

曲線描画に基づく即興演奏支援システム

北原 鉄朗^{1,a)} Sergio Giraldo^{2,b)} Rafael Ramírez^{2,c)}

概要：本稿では、ピアノロール上にユーザが曲線を描くことで即座に旋律が生成されて演奏される、新たな即興演奏支援システムを提案する。我々はこれまで旋律概形と呼ばれる曲線を用いて旋律を編集する手法について研究してきた。しかし、リアルタイムの旋律生成はできず、即興演奏支援への応用はできなかった。本研究では、この研究のコンセプトを踏襲し、ユーザが描いた旋律概形から即座に旋律が生成され、演奏表情付けされた上で演奏されるシステムを実現する。

1. はじめに

即興演奏は、音楽の楽しみ方のなかで最も高度で創造的な方法の1つと言えよう。即興演奏では演奏の進行中に瞬時に旋律を創作する必要があるため、楽器演奏経験や作曲経験がある人にとっても難しい演奏形態である。そのため、計算機技術を用いて十分なスキルのない人が即興演奏を楽しめるようにすることを目指した研究が、いくつか行われている [1], [2], [3], [4], [5]。

本稿では、旋律概形 [6] に基づいた即興演奏支援システムを提案する。本システムでは、ユーザがマウスなどで旋律の大まかな形を描画すると、システムは即座にメロディを生成する。線などの描画を用いた音楽インタラクションは、Hypersocre [7], InkSlorer [8], Music Sketcher [9] など、すでにいくつか提案されている。これらに対し、我々はこれまで、旋律を曲線として近似し、その曲線を描き直すことで旋律を編集する方法について研究してきた [6]。しかし、リアルタイムでは実行できず即興演奏の支援には使用できず、旋律生成のためのモデルパラメータは人手で決めなければならないという問題があった。本システムでは、この考え方を踏襲し、旋律の概形をユーザがマウスなどで描画すると、即座に旋律が生成する。その際、旋律の生成に必要なパラメータを既存の旋律データベースから求める。また、演奏表情付け手法 [10] を導入することで、より自然な演奏を実現する。

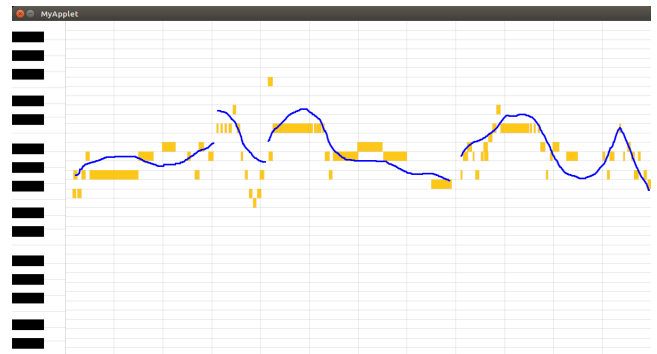


図 1 本システムの画面例

2. システム概要

2.1 処理の流れ

本システムの演奏中の画面の一例を図 1 に示す。起動するといわゆるピアノロール画面が表示される。本システムの画面を図 1 に示す。ここにユーザがマウスやタッチパネルなどを用いてフリーハンドで曲線（旋律概形）を描画すると、その概形に沿って旋律が生成されて表示される。旋律の生成処理は、マウスボタンを離すか、ドラッグを続けてマウスカーソルが次の小節の表示領域に移ったタイミングで開始される。また、旋律の生成は 1 小節単位で行われる。その後、演奏表情付けがなされ、演奏される。

この操作を演奏中に行うことで即興演奏を行うことができる。現状の実装では旋律生成に約 0.5 秒、その後の演奏表情付けに 0.2~0.3 秒程度（起動直後のみ 1.0 秒以上）かかるため、各小節の演奏が始まる 1.0 秒程度前までにその小節の旋律概形を描き終える必要がある。そのため、進行中の小節よりも 1~2 小節先の旋律概形を描くことになる。

¹ 日本大学文理学部情報科学科

² Music Technology Group, Universitat Pompeu Fabra

^{a)} kitahara@chs.nihon-u.ac.jp

^{b)} sergio.giraldo@upf.edu

^{c)} rafael.ramirez@upf.edu

2.2 伴奏の開始

本システムを立ち上げてキーを押すと、伴奏の再生が開始される。伴奏は標準 MIDI ファイル (SMF) としてあらかじめ用意するものとし、コード進行は既知とする。

2.3 旋律概形の描画

ピアノロール画面上でマウスやタッチパネルなどを用いて、フリーハンドで旋律概形 $\{y(t)\}$ を描くことができる。旋律概形 $\{y(t)\}$ は各時刻 t における音高を表し、MIDI ノートナンバーを連続値に拡張した値を用いる。現在の実装では、生成する旋律の最小単位は 3 連 8 分音符なので、 $y(t)$ を 3 連 8 分音符ごとに平均を取り、 t は離散時間とする。

2.4 旋律生成：リズム決定

マウスカーソルが m 小節目の領域にある状態でマウスボタンが離されるか、ドラッグによってマウスカーソルが $m+1$ 小節目の領域に移ったら、 m 小節目の旋律生成が始まる。

旋律はリズムを先に決定する。リズム決定に関する基本的なアイデアは、 $y(t)$ が大きく変化するところにオンセットを置き、近い値が続く場合はそのままの音を伸ばすということである。ただし、単なるしきい値処理ではまとまりのないリズムになってしまう。そこで、次のステップでリズムを決定する。

(1) リズムパターンの候補からなる集合 \mathcal{R} をあらかじめ決めておく。現在の実装では、旋律は 1 小節ごとに生成し、リズムの分解能を 3 連 8 分音符としているので、リズムパターンの各候補 R_k は、各要素が 1 (オンセット) か 0 (直前の音符の継続) の値を取る 12 次元ベクトルである。たとえば、 $R_1 = (1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)$ は「2 分音符が 2 つ続く」ことを意味する。

(2) 旋律概形 $y(t)$ から仮のリズム R' を決める。 R' は、 $y(t)$ があるしきい値 δ よりも大きく変化している時刻で 1、それ以外の時刻で 0 を取るように決める。旋律生成の候補となっている小節の先頭の時刻を t_0 とすると、 R' の i 番目の要素 $R'(i)$ は

$$R'(i) = \begin{cases} 1 & (y(t_0 + i) > \delta) \\ 0 & (\text{otherwise}) \end{cases}$$

で与えられる (ただし、 $i = 0, \dots, 11$ とする)。

(3) R' に最も近い候補を \mathcal{R} から選んで、それをリズム生成結果として確定させる。つまり、

$$\hat{R} = \operatorname{argmin}_{R_k \in \mathcal{R}} \|R_k - R'\|.$$

2.5 旋律生成：音高決定

リズム \hat{R} が決定したら、 \hat{R} の要素が 1 の時刻で発音すべき音の高さを決定する。 \hat{R} において値が 1 の要素が L 個あるとすると、 L 個の音高、つまり $N = (n_0, \dots, n_{L-1})$ を決

定する。各音高 n_i は MIDI ノートナンバーとする。

音高の決定には遺伝的アルゴリズム (GA) を用いる。GA では初期個体を複数個生成し、適応関数が最大の個体が得られるように個体の選択、交叉、突然変異を繰り返す。ただし、リアルタイム性を重視するため、この処理は 0.5 秒で打ち切る。以下、初期個体の生成と適応関数について述べる。

2.5.1 初期個体の生成

あらかじめ用意した旋律データベースから旋律のトライ木 (旋律木と呼ぶ) を生成し、旋律木をランダムにたどることで初期個体を生成する。旋律木の生成時には、通常のトライ木と同様にデータベース中の各旋律 (MIDI ノートナンバーの時系列) を格納していく。その際、出現頻度を記憶しておき、出現頻度の低い枝は後で削除する。旋律木の生成後は、木を根からランダムにたどることで初期個体を必要な個数分生成する。

2.5.2 適応関数

適応関数は、生成される旋律が次の条件を満たすときに高い値を返すように設計する：

- ユーザが描画した旋律概形に沿っている。
- 既存の音楽でも頻繁に出現する音高の並びである。
- 和音進行と不協和を生じない。
- 同じ音高が何度も続くなどの単純すぎる並びではない。

これらの条件を満たすように、音高列 $N = (n_0, \dots, n_{L-1})$ に対する適応関数 $F(N)$ を次のように定義する：

$$F(N) = w_0 \operatorname{sim}(N) + w_1 \operatorname{seq}_1(N) + w_2 \operatorname{seq}_2(N) + w_3 \operatorname{harm}(N) + w_4 \operatorname{ent}(N).$$

以下、それぞれについて述べる。

- $\operatorname{sim}(N)$: 旋律概形との類似度
 n_i の発音時刻を t_i とすると、次の式で定義する：

$$\operatorname{sim}(N) = - \sum_{i=0}^{L-1} (n_i - y(t_i))^2.$$

- $\operatorname{seq}_1(N)$: **bigram** 確率

$$\operatorname{seq}_1(N) = \sum_{i=1}^{L-1} \log P(n_i | n_{i-1}).$$

- $\operatorname{seq}_2(N)$: 隣り合う音高の差に対する **bigram** 確率

$$\operatorname{seq}_2(N) = \sum_{i=2}^{L-1} \log P(n_i - n_{i-1} | n_{i-1} - n_{i-2}).$$

- $\operatorname{harm}(N)$: 和音を条件とした条件付確率

ある和音において各音高 (特に非和声音) がどの程度許容されるかは、その音符の拍節位置に依存すると考えられる。そこで、拍節位置を小節の先頭 (head)、小節の先頭以外の表拍 (on-beat)、裏拍 (off-beat) に分類し、これも加えた条件付確率とする。すなわち、

$$\text{harm}(N) = \sum_{i=0}^{L-1} \log P(n_i | c_i, b_i).$$

ここで、 c_i は時刻 t_i における和音名、 b_i は時刻 t_i の拍節位置を表す ($b_i \in \{\text{head, on-beat, off-beat}\}$) .

• ent(N) : エントロピー

n_0, \dots, n_{L-1} を独立とみなしたときのエントロピーを $H(N)$, 既存の旋律のデータベースから求めたエントロピーの平均を H_{mean} とすると、

$$\text{ent}(N) = -(H(N) - H_{\text{mean}} - \varepsilon)^2.$$

ε は通常は 0 だが、正の数を与えることで通常よりも複雑さを増した旋律にできる .

上記のうち、 $P(n_i | n_{i-1})$, $P(n_i - n_{i-1} | n_{i-1} - n_{i-2})$, $P(n_i | c_i, b_i)$, H_{mean} は、旋律データベースから学習する . 重み値 w_0, \dots, w_4 および ε はあらかじめ定めた値を用いる .

2.6 旋律の表情付け

文献 [10] の手法を用いて旋律の表情付けを行う . この手法では、音符ごとに楽譜上の位置や音価、1 つ前の音符が楽譜上の位置や音価などの特徴量を抽出し、機械学習の手法を用いて発音時刻や音長の楽譜上のジャストな値からのずれ、音の強さを推定する . 推定手法には k 最近傍法を用いる . 逸脱の程度が大きすぎてかえって不自然な演奏になる場合は、制限を与えることができる . 制限の範囲は手動で設定する .

2.7 旋律の演奏

以上の処理によって生成された MIDI データが、演奏中の伴奏に合ったタイミングで演奏される . 演奏のタイミングまでに表情付けが完了しなかった場合は表情なしの旋律が演奏され、旋律生成が完了しなかった場合はその区間は旋律は演奏されない .

3. 実装・実行例

3.1 実装

以上で述べたシステムを実装した . 旋律のデータベースには、Weimar Jazz Database*1 から Key の項目に Blues と書いてある楽曲を抽出したものをを用いた . 楽曲数は 53 曲である . 演奏時のコード進行は、一般的なブルース進行である

| C7 F7 C7 C7 | F7 F7 C7 C7 | G7 F7 C7 G7 |

(上記において 1 小節あたり 1 つの和音を表しており、| は 4 小節ごとの区切りを表す) を用いた .

3.2 実行例

本システムを第 1 著者がタッチパネル液晶付きのノート

*1 <http://jazzomat.hfm-weimar.de/dbformat/dboverview.html>

PC にて試用した . その結果、演奏の進行中の小節よりも 1~2 小節先の概形を描くことで、即興演奏が可能であることが確かめられた . 旋律にはいわゆるブルー・ノートが多く含まれ、旋律データベースの特徴が反映されていた .

4. おわりに

本稿では、ユーザがピアノロール上でフリーハンドで線を描画することで旋律が生成される即興演奏支援システムを提案した . 今後は、評価実験を通じて本システムの有効性を定量的に検証するとともに、より大規模なデータベースを用いることで生成される旋律の改善を図っていきたい .

謝辞 本研究の一部は、日本学術振興会科学研究費助成事業 26240025, 26280089, 16H01744, 16K16180, 16KT0136 の補助を受けた .

参考文献

- [1] Parson, D. E.: Chess-based Composition and Improvisation for Non-musicians, *Proceedings of International Conference on New Interfaces for Musical Expression (NIME 2009)* (2009).
- [2] Amiot, E., Noll, T., Andretta, M. and Agon, C.: Fourier Oracles for Computer-aided Improvisation, *Proceedings of International Computer Music Conference (ICMC 2006)* (2006).
- [3] Buchholz, J., Lee, E., Klein, J. and Borchers, J.: coJIVE: A System to Support Collaborative Jazz Improvisation, *Technical Report* (2007).
- [4] 西本一志, 渡邊 洋, 馬田一郎, 間瀬健二, 中津良平: 創造的音楽表現を可能とする音楽演奏支援手法の検討: 音機能固定マッピング楽器の提案, *情報処理学会論文誌*, Vol. 39, No. 5, pp. 1556-1567 (1998).
- [5] 石田克久, 北原鉄朗, 武田正之: N-gram による旋律の音楽的適否判定に基づいた即興演奏支援システム, *情報処理学会論文誌*, Vol. 46, No. 7, pp. 1549-1559 (2005).
- [6] 土屋裕一, 北原鉄朗: 音符を単位としない旋律編集のための旋律概形抽出手法, *情報処理学会論文誌*, Vol. 54, No. 4, pp. 1302-1307 (2013).
- [7] Farbood, M. M., Pasztor, E. and Jennings, K.: Hyper-score: A Graphical Sketchpad for Novice Composers, *IEEE Computer Graphics and Applications*, Vol. 24, No. 1, pp. 50-54 (2004).
- [8] Garcia, J., Tsandilas, T., Agon, C. and Mackay, W.: Inksplorer: Exploring musical ideas on paper and computer, *Proceedings of International Conference on New Interfaces for Musical Expression (NIME 2011)* (2011).
- [9] Thiebaut, J.-B., Healey, P. G., Kinns, N. B. and Mary, Q.: Drawing electroacoustic music, *Proceedings of International Computer Music Conference (ICMC 2008)* (2008).
- [10] Giraldo, S. and Ramirez, R.: A Machine Learning Approach to Ornamentation Modeling and Synthesis in Jazz Guitar, *Journal of Mathematics and Music*, Vol. 10, No. 2, pp. 107-126 (2016).