alpha_3) = (1, 60, 1)$  としている。各関数制御定数の値を変更することで、類似度の計算に対してユーザの類似尺度がどの程度影響を与えるかを調整することができる。

### 3.4 類似事例の探索

前節で定義した類似ランクを用いて類似事例を探索する手法について述べる。今、 $t$  番目の音符  $N_t$  に対して演奏表情付けを行うとすると、類似事例は次のアルゴリズムによって探索される。

- (1) 近傍音の音符数  $k$  を初期値にする。
- (2) 事例データベースの各音符に対して  $N_t$  との類似度を 3.3 節で定義した類似ランクを使って求め、類似ランクが高いものから 3 つ選び出す (ただし、類似ランクが事前に設定した閾値を下回るものは除外する。)
- (3) (2) で選んだものが全て閾値処理によって除外された場合、 $k$  の値を 1 減じて以上の処理を再び行う。
- (4) (2) で選んだものの中で類似ランクが等しいものがある場合、 $k$  の値を 1 足して以上の処理を再び行う。

類似事例を探索する際、閾値を用いずに単純に類似ランクが高いものを選ぶと、類似性がないと判断すべき類似ランクが低い事例であるにもかかわらず類似事例として選択されてしまう可能性がある。これを回避するために、適切な閾値を与えることにより類似性の高い事例の演奏表情が転写されるようにしている。

### 3.5 特徴転写

前節の方法により、各音符に対して転写元の事例が決まれば、転写元の MIDI データに付与されているピッチベンド情報とベロシティ情報を対象音符に転写する。

類似ランクに基づく特徴転写を行う際に転写元の MIDI データと転写先の MIDI データの音長が異なる場合がある。すると、ピッチベンドといった時々刻々と変化するパラメータは音長に比例して長さを調節しなければならない。本システムは、転写元の MIDI データの音長とピッチベンドの時間変化量との比率が変わらないように転写する。

ベロシティ変化を付与する際、転写先の MIDI データの対象音のベロシティ  $V'_t$  は、転写元の MIDI データの対象音のベロシティ  $V_t$  を用いて、

$$V'_t = \frac{V_t}{V_{t-1}} V'_{t-1}$$

とする。ただし  $(V_0, V'_0) = (100, 100)$  とする。このようにすることで、転写先の MIDI

データに演奏表情付けが行われる際、転写元の MIDI データのペロシティの変化率が転写先の MIDI データのペロシティに反映される。

#### 4. 運指推定アルゴリズムの設計

本章では、前章で述べた演奏表情付けアルゴリズムの中で用いる運指推定アルゴリズムについて述べる。

##### 4.1 運指推定の困難性

ピアノなどとは違い、任意の旋律をギターで演奏する際には複数の押弦位置の可能性が存在するため、一意に押弦位置を決定することができない。運指を推定するシステムとして、市販されている幾つかのシステムや三浦らによる“Score-to-Tablature (S2T)<sup>[3]</sup>”、伊藤らの研究<sup>[4]</sup>などがあるが、以下のような点が考慮されていないことが挙げられる。

- 押弦する弦やフレットの位置によって異なる指への負荷が考慮されていない。
- ネックの分厚さやピックアップに近いほどフレットの間隔が小さくなるなどのギターの物理的特性が無視されている。
- ユーザが押弦を多用したい弦やフレットの位置を用いるような運指が得られない。

本研究では、ギタリストが習熟する過程で自然と習得するヒューリスティックな運指に関するこれらの要素を以下のようにしてアルゴリズムに組み込むことで、ユーザそれぞれに適した運指の推定を行う。

- 押弦する弦やフレットの位置に対して負荷係数を設定し、どの程度指に負荷を与えるかをパラメータ化する。
- ギターの物理的特性に基づいた運指最適化問題の定式化を行う。
- ユーザが押弦を多用したい弦やフレットの位置をパラメータとして与え、その値を反映させた運指推定を行う。

##### 4.2 運指最適化問題の定式化

運指には複数の可能性が存在するが、全てのギタリストが同じ運指を行うわけではないため、ギタリスト毎に最適な運指というものとは異なってくる。そこで、ギタリストが押弦を多用したい弦やフレットの位置を考慮することでギタリスト毎に適した運指を生成することを考える。本手法では、ユーザの運指に対する意図を考慮したなるべく負担の掛からない運指を最適な運指と定義する。運指を推定する際に考慮すべき要素として、左手首の移動量とポジション、左手の指の移動量、演奏に使用する弦の3つが挙げられる。

ギターソロの運指の最適化問題は、ある音から次音へ遷移する際の運指を段階とする多段

決定問題として捉えることができる。本研究では、伊藤らの研究<sup>[4]</sup>を参考に、多段決定問題を解くために動的計画法を用いて問題を解決する。動的計画法のマルコフ性を保つために、指のポジションが過去の履歴に依存するセーハやコードは本手法の適用対象外とする。また、手のポジションが過去の履歴に依存する開放弦は後述する変則 Viterbi アルゴリズムを採用することで考慮することを可能にする。

動的計画法が適用できるように運指最適化問題を定式化する。まず、目的関数を以下のように定義する。

$$\text{Minimize } \sum_{k=1}^{N-1} C(X_k, X_{k+1})$$

$$\text{subject to } C(X_k, X_{k+1}) \geq 0 \quad (k \in \mathbb{N})$$

この目的関数は、 $N$  個の音符を演奏する際の各音符間での手首や指の移動コスト（実際の移動距離）の総和を最小化することを表す。 $C(X_k, X_{k+1})$  は、 $k$  番目の音符を演奏状態  $X_k$  で発音させてから  $k+1$  番目の音符を演奏状態  $X_{k+1}$  で発音させるまでに要する移動距離に相当する。 $X_k$  は各音符の演奏状態を表し、使用する弦番号、フレット番号、指番号の組み合わせから構成される。ここでいう指番号はそれぞれ人差し指が 1、中指が 2、薬指が 3、小指が 4 に対応している。

次に、移動コスト  $C(X_k, X_{k+1})$  を定式化する。運指の動作は離弦、ポジション移動、押弦という 3 つのステップの繰り返しと考えられる。すると、求める移動コスト  $C(X_k, X_{k+1})$  は各ステップの移動コストの和と定義できる。離弦に要する移動コストを  $C_1(X_k, X_{k+1})$ 、ポジション移動に要する移動コストを  $C_2(X_k, X_{k+1})$ 、押弦に要する移動コストを  $C_3(X_k, X_{k+1})$ 、各ステップに対する負荷係数を  $W_j$  とすると、

$$C(X_k, X_{k+1}) = \sum_{j=1}^3 W_j C_j(X_k, X_{k+1})$$

となる。ここで、各ステップの移動コストについて定式化を行う。

離弦に要する移動コスト  $C_1(X_k, X_{k+1})$  は、押弦状態から指の自然なポジションに戻るまでの距離で表される。よって、求める移動コスト  $C_1(X_k, X_{k+1})$  は、指番号  $n$  の指の自然なポジションが  $F(n) = 18 - 4n$  に従うとすると、 $X_k$  での弦番号  $s_k$ 、 $X_k$  での指番号  $n_k$ 、指板半径  $r$ 、指板の幅  $w$  を用いて、以下の式で記述される。

$$C_1(X_k, X_{k+1}) = |(r + F(n_k)) - r \sin \theta|$$

$$\theta = \frac{\pi}{2} + \frac{(7-2s_k)}{10} \left( \arccos \left( 1 - \left( \frac{w}{\sqrt{2}r} \right)^2 \right) - \frac{6}{r} \right)$$

ポジション移動に要する移動コスト  $C_2(X_k, X_{k+1})$  は、離弦したポジションから次のポジションへ移動するまでの距離で表される．よって、求める移動コスト  $C_2(X_k, X_{k+1})$  は、 $X_k$  でのフレット番号  $f_k$ 、 $X_k$  での指番号  $n_k$ 、 $X_{k+1}$  でのフレット番号  $f_{k+1}$ 、 $X_{k+1}$  での指番号  $n_{k+1}$ 、指板のサイズ  $L$  を用いて、以下の式で記述される．

$$C_2(X_k, X_{k+1}) = \left| \frac{L}{\sqrt{2}f_{k+1}-n_{k+1}+1} - \frac{L}{\sqrt{2}f_k-n_k+1} \right|$$

押弦に要する移動コスト  $C_3(X_k, X_{k+1})$  は、前述の指の自然なポジション  $F(n)$  から押弦するまでの距離で表される．よって、求める移動コスト  $C_3(X_k, X_{k+1})$  は、 $X_k$  での弦番号  $s_k$ 、 $X_{k+1}$  での弦番号  $s_{k+1}$ 、 $X_{k+1}$  での指番号  $n_{k+1}$ 、指板半径  $r$ 、指板の幅  $w$  を用いて、以下の式で記述される．

$$C_3(X_k, X_{k+1}) = \sqrt{(r \cos \theta_1 - r \cos \theta_2)^2 + \{r \sin \theta_1 - (r + F(n_{k+1}))\}^2}$$

$$\theta_d = \frac{\pi}{2} + \frac{(7-2s_{k-d+2})}{10} \left( \arccos \left( 1 - \left( \frac{w}{\sqrt{2}r} \right)^2 \right) - \frac{6}{r} \right)$$

負荷係数  $W_1, W_2, W_3$  を適切に与えることにより、ユーザが押弦したい弦やフレットの位置を重視した運指を推定することが可能である．例えば  $(W_1, W_2, W_3) = (1.0, 5.0, 1.0)$  のようにポジション移動に要する移動コストの負荷係数  $W_2$  の値を大きくすれば、左手首の移動量が少ない運指を推定結果として得ることができる．

#### 4.3 変則 Viterbi アルゴリズム

第  $n$  段階の音符の演奏状態  $X_n$  に至るまでの最小コストを  $f(X_n)$  とすると、 $n = 1, 2, \dots, N-1$  に対して、 $f(X_{n+1})$  は以下の式で記述される．

$$f(X_{n+1}) = \begin{cases} \min_{X_{n-1}} [f(X_{n-1}) + C(X_{n-1}, X_{n+1})] & \text{if } X_n = X_{open} \\ 0 & \text{if } X_{n+1} = X_{open} \\ \min_{X_n} [f(X_n) + C(X_n, X_{n+1})] & \text{otherwise} \end{cases}$$

ただし、 $X_{open}$  は開放弦で演奏する状態を表す．また、初期条件として  $f(X_1) = 0$ 、

$X_1 \neq X_{open}$  を与える．

基本的に押弦位置は直前の手のポジションにのみ依存するため、目的関数を最小化するアルゴリズムとして Viterbi アルゴリズムを採用すれば最適な運指が求められる．しかし、開放弦を使用する場合、開放弦で演奏した音の次の音のポジションは開放弦で演奏した音より前の音のポジションに依存することになるのでマルコフ性は失われる．これに対応するため、現在の演奏状態が開放弦である場合に限り過去に開放弦ではない音の演奏状態を考慮して再帰的に移動コストを計算することにより、開放弦を考慮した最適な運指推定を実施する．

## 5. 実装・実験

3章および4章で述べた演奏表情付けアルゴリズムと運指推定アルゴリズムに基づいて実装したシステムおよび性能を確認するために実施した予備実験について報告する．

### 5.1 システムの実装

本システムでは、様々な演奏表情付けを実現するために以下のパラメータを操作できるようなインタフェースを提供している（図2）．

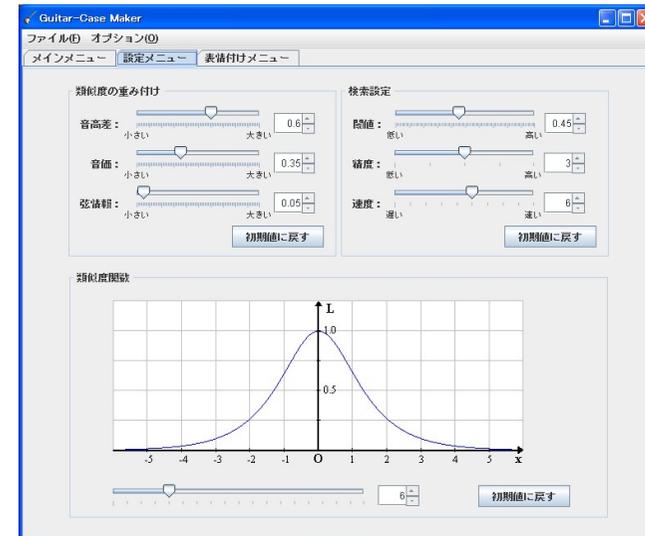


図2 Guitar-Case Maker の GUI 画面

- 類似度の重み付け
  - 類似ランクの重み係数  $w_1$  類似ランクを計算する際の相対音高情報に対する重み
  - 類似ランクの重み係数  $w_2$  類似ランクを計算する際の音長情報に対する重み
  - 類似ランクの重み係数  $w_3$  類似ランクを計算する際の運指情報に対する重み
- 探索設定
  - 探索アルゴリズムの閾値 転写候補の類似ランクの下限
  - 探索アルゴリズムの精度 近傍音の音符数  $k$  の初期値
  - 探索アルゴリズムの速度 探索アルゴリズムの試行回数の制限
- 類似関数  $f(x)$  (但し, 設定可能範囲は  $2 \leq x \leq 20, 0 \leq f(x) \leq 1$ )  
操作できるパラメータは大きく類似度の重み付け, 探索設定, 類似関数  $f(x)$  の設定の 3 つに分けられ, それぞれ独立にシステムの設定を変更することができる.

### 5.2 事例データベースの構築

システムの動作確認をするために 3 つのデータベースを用意した。データベースを構成する事例データにギタリストの演奏を再現した MIDI データを用いた。本研究では, 選ばれた類似事例の違いによる演奏表情付けの生成結果の違いを考察しやすくするため, B.B. キング, ブライアン・メイ, ラリー・カールトンの 3 名のギタリストを取り上げ, 各ギタリストの演奏の特徴をできるだけ高精度に模倣した MIDI データからなる事例データベースを構築した。この MIDI データは, 市販されているものや WWW 上で配布されているものから, 当該ギタリストの特徴がよく表現されているものを注意深く選定して構成した。その結果, B.B. キングは 18 曲, ブライアン・メイは 66 曲, ラリー・カールトンは 10 曲になった。

### 5.3 演奏表情付けアルゴリズムに関する予備実験

類似関数を編集することで様々な演奏表情付けが行われているかを確認するために予備実験を行う。同一の旋律に対して異なる類似関数を設定することで生成される演奏表情付けが異なっているのかを検証する。

実験によって生成されたデータの一部を図 3 に示す。図 3 は, ブライアン・メイのデータベースを用い, レッド・ツェッペリンの楽曲「Stairway To Heaven」のギターソロを入力楽曲とした場合の演奏表情付けの生成結果である。アルゴリズムのパラメータの設定は類似ランクの重み係数が  $(w_1, w_2, w_3) = (0.6, 0.35, 0.5)$ , 探索アルゴリズムの閾値が 0.8 (やや高い), 精度は 5 (高い), 速度は 1 (遅い) となるように設定した (a) と (c) は類似関数を初期設定のものとした場合の類似関数の概形 (b) は類似関数を (a) のものとした場合の演奏表情の生成結果 (c) は類似関数を編集した場合の類似関数の概形 (d) は類

似関数を (c) のものとした場合の演奏表情の生成結果である (b) と (d) の図は, 上から順にノートナンバー, ピッチバンド, ペロシティ, TAB 譜 (運指情報) の 4 段形式で表示されており, 横軸は時間を表す。ノートナンバーは演奏表情付けが行われている箇所はピンク色で表示され, それ以外の箇所は青く表示されている (b) と (d) を比較すると, 特殊奏法の挿入箇所や探索されている事例, ペロシティ変化が異なることがわかる。特に (d) は, 類似関数以外同じ設定であるにもかかわらず (b) より多くの演奏表情付けが行われており, 特殊奏法も多く挿入されているのが確認できる。これは (b) の類似関数より (d) の類似関数の設定の方が演奏表情の類似性に対する条件が寛容であるためであり, 類似関数の設定によって異なる類似尺度が表現できているのが確認できる。他の実験データにおいても同様の結果が得られたため, 類似関数によって異なる類似尺度が定義され, 様々な演奏表情が生成可能になったと言える。

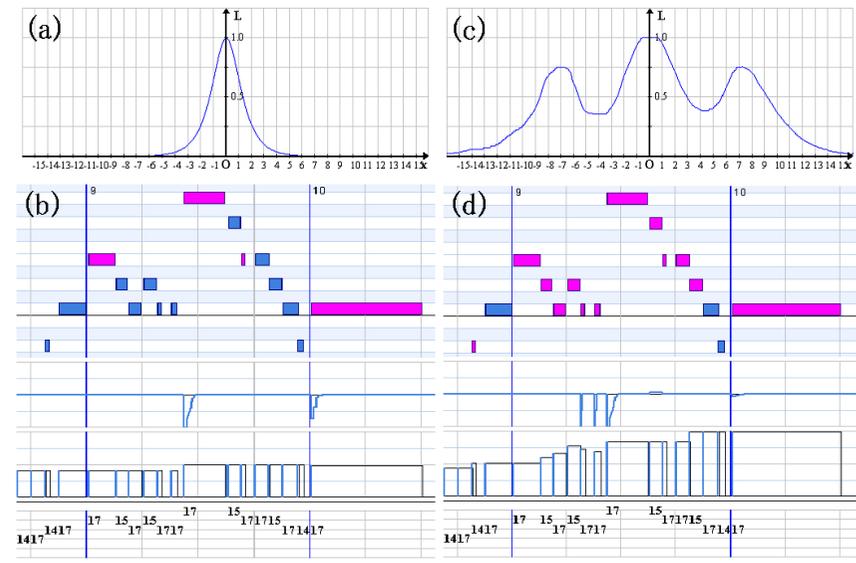


図 3 演奏表情付けアルゴリズムの生成結果の一例

### 5.4 運指推定アルゴリズムに関する予備実験

演奏表情付けアルゴリズムに運指推定アルゴリズムで推定した運指情報を導入したこと

で演奏表情の生成結果に変化が見られるかを確認するために予備実験を行う。同一の旋律に対して異なる類似ランクの重み係数を設定することで生成される演奏表情付けが異なっているのかを検証する。

実験によって生成されたデータの一部を図4に示す。図4は、ラリー・カールトンのデータベースを用い、クイーンの楽曲「Killer Queen」のギターソロを入力楽曲とした場合の演奏表情付けの生成結果である。探索アルゴリズムの閾値が0.5（普通）、精度は5（高い）、速度は1（遅い）となるように設定した（a）は類似ランクの重み係数を $(w_1, w_2, w_3) = (1.0, 1.0, 0.0)$ と設定することで、類似事例探索に運指情報を用いないような場合の演奏表情の生成結果である。また（b）は類似ランクの重み係数を $(w_1, w_2, w_3) = (1.0, 1.0, 1.0)$ と設定することで、類似事例探索に運指情報を用いるようにした場合の演奏表情の生成結果である。図は前節と同じ4段形式で表示されている（b）は（a）では見られない演奏表情が付与されており、図の3音目の演奏表情は（a）の同じ箇所の演奏表情より適切なものが付与されている。他の実験データにおいても同様の結果が得られた。

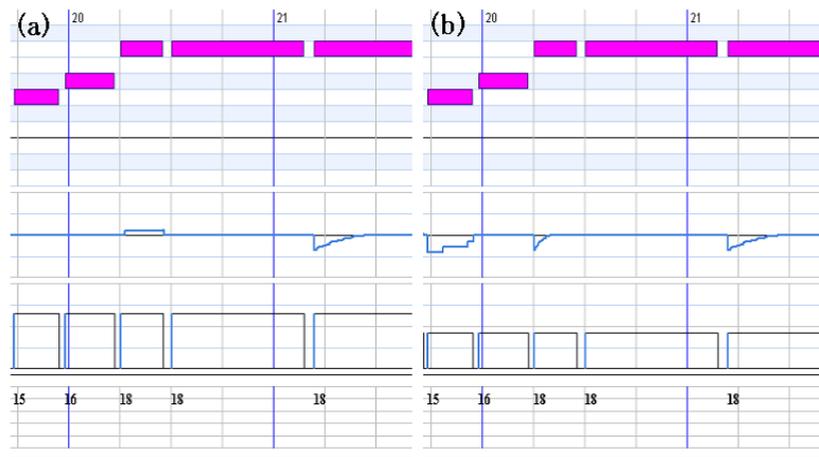


図4 運指推定アルゴリズムの推定結果の一例

## 6. まとめ

本稿では、エレキギターに対する演奏表情付けシステムの開発を目指し、それを実現する

ために演奏表情付けアルゴリズムと運指推定アルゴリズムを提案した。演奏表情付けに運指情報を導入したことにより、ピッチの遷移やベロシティの変化だけではとらえきれない演奏表情事例を得ることができた。また、提案した2つのアルゴリズムの各パラメータを調節することで、ユーザ間で異なる演奏表情に対する類似尺度を反映した様々な演奏表情付けが可能となった。今後は、提案アルゴリズムの有効性の検証する実験とともに、インタフェースのユーザビリティと生成された演奏表情の妥当性の検証に向けて、被験者を募っての評価実験を行う予定である。

## 参考文献

- 1) Alexis Kirke, Eduardo Reck Miranda: A Survey of Computer Systems for Expressive Music Performance, ACM Computing Surveys, Vol.42, No.1, pp.1-41 (2009)
- 2) 星芝貴行, 堀口進: 複数の演奏データを用いた標準的演奏の導出, 日本音響学会音楽音響研究会資料, MA96-13, pp.31-38 (1996)
- 3) 水谷哲也, 田中崇之, 五十嵐滋, 構造機能に基づく演奏法則と演奏創造, 第15回人工知能学会全国大会論文集, Vol.2, pp.1-4 (2003)
- 4) 小田安彦, 白川健一, 村上豊, 梶川嘉延, 野村康雄: 演奏者情報加味したピアノの自動演奏システムの構築～ニューラルネットワークによる局所部分に於ける演奏特徴の抽出～, 情報処理学会研究報告, Vol.1995, No.46, pp.7-12 (1995)
- 5) 木野弘樹, 梶川嘉延, 野村康雄: 特定ピアノ演奏者の特徴抽出による自動演奏システムの構築, 全国大会講演論文集, 第59回平成11年後期, No.2, pp.13-14 (1999)
- 6) 森口徳崇, 山梶雄一郎, 森田慎也, 三浦雅展: ピアノ演奏を対象とした感情付与システムの基礎的検討, 日本音響学会音楽音響研究会資料, MA2008-2, pp.7-12 (2008)
- 7) Josep Lluís Arcos, Ramon Lopez de Mantaras, Xavier Serra: SaxEx: a case-based reasoning system for generating expressive musical performances, International Computer Music Conference, pp.329-336 (1997)
- 8) Satoshi Shimizu, Saori Yoshinaga, Koichi Furukawa: Feature Extraction of Violinists by Rough Sets, The 19th Annual Conference of the Japanese Society for Artificial Intelligence (2005)
- 9) 中西正洋, 門田暁人, 松本健一, 井上克郎: ギターの演奏情報の抽出と分析, 全国大会講演論文集, 第59回平成11年後期, No.2, pp.9-10 (1999)
- 10) 水本直希, 北原鉄朗, 片寄晴弘: 事例データに基づくエレキギターの表情付けシステム「Guitar-Case Maker」, 情報処理学会研究報告, 2010-MUS-87, No.4, pp.1-6 (2010)
- 11) Mitsuyo Hashida, Noriko Nagata, Haruhiro Katayose: jPop-E: An Assistant System for Performance Rendering of Ensemble Music, NIME, pp.313-316 (2007)
- 12) 鈴木泰山, 宮本朋範, 西田深志, 徳永健伸, 田中穂穂: Kagurame Phase I 事例ベースの演奏表情生成システム, 情報処理学会研究報告, Vol.1998, No.14, pp.61-68 (1998)
- 13) 三浦雅展, 廣田勲, 浜将彦, 柳田益造: 単旋律ギター演奏における最適押弦位置決定システムの構築, 電子情報通信学会論文誌, J86-D-II, No.6, pp.755-763 (2003)
- 14) 伊藤雅, 林田巧: 単旋律におけるギター運指の最適化, 電気学会論文誌.C, 電子・情報・システム部門誌, Vol.124, No.7, pp.1396-1403 (2004)