

## 実演奏における演奏表情情報の抽出と適用

中川 渉<sup>†</sup> 高嶋 章雄<sup>†</sup> 山本 恭裕<sup>†</sup> 蔵川 圭<sup>†</sup> 中小路 久美代<sup>†\*</sup>  
e-mail: {wataru-n, akio-ta, yasuhi-y, kurakawa, kumiyo}@is.aist-nara.ac.jp

<sup>†</sup> 奈良先端科学技術大学院大学 情報科学研究科  
<sup>‡</sup> (株) SRA ソフトウェア工学研究所  
\* 科学技術振興事業団 さきがけ研究 2 1

**概要** 計算機を用いた音楽演奏データ制作における演奏の表情付けは専門性と労力とを要する作業である。本稿では、実演奏における演奏表情情報を自動的に抽出、他パートに適用することによってその負荷を軽減し音楽制作の支援を行う手法について述べ、それに基づく支援システム CAPADY (a Composition Assistance by Producing Agogics and Dynamics) について説明する。

## The Extraction and Application of Expressive Features Based on Human Performance

Wataru Nakagawa<sup>†</sup> Akio Takashima<sup>†</sup> Yasuhiro Yamamoto<sup>†</sup>  
Kei Kurakawa<sup>†</sup> Kumiyo Nakakoji<sup>†\*</sup>

<sup>†</sup> Graduate School of Information Science, Nara Institute of Science and Technology  
<sup>‡</sup> Software Engineering Laboratory, Software Research Associates, Inc.  
\* PREST, Japan Science and Technology Corporation

**Abstract** This paper presents an approach to support the production of an expressive musical prototype. With this approach expressive features are automatically extracted from a piece of human musical performance and applied to produce agogic and dynamic accents for another piece of musical notation in accordance with the original musical performance. CAPADY (a Composition Assistance by Producing Agogics and Dynamics) has been built based on the approach to help a musical composer construct music prototypes using multiple timbres.

### 1 はじめに

音楽制作に計算機を用いる手法が普及しつつある。作曲者自身が楽曲に対する理解を深めたり、共同作曲者間で理解を共有したりする目的で計算機上に音楽演奏データを制作する機会が多くある。本稿では特に軽音楽を対象に、上記の目的で音楽演奏データを制作する場合の支援を考察する。

音楽演奏データの制作には主として、

1. 五線譜やそれに代わる図形をポインティングデバイス等を用いて編集することによって音楽演奏データを入力する。

2. 電子楽器を実際に演奏し、音楽演奏データとして記録する。

という二つの方法がある。

図 1 は計算機に入力された音楽演奏データの一例である。図 1(a) は 1 の方法、すなわち楽譜を編集する形態で入力したものを、図 1(b) は 2 の方法で入力したもの、すなわち実際に人間が行った演奏を音楽演奏データとして記録したものを示す。これらのデータにおいては 1 行が 1 つのイベントを表している。ここで言うイベントとはある音の発音と消音、強さの変更といった演奏のための処理の一手順を指すものである。

図 1において、最も左の列の数値はイベントの発生時刻 [ms] を示したものであり、その次の列の数値は発音、消音といったイベントの種類を示している。この場合の数値 144 はこのイベントが「音の発生」であることを表している。144 はあとに続く二つの列の数を引数としてとり、それぞれ音の高さ、強さを表している。

図 1(a) に示すように、1 の方法で制作した音楽演奏データでは、発音時刻 (1 列目データ) や音の強さ (4 列目データ) などが厳密に規定される。そのため聞き手に与える印象は計算機特有のものとなり、人間が実際に演奏することによって与える印象とは異なる場合が多い。このため作曲者が、音楽演奏データに人間が行う演奏のように自然な印象を求める場合やテンポやリズムの微妙な表現を行う場合には、まず五線譜等を編集することによって (a) のような音楽演奏データを制作し、その後 (a) の数値データを直接編集することによって (b) のような音楽演奏データに改良するという作業を行う必要がある。これら一連の作業を「演奏表情の付加」と呼ぶ。多くの場合、演奏表情の付加は直接数値を変更することによって行われる。

図 1(a) と (b) の数値の違いによって表現されるような演奏の違い、すなわち音の発生時刻や強さといった演奏表情の与え方の違いは、実際に演奏を行う際に演奏者の意識にのぼることはまれであり、どのような意図に基づいて演奏表情が生成されたかが不明確である場合が多い。その一方で、演奏表情の与え方によって音楽演奏データから受ける印象が大きく変わってしまう場合もある。このような数値の違いとしての演奏表情の与え方の違いを明確に認識しながら、意図する音楽演奏データへと調整する作業は高い専門性と労力を要する作業である。

本稿では、五線譜等を編集することで制作した音楽演奏データに作曲者の意図を反映した演奏表情を付加する方法の一つとして、作曲者自身が実際に演奏を行った別のパートから抽出した演奏表情情報を、五線譜等を編集することで入力したパートに適用するという方法を提案する。

以下 2 章では提案する手法について詳述し、3

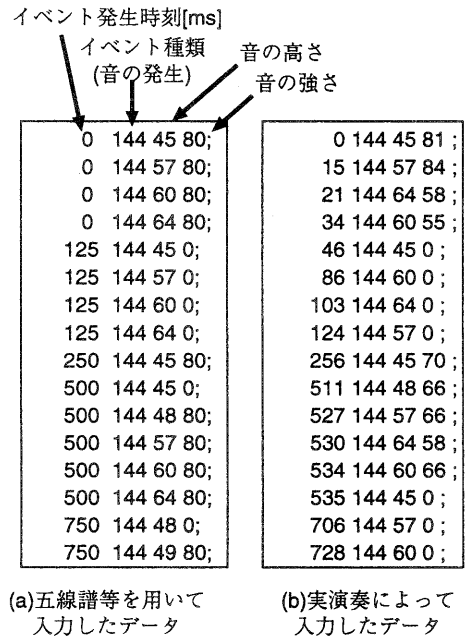


図 1: 音楽演奏データの例

章でそのアプローチに基づき構築されたプロトタイプシステム CAPADY について説明する。4 章で関連研究について述べ、5 章で本稿を結ぶ。

## 2 演奏表情を抽出・適用するアプローチ

本稿では、ユーザ自身があらかじめ五線譜等を用いて入力しておいたパートに対し、ユーザ自身が実際に楽器演奏を行ったパートから演奏表情情報を抽出しそれを適用するという手法を提案する。

### 2.1 システムの機能

図 2 にシステムの機能を示す。あらかじめパート A の音楽演奏データを五線譜等を用いて入力しておき、それに対して演奏表情を付加する作業をパート B の実演奏によって行う。

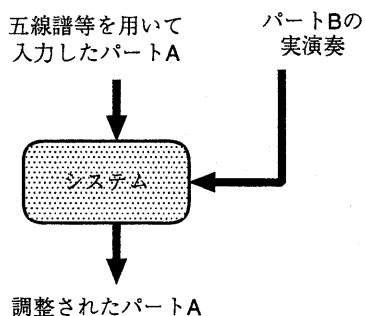


図 2: システムの機能

## 2.2 演奏表情情報

演奏表情の付加は、

1. パート B の実演奏から演奏表情情報を抽出しパート A に適用する。
2. パート A の楽器演奏について知られている知識を適用する。

という二つの段階に分けて行う (図 3)。

1 の段階では、

- 音毎の発音時刻
- 音毎の終了時刻
- 音毎の強さ

という三つの各パラメータ [1] についてパート B の実演奏から値を抽出しパート A に適用する。楽曲について作曲者がもつ演奏意図を音楽演奏データに反映させることが目的である。

2 の段階ではパート A の楽器演奏に関して既知の知識を適用する。パート A としてドラムを例にとれば、

- 左手で演奏される音は右手で演奏される音に比べ音の強さが弱い傾向がある。

といった知識を適用することによる表情付けの支援である。

以下、本稿ではこれら二つの段階のうち 1 の段階についての手法および実現法について論じる。

2 の段階については今後の課題とする。

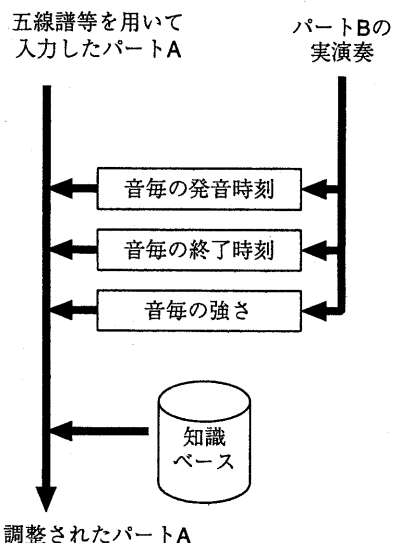


図 3: システムを実現するアプローチ

## 2.3 演奏表情情報の抽出と適用

以下では、実演奏から演奏表情情報を抽出し、他パートに適用する 1 の段階の部分について述べる。

### 2.3.1 パート間の対応関係に着目した演奏表情情報の抽出

実演奏を音楽演奏データとして記録したパート B ではイベントの発生時刻にばらつきがある。1 章で述べたように、本稿におけるイベントは、ある音の発音、消音、強さの変更といった演奏のための一つの手順に対応する。このばらつきのある時刻列を簡単な整数比に割り付ける処理を量子化と呼ぶ。パート B を量子化したものと、それにテンポを合わせたパート A を比較するとパート A のイベントは、

- パート B のイベントと同じタイミングで発生するもの

- パート B のイベントが発生しないところで発生するもの

という二種類に分けることができる。

前者のイベントに関してはパート B の対応するイベント (同じタイミングで発生するイベント) を用いて演奏表情情報を得ることができる。ここで、後者のイベントの発生時刻をどのように決定するかが問題となる。

本研究では図 4 に示すように、まずパート B に現れる最小音符 (ここでは 16 分音符) を発音処理のための一単位とする。そして、一旦すべての音符に対応する発音時刻列を関数近似によって求め、それをパート A に適用するというアプローチをとる。今回は簡単化のため、内挿するイベント発生時刻を等間隔とする線形補間を用いた。

具体的な手法は次の通りである。

### 1. 量子化幅の決定

パート B の実演奏から抽出した  $n$  番目の発音イベントの発生時刻を  $B_n$  ( $1 \leq n \leq N$ ) とする。その次にあたる発音イベントの発生時刻との時間間隔、すなわち  $B_{n+1} - B_n$  が、近似的に標準単位音符長  $s$  の何倍に相当するかによってグループ分けをおこなう。 $s$  の値として例えば単位音符を 16 分音符とすると  $s = 125$  とおくことができる。これは標準的なテンポである 120[beats/sec] では 16 分音符長が 125[ms] であるためである。グループ  $G^{(k)}$  は次のように定められる。

$$\{G^{(k)}\} = \bigcup_{j=1}^{N-1} \left\{ B_j \left\lfloor \left\lfloor \frac{B_{j+1} - B_j}{s} + 0.5 \right\rfloor \right\rfloor = k \right\}$$

次に、 $G^{(k)}$  の要素の平均値をとり、 $\langle G^{(k)} \rangle$  とする。 $\langle G^{(k)} \rangle / k$  を量子化幅  $q$  の候補とする。現状では、後述のプロトタイプシステムにおいては  $\langle G^{(1)} \rangle / 1$  の値を量子化幅  $q$  (テンポに相当) としている。今後は、 $\langle G^{(2)} \rangle / 2$ 、 $\langle G^{(3)} \rangle / 3$  などの値も考慮して、より妥当な量子化幅を設定できるようにする必要がある。

### 2. イベント発生時刻の内挿

この量子化幅に基づいて  $B_n$  を量子化したものを  $B_n^q$  とすると、 $(B_{n+1}^q - B_n^q) / q$  は  $n$  番目の発音イベントとその次にある発音イベントの間に単位音符を内挿した結果として得ることができる発音イベントの回数を示している。つまり  $B_{n+1} - B_n$  間に  $(B_{n+1}^q - B_n^q) / q$  回発音イベントがくるように単位音符内挿を行う。したがって内挿された発音イベント一回の時刻間隔は  $\frac{B_{n+1} - B_n}{(B_{n+1}^q - B_n^q) / q}$  となる。そこでこの値を  $B_n$  に  $(B_{n+1}^q - B_n^q) / q$  回、順次加えて行くことで内挿を行う。すなわち、最小音符について発音時刻を内挿した時刻列  $\{B_m^i\}$  を考えれば

$$\{B_1^i, \dots, B_m^i\}$$

$$= \bigcup_{j=1}^{N-1} \left\{ B_j + \frac{B_{j+1} - B_j}{(B_{j+1}^q - B_j^q) / q} (k-1) \right.$$

$$\left. \left\{ k = 1, 2, \dots, \frac{B_{j+1}^q - B_j^q}{q} \right\} \right\}$$

と与えることができる。

上述の量子化幅決定の手法は、標準単位音符長  $s$  をどのように設定するかに依存する。そこで  $s$  の値をユーザの入力によって補正する機能を導入し、ユーザとシステムが協調してイベント発生時刻を決定できるようにする。

以上によって決定された  $B_m^i$  はすべての単位音符に対してのイベント発生時刻を示しているためパート A のすべてのイベント発生時刻が包含されることになる。

同様の方法で音の強さについても内挿を行った。後に述べるプロトタイプシステムではパート A としてドラムを例にとったため、終了時刻については考慮しないものとするが、他の楽器については今後検討する必要がある。

### 2.3.2 演奏表情の適用

パート B の実演奏を考慮することによるパート A の調整の度合をユーザが指定することができ

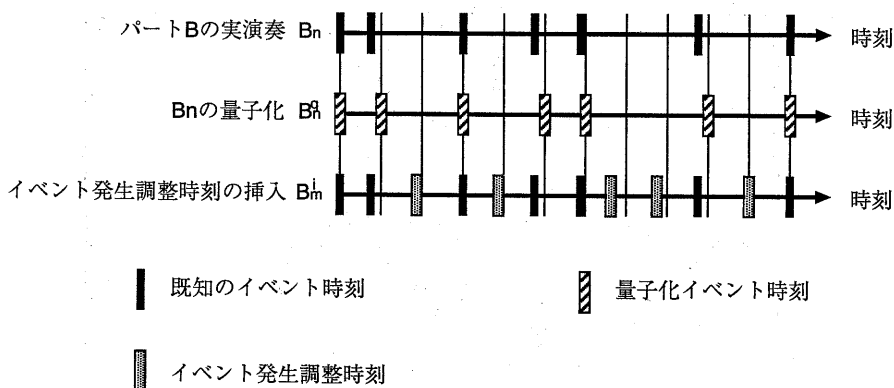


図 4: パートの違いによる音の有無の違いと発音時刻の調整

るようにする。上記の方法で求めた  $B_m^i$  をさらに量子化した  $B_m^{iq}$  は、実演奏から平均的なテンポを求め、そのテンポに基づいて一定間隔を保った時刻列であると考えることができる。そこでユーザが重み係数を入力し、それを基に  $B_m^i$  と  $B_m^{iq}$  の重みつき平均を求め、その重みつき平均をパート A に当てはめることとする。すなわちユーザの入力した重み係数を  $p (0 \leq p \leq 1)$  とすると、

$$B_m^{iw} = \frac{pB_m^i + (1-p)B_m^{iq}}{2}$$

によって定まる  $B_m^{iw}$  の対応する値をパート A のイベントの発生時刻に適用する。

### 3 CAPADY

CAPADY(a Composition Assistance by Producing Agogics and Dynamics) は上述のアプローチに基づいて構築したプロトタイプシステムである。音楽演奏データ制作上必要となる他パート（パート A）としてドラムを、ユーザが実演奏を行う楽器（パート B）としてキーボードをそれぞれ例にとりシステムの構築を行った。

#### 3.1 CAPADY の利用方法

本節ではプロトタイプシステム CAPADY の利用手順について述べる。

1. あらかじめ五線譜やそれに代わる図形等をポインティングデバイス等を用いて編集することで、ドラム演奏データを作成する。このデータを図 5 の A 部分のボタンを利用して読み込む。
2. あらかじめデータとして記録したキーボードの実演奏データを図 5 の B 部分のボタンを利用して読み込む。または、キーボードを実際に演奏し演奏内容を図 5 の B 部分のボタンを利用して記録する。作曲者はドラムに付加したい演奏表情をキーボード演奏によって表現する。
3. ドラムの発音時刻および音の強さについて、キーボードの実演奏の影響を受ける程度を示すパラメータをそれぞれ入力する。
4. パラメータ入力後、変換ボタンを押すとシステムはキーボード実演奏のテンポを計測し、作曲者に提示する。作曲者はここでテンポを訂正することができる。
5. 以上により、調整されたドラムデータを図 5 の C 部分で操作することができる。
6. 3, 4, 5 を複数回くり返し、複数種類の演奏表情情報を付加されたドラムデータを得



### 3.3 考察

本稿では抽出した演奏表情情報を直接適用したため、五線譜等を編集することで制作したパートは演奏表情を適用されることで実演奏に完全に追従する。

しかし実際には合わせなければならないポイントは外さず、また、多少のテンポのズレなどを含みながらも、音楽的に逸脱しない範囲で人間の演奏は行われる [1]。このため、完全に追従すると不自然な印象になる場合もある。

今後は合わせなければならないポイントとそうでないところに関して検討する必要がある。

## 4 関連研究

計算機を用いて、人間が行う演奏のような自然な演奏を実現しようとする点で、本研究は演奏の表情付けに関する研究ととらえることができる。

しかしながら表情付けに関する研究の多くは、演奏表情の生成そのものを対象としている。これに対し本研究の主眼はユーザの意図を演奏表情に反映する点にある。

たとえば鈴木らの研究 [4] [5] では、事例ベースに基づき演奏表情を生成する。演奏状況の与え方によって異なった演奏表情を生成することが可能である。ニューラルネットワークを用いるアプローチ [6] [7] [8] [9] [10] では、演奏者の個性に注目し、演奏者情報の与え方によって複数の演奏表情を生成することができる。実演奏から汎用的な演奏表情生成ルールを獲得する研究 [11] もあり、本研究と実演奏から演奏表情情報を抽出するという点で類似している。最近の商用ソフトウェアでは、テンポを微妙に変化させることでビート感を出すといった機能も実装されている [1]。

人間の行う独奏に追従するようにシステムが伴奏を出力するという研究も盛んに行われ、自動伴奏システムと呼ばれている [12]。独奏者の演奏を監視しながら伴奏パートのスケジューリン

グを行う。これらのアプローチでは、あらかじめ独奏者の演奏する楽譜と伴奏楽譜の両方が与えられていることを前提としている場合が多い。

人間の演奏意図を計算機に反映する研究としては演奏指揮システム [13] をあげることができる。指揮法を知識として持つことにより、計算機による自動演奏システムを身振りなどのジェスチャで制御しようとするものであり、演奏意図の考慮という点で本研究と関連が深い。

## 5 おわりに

本稿では、計算機上に音楽演奏データを制作する際、人間が行う演奏のような自然な音楽演奏データを制作する作業の難しさについて述べた。次いで、その作業を効果的に行うために他パートの実演奏から演奏表情情報を抽出、適用するというアプローチを提案し、そのアプローチに基づいて構築したプロトタイプシステム CAPADY を紹介した。

今後は 2.2 で述べた楽器演奏に関する知識の適用手法および本稿で述べたアプローチの有用性を評価するための実験手法について検討して行く予定である。

## 参考文献

- [1] 長尾 真, 宇津呂 武仁, 島津 明, 匂坂 芳典, 井口 征土, 片寄 晴弘, “4 文字と音の情報処理,” 岩波講座 マルチメディア情報学, 岩波書店, pp.163-219, 2000.
- [2] 片寄 晴弘, “特集 ビジュアル言語 MAX,” bit, Vol.30, No.1, pp.46-49, 1998.
- [3] Miller S. Puckette, “[Pure Data]: 新たな統合されたコンピュータ音楽環境,” 情報処理学会音楽情報科学研究会研究報告, 96-MUS-17-8, pp.37-41, 1996.
- [4] 鈴木 泰山, 徳永 健伸, 田中 穂積, “Kagurame Phase-I 事例ベースの演奏表情生成システム,” 情報処理学会音楽情報科学研究会研究報告, 98-MUS-24-11, pp.61-68, 1998.

- [5] 鈴木 泰山, 徳永 健伸, 田中 穂積, “事例ベースの演奏表情生成手法に関する研究,” 情報処理学会音楽情報科学研究会研究報告, 97-MUS-21-2, pp.7-12, 1997.
- [6] 坂本 崇, 藤井 敬三, 波多 正臣, 梶川 嘉延, 野村 康雄, “演奏者情報を加味したピアノの自動演奏システムの構築～曲の印象, 特徴による感性空間に関する検討～,” 日本音響学会講演論文集, 2-5-18, pp.643-644, 1996.
- [7] 白川 健一, 熊谷 俊行, 坂本 崇, 梶川 嘉延, 野村 康雄, “演奏者情報を加味したピアノの自動演奏システムの構築～曲の特徴を考慮したルールの生成～,” 日本音響学会講演論文集, 2-7-4, pp.667-668, 1996.
- [8] 小田 安彦, 村上 豊, 波多 正臣, 梶川 嘉延, 野村 康雄, “演奏者情報を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個人的揺らぎの生成～,” 日本音響学会講演論文集, 2-7-5, pp.669-670, 1996.
- [9] 白川 健一, 熊谷 俊行, 小田 安彦, 梶川 嘉延, 野村 康雄, “演奏者情報を加味したピアノの自動演奏システムの構築～エキスパートシステムを用いた演奏パラメータの生成～,” 日本音響学会講演論文集, 2-8-5, pp.619-620, 1995.
- [10] 小田 安彦, 白川 健一, 村上 豊, 梶川 嘉延, 野村 康雄, “演奏者情報を加味したピアノの自動演奏システムの構築～個人的引き癖を加味した人間的演奏の実現～,” 日本音響学会講演論文集, 2-8-6, pp.621-622, 1995.
- [11] 野池 賢二, 乾 伸雄, 野瀬 隆, 小谷 善行, “演奏情報と楽譜情報の対からの演奏表情規則の獲得とその応用,” 情報処理学会音楽情報科学研究会研究報告, 98-MUS-26-16, pp.109-114, 1998.
- [12] 浜中 雅俊, 築根 秀男, “ドラムスの生演奏にMIDI音源の再生音をリアルタイムでシンクロさせるシステムの開発,” 情報処理学会音楽情報科学研究会研究報告, 98-MUS-26-8, pp.53-60, 1998.
- [13] H.Morita, S.Ohteru, S.Hashimoto, “Computer Music System that Follows a Human Conductor,” IEEE Computer, Vol.24, No.7, pp.452-459, 1996.