

4分音符分の演奏に基づく演奏家の特徴抽出手法の提案

Feature Extraction Method based on A Quarter Note Performance.

川村 修*¹ 伊藤 孝行*¹ 新谷 虎松*¹
 Osamu KAWAMURA Takayuki ITO Toramatsu SHINTANI

*¹名古屋工業大学大学院 工学研究科 情報工学専攻
 Graduate School of Engineering, Nagoya Institute of Technology.

In this paper, we propose a method to extract musician's habits and preferences from her adlibs. And then, we implement a frame system for database. Adlib, a free play to a given song with musician's feelings, indicates her features. The method divides adlib performances into short performances equal to a quarter note span. The proposed method can extract musician's features.

1. はじめに

近年の家庭用の各種デジタルオーディオシステムの発達や、音楽ソフトのネットワーク配信が普及しつつあることによって、ユーザが求める音楽ソフトをいつでも容易に入手できることができるようになってきている。しかし、現在、膨大な音楽ソフトがあるとはいっても、最新の音楽や、デジタル編集された一部の音楽が提供されているのみで、多様化しているニーズの一部にしか対応しておらず、入手することができない音楽が多数を占めているといえる。以上のように入手できない楽曲の例として、アドリブ演奏が挙げられる。一般に、アドリブ演奏は演奏時に即興で行われるものであり、その場に居合わせた人間により楽しまれる。また、アドリブ演奏は、楽譜等にあらかじめ記録されている、決まりきった演奏ではなく、その瞬間にしか演奏されないフレーズを含んでいる。以上のようなアドリブ演奏自体は、記録として残らないことが多い。

これまでに、4分音符分の演奏事例に基づくアドリブ演奏模倣システム [1] に関する研究を行ってきたが、抽出された演奏事例の構造が理解しにくく、汎用性に欠けるという問題がある。本稿では、4分音符分の演奏に基づいた演奏家の特徴抽出手法を、より汎用性のあるものへと発展させる。本稿で提案する手法に関連する研究として、即興演奏をテーマにしている小谷 [2] の研究では、楽器エージェントが生成する演奏は、あらかじめ与えられた値に最も近いスケール上の音より構成される。この方法では、生成された演奏にスケール感はあるがコード感が反映されず、半音進行を考慮していない。また、音を決定する際のアルゴリズムに、ランダム関数を使用しているため、音の前後関係が考慮されず、機械的な演奏となってしまう。一方、音楽活動を支援する研究として、様々な研究が盛んに行われている。音楽に関する様々な要素を可視化することにより、旋律創作支援を試みる研究 [3] は、演奏家が行っている演奏の音楽的な要素を提示することにより、自身が行っている演奏を認知するのを手助けし、新たな演奏の方向性を決定するきっかけを与えている。本研究で提案する特徴抽出手法に基づくデータベースを用いることでより人間的なフレーズの生成を可能になる。また、フレーズ前後のつながりをデータベースから得た確率分布に従い決定することで、自然なつながりでフレーズを生成することができる。自ら演奏した旋律から、本手法を用いて抽出された特徴を参照することにより、自分の演奏

の特徴を認知するのを手助けし、今後の演奏方法を決定する判断基準を与えることもできる。

本稿では、第2章で、4分音符分の演奏について説明する。第3章では、演奏事例の構成について示す。第4章で、実験と評価を示す。最後に、第5章で、本稿をまとめる。

2. 4分音符分の演奏

2.1 演奏家の特徴を反映した旋律

アドリブ演奏における特徴を聴き手側からの観点と、演奏者側からの観点でまとめる。

- 聴き手が感じるアドリブ演奏の特徴
 - ・聴き慣れた旋律：演奏家が演奏した際に、聴き手がある特定の旋律がよく使われると感じる旋律や、聴いただけで演奏家が連想できてしまう旋律がある。これらの旋律には、演奏家の特徴が反映されていると言える。
- 演奏者が自覚するアドリブ演奏の特徴
 - ・好んで使う旋律：演奏家は、アドリブ演奏を行う際、これまでに行ってきたアドリブ演奏で、あるコードの時にある旋律を弾いたらよく聴こえたなどの過去の演奏を思い出し、現在の状況に合わせて使い分ける。演奏家により好みは異なり、好む旋律の使い方も異なる。よって、これらの旋律には演奏家の特徴が反映されていると言える。
 - ・手癖に基づく旋律：演奏家は、アドリブ演奏を行う際、自分が弾きやすいと思う旋律を無意識的にアドリブ演奏に入れていく。これを手癖に基づく旋律と呼ぶ。手癖に基づく旋律は、アドリブ演奏をする際に旋律が思い浮かばなかった時に、演奏されたり、好んで使う旋律などの、その他の旋律同士をつなぐ際に演奏される。好んで使う旋律が意識的に用いられるのに対して、手癖に基づく旋律は、アドリブ演奏に頻繁に用いられるので、演奏家の特徴が反映されていると言える。

2.2 4分音符分の演奏に基づく特徴抽出

旋律から演奏家の特徴を抽出する手法として、本研究では、旋律を4分音符分の演奏で分割し、4分音符分の演奏を取り出すことにより、特徴の抽出を行う。アドリブ演奏により作られた旋律を楽譜で表すと、音程の高低関係が直感的につかみやすくなる。図1にアドリブ演奏の一例を楽譜で表す。旋律を4分音符分の演奏で分割すると同様の形の演奏をつなぎ合わせた形となる。図2に旋律を4分音符分の演奏で分割した例を

連絡先: 466-8555 名古屋市昭和区御器所町名古屋工業大学
 大学院情報工学専攻 新谷研究室, TEL:(052)733-6550,
 {osamu, itota, tora} @ics.nitech.ac.jp



図 1: アドリブ演奏の一例

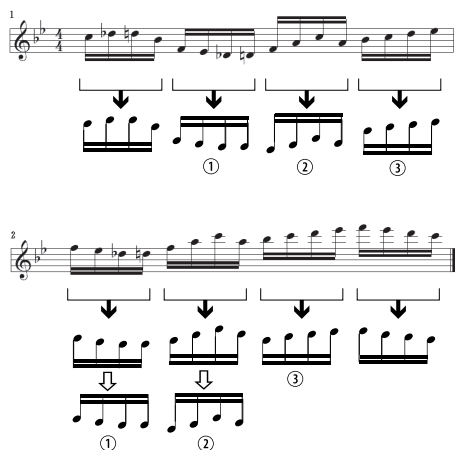


図 2: 旋律を4分音符分の演奏で分割

表す。図2では、黒い矢印で分割された旋律を示している。また、白い矢印は、分割された旋律を表す音符のぼう（符幹）とはた（符尾）の上下を入れ替えたものを示している。ぼう（符幹）とはた（符尾）の上下を入れ替えることにより、同様の形の演奏を判別しやすくなる。図2において、①の旋律が2回、②の旋律が2回、③の旋律が2回と、他2種類の旋律が1回ずつ用いられているのがわかる。以上のようにアドリブ演奏は、4分音符分の演奏をスケールとコードに合わせて音程の修正を行い、つなぎ合わせたものと考えることができる。この4分音符分の演奏は演奏家の特徴である、聴き慣れた旋律、好んで使う旋律、手癖に基づく旋律の3種類の旋律をそのまま用いたものとなっているので、当然、演奏家の特徴を反映している。

よって、本研究では、4分音符分の演奏を演奏家の特徴とし、アドリブ演奏を4分音符分の演奏で分割することにより特徴を抽出する。

3. 演奏事例の構成

3.1 データ構造

提案手法を用いて、4分音符分の演奏事例に基づくアドリブ演奏模倣システムを実装した。本システムにおける、演奏事例のデータ構造を示す。演奏事例は次の2つのデータにより構成される。

- ・単音データ (問題空間における状況を示すデータ)
 - ・旋律データ (単音データが示す状況に対して行われた演奏)
- 単音データは、曲中における4分音符分の時間に相当する状態についての詳細な情報である。また、旋律データは、4分音符分の演奏に基づく特徴抽出を用いて取り出された、演奏家のアドリブ演奏を示す。単音データは、次の3つのデータにより構成される。
- ・スケール・ルート (表1に示される値をとる)
 - ・スケール・タイプ (スケールの種類を示す文字列)
 - ・コード・ルート (表1に示される値をとる)

表 1: 音名と値の対応

音名	C	C #	D	...	B
値	1	2	3	...	12

- ・コード・タイプ (コードの種類を示す文字列)
 - ・音程 (MIDI ノートナンバーを用いて、0~127の自然数により示される)
 - ・音長 (4分音符分の演奏を示す値、480をとる)
- 一般的に、スケールは旋律の印象を決定するものであり、ルートとタイプを決定することにより、構成する音列が一意に決定される [5]。また、コードは、スケールで与えられた印象をより詳細に決定するものである。コードもスケールと同様、ルートとタイプを決定することにより一意に決定される [6]。旋律データは、以下に示すアドリブデータの列により構成される。アドリブデータは、次の2つのデータにより構成される。
- ・相対音程 (単音データ中の音程に対する音程差 (整数) で示される値)
 - ・音長 (4分音符を480とする自然数により表現される音長の列)
- 単音データの音程を N_1 、アドリブデータの対象となる音程を N_2 とすると、相対音程 N は、 $N = N_2 - N_1 + 1$ により決定される。

3.2 演奏事例の取得方法

4分音符分の演奏を演奏家の特徴として扱うためには、アドリブ演奏を行う前の曲と、アドリブ演奏を4分音符分の演奏で分割し、演奏事例として整形して格納する必要がある。以下、アドリブ演奏を行う前の曲とアドリブ演奏から演奏事例を取得する方法を示す。

【単音データの取得】

アドリブ演奏を行う前の曲には、4分音符以上の音長を持つ音がある。したがって、アドリブ演奏を行う前の曲を4分音符分の演奏で分割し、得られた4分音符分の演奏から、音長が最大の音を取り出す。この音長が最大である音は、持続音と呼ばれ、持続音と持続音をつなぐ短い音である経過音と比べ、与える印象が大きい。よって、取り出された音長が最大の音を単音データとして、音長を4分音符とし、演奏事例に格納する。最大の音長をとる音が複数取り出された場合、より前にある音の方が拍子に近く、与える印象が大きいので、より前にある音を単音データとする。図3に単音データの取得方法を示す。

【旋律データの取得】

アドリブ演奏には拍や小節をまたぐような音長を持つ音が使われることがある。したがって、4分音符分の演奏で音を分割し、分割された音は独立した音と見なし、演奏を取り出す。図4に旋律データの取得方法を示す。

3.3 フレームシステムを用いたデータ表現

演奏家から抽出された特徴を無秩序なデータ列として集める方法では、抽出された特徴の情報量や利用価値を下げてしまう。本手法による演奏家の特徴抽出手法は、演奏に関して演奏家による意見や説明の無い状況での、演奏からの抽出手法である。したがって、本稿では、演奏家の特徴を知識ベースとして獲得するため、演奏事例の表現方法として、フレームシステムを用いる。本稿では、フレームシステムの実装にXMLを用いる。XMLを用いた演奏事例の定義を図5に示す。図5において、CASE, NOTE-DATA, NOTE, TIME, PHRASE, ADLIB, 及び、DEGREE は、上で述べた、演奏事例、単音

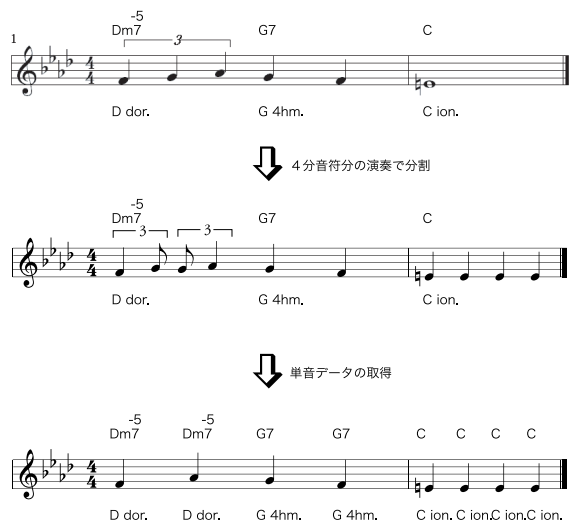


図 3: 単音データの取得方法

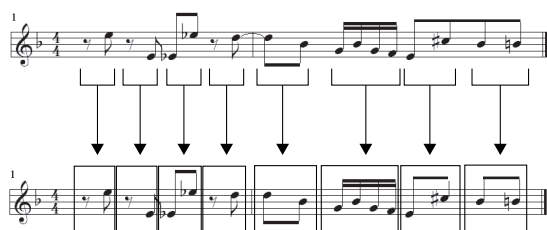


図 4: 旋律データの取得方法

データ, 音程, 音長, 旋律データ, アドリブデータ, 及び, 相対音程を示す. 要素“SCALE”は“ROOT”及び“TYPE”を要素に持つ.“SCALE”の要素である“ROOT”及び“TYPE”は, それぞれスケール・ルート及びスケール・タイプを示す.“SCALE”は, 属性として“ATT”を持ち, ドミナントスケールであるかそうでないかを示す.“SCALE”の要素が示す値は必ずどちらかの値をとるので, “SCALE”の属性は, “SCALE”の要素を抽象化したものと言える. また, 要素“CHORD”は“ROOT”及び“NAME”を要素に持つ.“CHORD”の要素である“ROOT”及び“NAME”は, それぞれコード・ルート及びコード・タイプを示す.“CHORD”は, 属性として“BASE”を持ち, トニック, サブドミナント, 及び, ドミナントのいずれかを示す. したがって, “CHORD”の属性は, “CHORD”の要素を抽象化したものと言える. XML を用いた演奏事例の例を図 6 に示す. 図 6 では, スケール“F Dorian”, コード“Fm7”, 及びアドリブ前の元となる音程 70 の状況で, 演奏家が, 休符 (x) を音長 320 の後, 音程 75 を音長 160 で演奏した事例を示している. この状況において, スケール及びコードは, ドミナントでない (NON-DOMINANT) という属性を持つスケール, サブドミナント (SUBDOMINANT) という属性を持つコードへと抽象化することができる.

4. 実験と評価

提案手法の有効性を確認するため, アドリブ演奏からの特徴抽出実験を行った. 特徴抽出の対象となる演奏家は, ジャズピアノのアドリブに関する教本を書いている藤井英一氏とした.

- <! ELEMENT CASE (NOTE-DATA, PHRASE) >
- <! ELEMENT NOTE-DATA (SCALE, CHORD, NOTE, TIME) >
- <! ELEMENT PHRASE (ADLIB+) >
- <! ELEMENT SCALE (ROOT, TYPE) >
- <! ATTLIST SCALE ATT NMTOKEN # IMPLIED >
- <! ELEMENT CHORD (ROOT, NAME) >
- <! ATTLIST SCALE BASE NMTOKEN # IMPLIED >
- <! ELEMENT NOTE (# PCDATA) >
- <! ELEMENT TIME (# PCDATA) >
- <! ELEMENT ROOT (# PCDATA) >
- <! ELEMENT TYPE (# PCDATA) >
- <! ELEMENT NAME (# PCDATA) >
- <! ELEMENT ADLIB (DEGREE, TIME) >
- <! ELEMENT DEGREE (# PCDATA) >

図 5: XML を用いた演奏事例の定義

データベースには, 実例ジャズ・ピアノ・アドリブ集 2 [4] より, 6 曲のアドリブ演奏を用いた. 本実験では, 4 分音符分の演奏からの特徴抽出について, ベクトル空間モデルを用いた余弦を利用する類似度評価を用いた.

【類似度計算アルゴリズム】

類似度を比較するアドリブ演奏を I とする. 本研究で提案した 4 分音符分の演奏に基づく特徴抽出で用いた手法により, アドリブ演奏 I から旋律データを抽出する. アドリブ演奏 I から抽出された旋律データを x_i と, y_i (i は自然数) で示す. 本類似度計算手法では, 旋律データの出現頻度をベクトル要素として用いる. したがって, 与えられた 2 つの旋律データが等しいかを判別する必要がある. ここで,

$size(x_i)$: x_i が保持するアドリブデータの数

とする. また, 旋律データが保持する k 番目のアドリブデータを a_k, b_k (k は自然数) とし,

$relative(a_k)$: アドリブデータの相対音程

$gate(a_k)$: アドリブデータの音長

とすると, $x_i = y_j$ (i, j は自然数) とは,

$$size(x_i) = size(y_j)$$

かつ, $1 \leq k \leq size(x_i)$ であるすべての k において,

$$relative(a_k) = relative(b_k)$$

$$gate(a_k) = gate(b_k)$$

である. アドリブ演奏 I において, 旋律データ x_1, x_2 の出現回数が表 2 のようになっていたとする. この場合, アドリブ演奏 I は, 表 3 のベクトルで示される. したがって,

$N(x_i)$: 旋律の出現回数

表 2: アドリブ演奏 I における旋律データの出現回数

旋律データ	x_1	x_2
出現回数	2	8

```

<CASE>
  <NOTE-DATA>
    <SCALE ATT="NON-DOMINANT">
      <ROOT> 6 </ROOT>
      <TYPE> Dorian </TYPE>
    </SCALE>
    <CHORD BASE="SUBDOMINANT">
      <ROOT> 6 </ROOT>
      <NAME> m7 </NAME>
    </CHORD>
    <NOTE> 70 </NOTE>
    <TIME> 480 </TIME>
  </NOTE-DATA>
  <PHRASE>
    <ADLIB>
      <DEGREE> x </DEGREE>
      <TIME> 320 </TIME>
    </ADLIB>
    <ADLIB>
      <DEGREE> 6 </DEGREE>
      <TIME> 160 </TIME>
    </ADLIB>
  </PHRASE>
</CASE>

```

図 6: 演奏事例を XML で記述した例

とすると与えられた2つのアドリブ演奏のベクトル空間 I_1, I_2 は, 表 4 のようになる. よって, I_1 と I_2 の余弦を利用した類似度 $\sigma(I_1, I_2)$ は,

$$\sigma(I_1, I_2) = \frac{I_1 \cdot I_2}{\|I_1\| \|I_2\|} = \frac{\sum_{i=1}^n N(x_i)N(y_i)}{\sqrt{\sum_{i=1}^n N(x_i)^2} \sqrt{\sum_{i=1}^n N(y_i)^2}}$$

により求められる.

【評価】

各アドリブ演奏の類似度を表 5 に示す. 抽出された4分音符分の演奏は, 1曲あたりおよそ300程度であった. 表 5 には, 曲 1 と曲 2 の類似度が0.80, 曲 1 と曲 3 の類似度が0.84 と記述されている. 本実験では, 類似度の最大値が0.88, 最小値が0.76, そして, 平均値が0.84となった. ここで行った類似度の比較は, アドリブ演奏の元となる曲が全て違うものである. したがって, 藤井氏のアドリブ演奏が, 一部の4分音符分の演奏を中心に構成されているといえる. 以上により, 藤井氏の特徴を抽出することができたといえる. また, 全く同じアドリブ演奏の類似度を評価すると, その値は1となる. しかし, 本手法での類似度評価では, 違う曲に対する違うアドリブ演奏同士の類似度が1となることがあり得る. 以上より, 本手法における類似度が1であることは, まったく同じアドリブ演奏であることを示すわけではなく, アドリブ演奏を構成する旋律が等しいことを示している. 類似度が1とならなかった要因を考えると, 類似度の平均と1との差が0.16であるこ

表 3: アドリブ演奏 I のベクトル空間

$$I = \begin{pmatrix} x_1 & x_2 \\ 2 & 8 \end{pmatrix}$$

表 4: アドリブ演奏 I_1, I_2 のベクトル空間

$$I_1 = \langle N(x_1) \ N(x_2) \ \dots \ N(x_n) \rangle$$

$$I_2 = \langle N(y_1) \ N(y_2) \ \dots \ N(y_n) \rangle$$

表 5: 藤井英一氏のアドリブ演奏の類似度

	曲 2	曲 3	曲 4	曲 5	曲 6
曲 1	0.80	0.84	0.78	0.86	0.76
曲 2		0.88	0.88	0.84	0.85
曲 3			0.80	0.86	0.88
曲 4				0.78	0.84
曲 5					0.88

とから, 多数の一致しなかった旋律が少数回使われている, もしくは少数の一致しなかった旋律が多数回使われていることが考えられる. 少数の一致しなかった旋律が多数回使われているということは, 演奏家の一時的な好みを表しているとも考えられる. また, 多数の一致しなかった旋律が少数使われているということは, アドリブ演奏において, 演奏家が新しい演奏の試みを行っているともいえる.

5. おわりに

本稿では, 4分音符分の演奏に基づく特徴抽出手法を提案し, データ構造を示した. 本手法の特徴は, (1) 4分音符分の演奏からの特徴抽出は, 曲全体における各単音の使用頻度や, 音から受ける印象等の状態遷移に基づく演奏家の特徴抽出と違い, 聴き手側と演奏者側からの両者の観点から感じる, 演奏の特徴をそのまま取り出しているため, より音楽的な性質を含んだまま, 演奏家の特徴を抽出することができる点, (2) 1曲のアドリブ演奏から多数のデータを抽出することができるため, アドリブ演奏のデータが少ない演奏家の特徴も抽出することができる点, (3) 演奏事例をフレームシステムにより表現することで, 演奏家の演奏事例から, 知識ベースを構築でき, 汎用的に用いることができる点, の3点である. 実験では, ベクトル空間に基づく類似度計算手法を用いて, 提案手法の評価を行うことで, 本システムの有効性が示された.

参考文献

- [1] 川村 修, 伊藤孝行, 新谷虎松: 演奏事例ベースに基づくアドリブ演奏模倣システムの実現, 情報処理学会全国大会, 2004.
- [2] 小谷 亮: エージェントとユーザの共同演奏システム, 情報処理学会論文誌, Vol.39, No.5, pp1547-1555(1998).
- [3] 西本一志, 間瀬健二, 中津良平: フレーズと音楽プリミティブの相互関係の可視化による旋律創作支援の試み, 情報処理学会論文誌, Vol.40, No.2, pp687-697(1999).
- [4] 藤井英一: 実例ジャズ・ピアノ・アドリブ集2<ジャズ・ピアノ・アドリブ解説と実例>, リズム・エコーズ (1999).
- [5] 篠田元一: 新・実践コード・ワーク① ボイシングとプログレッション, リットーミュージック (1997).
- [6] 篠田元一: 新・実践コード・ワーク② スケールとモード, リットーミュージック (1997).