

旋律包絡抽出に基づく直感的な旋律編集手法

土屋 裕一^{1,a)} 北原 鉄朗^{2,b)}

概要: 本研究では、音楽経験が少ないユーザに対して直感的に旋律を編集する手法を提案する。現在では様々な自動作曲システムが開発されているものの、生成された楽曲に納得いかない場合ユーザが自由に編集することができるシステムはない。一般的に MIDI データの編集は MIDI シーケンサなどに搭載されているピアノロールで行われるが、音楽経験が少ないユーザが利用する事が困難である。そこで本研究では、まず旋律をおおまかな形（旋律包絡と呼ぶ）に変換し、その旋律包絡上で編集を行う手法を提案する。旋律包絡とは、音高の時系列に対してフーリエ変換を行いフーリエ級数の低次の次数を抽出することで得ることができる。実験の結果から、提案した手法ではピアノロールのような音符レベルの編集に馴染みのないユーザがスムーズに旋律を編集できることがわかった。

キーワード: 旋律編集, フーリエ変換, 旋律表現, 旋律包絡, 隠れマルコフモデル

Intuitive Melody Editing Method Based on Melodic Envelope Extraction

YUICHI TSUCHIYA^{1,a)} TETSURO KITAHARA^{2,b)}

Abstract: In this paper, we propose a novel method for enabling musically untrained users to edit melodies intuitively. Although various automatic music composition systems have been developed, almost no systems enable users to edit automatically generated melodies as they intended. The piano-roll editor in the conventional MIDI sequencer can be used for this purpose, but it is difficult for musically untrained people to use. Our method first transforms a melody to a rough pitch contour, called a *melodic envelope*, and allows users to edit the melody at the melodic envelope level. The melodic envelope is obtained by applying a Fourier transform to the pitch trajectory of the melody and extracting low-order Fourier coefficients. Experimental results show that the proposed method enables users who are unfamiliar with piano-roll-like note-level melody editing to smoothly edit melodies.

Keywords: Melody edited, Transform, Melody expression, Melodic envelope, HMM

1. はじめに

近年のマルチメディア技術の発展によって、我々はこれまでになかった方法で音楽を楽しめるようになった。例えば、自動作曲システムを使えば音楽知識がない人で

も簡単にオリジナル曲を作曲することができる。そのためこれまでに様々な自動作曲に関する研究が行われてきた [1], [2], [3], [4], [5], [6]。これらのシステムは、ユーザが歌詞や鼻歌などの入力に基づき、自動的に旋律を生成する。しかし、実際には生成された旋律がユーザの意図するものになることは多くなく、ユーザが満足する楽曲が得られるとは限らない。生成された旋律がユーザの意図と異なる場合、次の2つの方法が考えられる。

- (1) 再度作曲システムを実行する。
- (2) 生成された部分を手動で編集する。

(1) は音楽の知識のない人でも簡単に試す事ができるが、

¹ 日本大学大学院総合基礎科学研究科
Graduate School of Integrated Basic Sciences, Nihon University

² 日本大学文理学部
College of Humanities and Sciences Nihon University

a) tsuchiya@kthrlab.jp

b) kitahara@chs.nihon-u.ac.jp

再度作成されたメロディーがユーザが意図するものになるという保証はない。一方、音楽知識があるユーザにとって、(2)は適切な解決法であるが、手動で旋律を部分編集することは、音楽知識のない人にとって難易度がとても高い。

本稿では、メロディーを編集できる新たな手法を提案する。音楽の非専門家は、音符単位ではうまく表現できないとしても「音が徐々に高くなって急速に低くなる」のような旋律の大まかな形についてイメージを持っている場合が多いと考えられる。本手法では、入力された旋律をおおまかな形（旋律包絡と呼ぶ）に変換しこれを編集する手法を提案する。以下、2章では旋律編集方法の基本コンセプトについて議論する。3章では、2章で議論した要件を満たすために達成すべき技術的課題およびその実現方法について議論する。4章では実際に本手法を用いた編集の実例を紹介および考察を述べる。最後に5章でまとめをする。

2. 旋律編集の基本コンセプト

旋律を編集する際に、もっとも一般的に使用されている方法はMIDIシーケンサなどに搭載されているピアノロールである。旋律中の音符がそれぞれ一本の棒として表現され、ユーザはこの棒を上下に移動させる事で旋律を編集することが可能である。この編集手法では、旋律編集経験の豊富なユーザは、旋律中の各音符を自由に編集する事ができる。しかし旋律編集経験が少ないユーザにはとても難しいと考えられる。例えば不適切な編集を行うと、伴奏と不協和な音を引き起こす可能性が少なくない。

そこで我々は音符表現を使用せずに旋律を編集する手法を提案する。そのために必要な技術は、旋律包絡と音符表現の相互変換系である。つまり音符表現で与えられた旋律を旋律包絡に変換し、ユーザが旋律包絡を編集した後、改めて音符表現に変換する。これを実現する上で、次の4つの要件を満たすべきである。

- (1) 旋律包絡では、各音符の音高や音価が陽には表現されない。
- (2) 音符表現を旋律包絡に変換し、編集せずに音符表現に再変換すると、元の音符表現に戻る。
- (3) 旋律包絡上で編集を行って音符表現に変換した場合、音楽的に不適切な音は避けられる。
- (4) 旋律包絡に旋律のどの程度細かな動きが表現されるかは、ユーザが制御できる。

従来研究では、これらの必要条件をすべて満たす旋律編集手法は存在していなかった。OrpheusBB[7]はユーザとインタラクションを取りながら、楽曲生成処理を繰り返して作曲していくシステムである。このシステムの特徴は、作曲された曲にユーザの意図を満たさない場合、ユーザはメロディやコード進行を自由に編集することができ、編集した部分が音楽的に不自然にならないように他の部分を自動的に再生成することである。しかし、編集は音符表現で

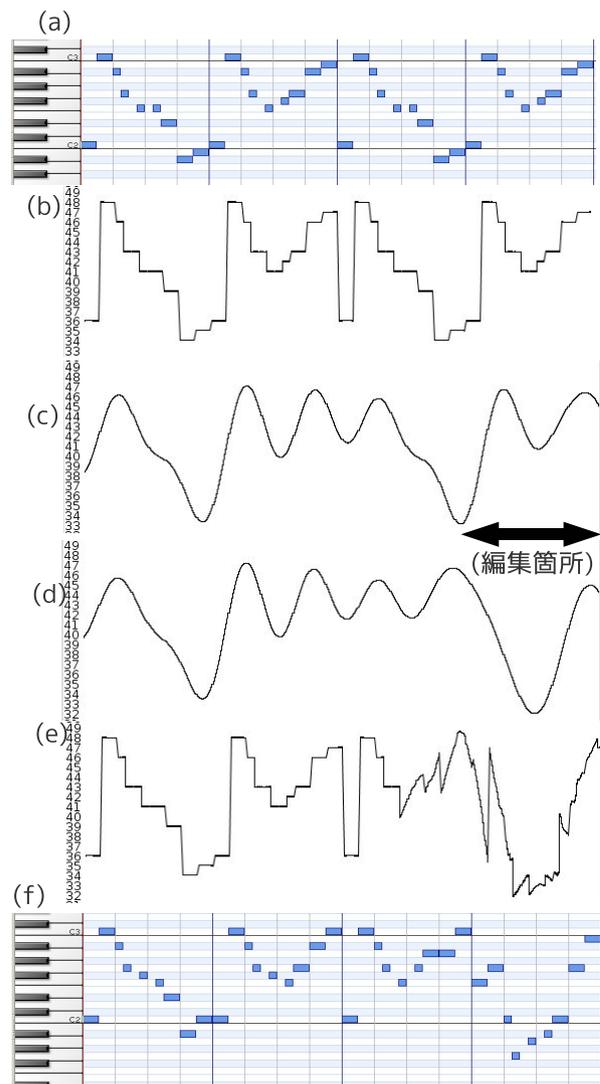


図1 旋律編集手法の概要。(a) 編集前の旋律、(b) 音高の時系列、(c) 編集前の音高の時系列。(d) 編集後の音高の時系列、(e) 編集後の音高の時系列、(f) 編集後の旋律。

あるので音楽編集経験の少ない人には難しいと考えられる。我々と同様な考え方に基づく作曲支援システムとしてLazyComposer[8]が提案されている。LazyComposerではユーザが一本の曲線を描くことにより、メロディーを作曲する事ができる。しかし、包絡上の細かさをユーザがコントロールできない点や作曲支援システムであり旋律を編集できない点がある。

本研究では、フーリエ変換を使用して旋律から旋律包絡への変換する手法を提案する。まず与えられたMIDIシーケンスを音高の時系列に変換する。音高の時系列を周期信号とみなしフーリエ変換を行う。この時低次の係数は旋律のおおまかな部分を表現しており、高次の係数は旋律の細部を表現している。したがって、低次の次数を逆フーリエ変換することによって旋律の大まかな部分を表現する旋律包絡を得る。

3. 提案する旋律編集手法

前章のコンセプトに基づいて考案した旋律編集手法を述べる(図1). 我々の旋律編集手法は, 以下の3つのステップで構成されている.

- (1) 音符表現から旋律包絡への変換
- (2) 旋律包絡の編集
- (3) 旋律包絡から音符表現への逆変換

旋律情報が MIDI 形式で入力されると, 音高の時系列が表示される. これを旋律包絡に変換しユーザがマウスを使って旋律包絡を編集した後, 音符表現に変換されて MIDI 形式で出力される.

3.1 音符表現から旋律包絡への変換

入力された MIDI データ(図1(a))は, 音高の時系列(図1(b))に変換される. ここで言う音高の時系列とは, 旋律の音の高さを時間の流れに沿って並べられた系列のことである. 音高が対数で表現され, その値は中央の C の音が 60.0, 半音が 1.0 (ノートナンバーの表現とは異なり, 非整数値でも構わない)となる. 音高の時系列を周期信号とみなしてフーリエ変換を行う. この時高次のフーリエ係数は旋律の細かい部分を表現するのに対して, 低次のフーリエ係数は旋律の大まかな動きを表すものである. これにより, フーリエ変換で得られた係数のうち低次のもののみを取り出して逆フーリエ変換を行う. 音高の時系列から旋律の大まかな部分を表現する成分のみ取り出すことができる(図1(c)). 抽出する低次のフーリエ係数の次数をユーザが決めるとして旋律包絡が表す旋律の細かさを制御することができる(図2).

3.2 旋律包絡の編集

旋律包絡はマウスのドラッグ操作で編集する事ができる. マウスのドラッグ操作で新たな曲線を描くと, 3.1と同様のフーリエ変換, 低次のフーリエ係数の抽出, 逆フーリエ変換が直ちに行われ, 設定された次数の範囲内で表現できる旋律包絡に変換される(図1(d)).

3.3 旋律包絡から音符表現への逆変換

編集後の新しい旋律包絡が得られ, 挿入するノートナンバーを選択し MIDI ファイルを生成する. 現在では簡単のため編集対象を音高のみの編集とし, 音符数や各音符の音価は元の MIDI データと等しいものとする. 3.2節の編集後に得られた低次のフーリエ係数と, 高次のフーリエ係数を結合して逆フーリエ変換をすることで, 微細な時間変化も含めた音高の時系列を得, MIDI 形式に変換する. まったく編集していない状態であれば, 音高の時系列は入力されたものと同じものを得ることができる. しかし, 編集をし

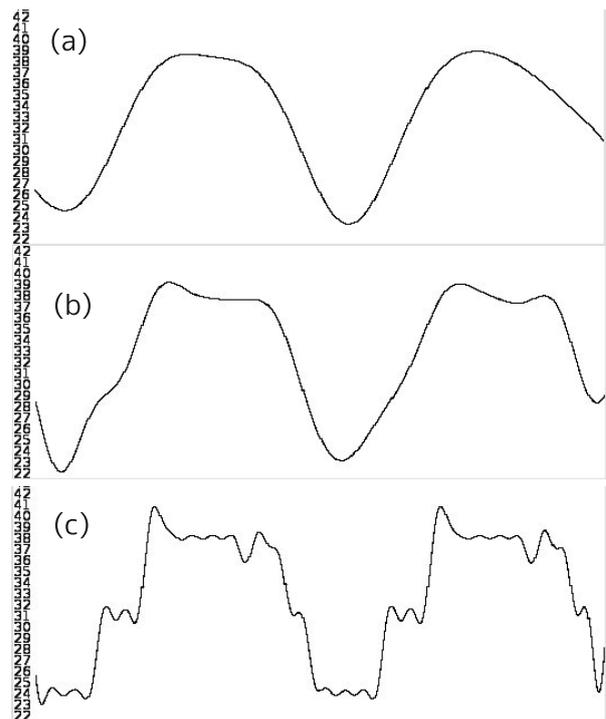


図2 低次の次数の範囲で表現される旋律包絡の粗さ. (a) 低次の次数が5次までで表現される図, (b) 低次の次数が10次までで表現される図, (c) 低次の次数が30次までで表現される図.

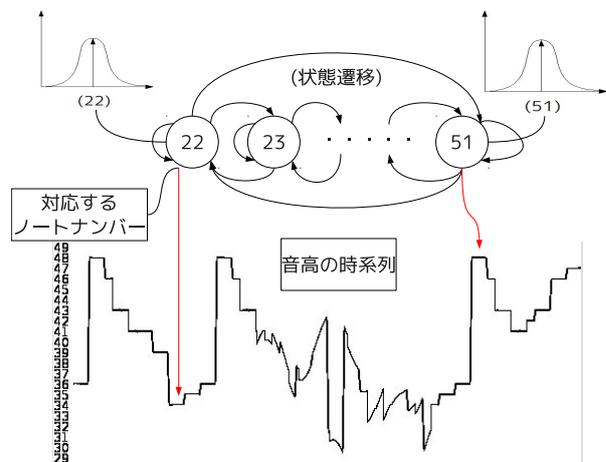


図3 編集をした音高の時系列から挿入するノートナンバーを推定する HMM の概要

た場合低次の係数にのみ作用するので, 得られる音高の時系列は歪んだものとなる. これを解決するために HMM を用いて, 歪んだ音高の時系列から適切なノートナンバーを推定する.

本研究で用いる HMM の概要を図3に示す. 歪んだ音高の時系列は, 音楽的に正しいノートナンバーの系列にノイズが加わって観測されたものであると考える. 各状態が各ノートナンバーに対応し, 正規分布に従ったノイズが付加されて音高の値が出力されると考える. 例えばノートナンバー 60 に対応する状態は 60.0 の平均と σ^2 の分散を持つ正規分布に従って音高の値を出力する. なお分散の σ^2 の

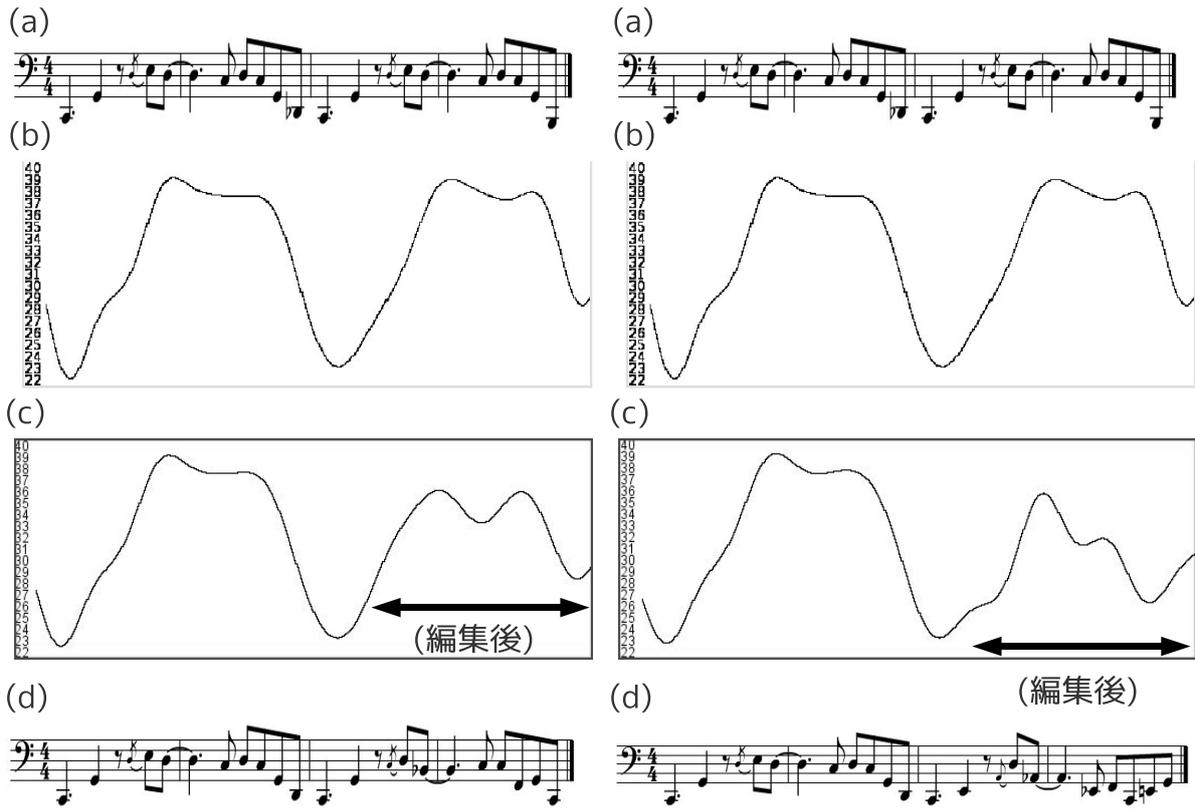


図 4 旋律編集の例. (a) 入力された旋律, (b) (a) の旋律包絡, (c) (b) を編集した旋律包絡,
 (d) (c) を音符表現に変換した旋律.

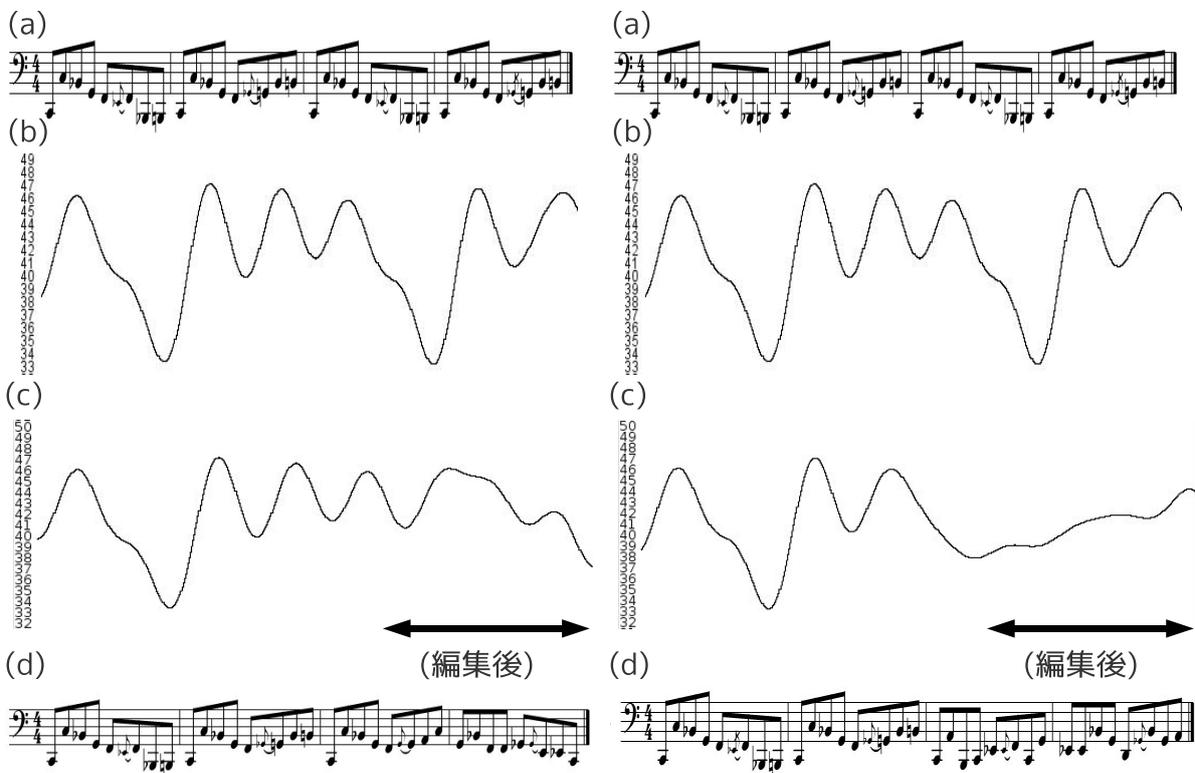


図 5 旋律編集の例. (a) 入力された旋律, (b) (a) の旋律包絡, (c) (b) を編集した旋律包絡,
 (d) (c) を音符表現に変換した旋律.

値はすべての状態の間で共通であり、実験的に決定される。

遷移確率は、編集後の旋律の音楽的特徴をモデル化する。メジャースケールの音に対応する状態遷移確率が高く設定されている場合は、推定される音符はメジャースケールになりやすい傾向がある。適切にこれらの状態遷移確率を設定することにより、音楽的に不適切な音高をとる可能性を少なくする事を可能にする。ここでは簡単のため、編集は音高のみとし音符数や音価は編集前の MIDI データと等しいものとする。そのため、決めるべき内容は各音符のノートナンバーのみである。そこで、状態は各音符の発音時刻のみで遷移してよいものとし、この HMM にビタビ (Viterbi) アルゴリズムを適用することにより、歪んだ音高の時系列 (図 1(e)) から音楽的に適切なノートナンバー列を推定する事が可能である。これらの手順を行い得た結果を MIDI 形式で出力する。

4. 手法の実装と試用

前章で提案した手法に基づき、旋律編集するシステムを実装し試用した。なお本研究では編集対象としてベースラインを取り上げる。これは、ベースラインはジャンルごとに特徴的な動きがあり、楽曲の雰囲気の特徴づける重要な要因になっていると考えているためである。

入力したのは 4 小節からなるベースラインの MIDI データである。それぞれノートナンバーの上限と下限を 51 と 21 に設定した。HMM の状態遷移確率を決定するために経歴 12 年のベシストに設定を依頼した。なお旋律包絡の低次のフーリエ係数は 10 に設定した。

4.1 試用結果 1

図 4 は、同じベースラインを異なる 2 つの方針で編集した結果を表すものである。図 4 の (a) が入力されたベースラインを表す。一般的な八長調のベースラインとなっており、2 小節めにレ♭がある以外は、すべてダイアトニックスケールの音である。(b) は、(a) を旋律包絡に変換したものである。この図より、急速に音高が上昇した後、ゆるやかに降下してから一気に降下するという流れが 2 回繰り返されるベースラインになっていることが読み取ることができ、入力されたベースラインの大まかな流れが適切に表現できていると言うことができる。

(c) は (d) をマウスのドラッグ操作で編集したもの、そして、(d) はそれを音符表現に再変換したものである。左の例では、後半の音が高くなっている箇所をもう少し低くしたいと思い編集した。出力された旋律では、元のものよりも編集箇所の音高が低くなっており、ユーザの意図を反映した旋律が出力されたと言える。しかし、3 小節めの最後から 4 小節めの初めにかけて比較的長い音がラ♭となっており、八長調としては違和感のある旋律となった。これは、HMM の状態遷移確率を設定する際に、ブルースケールのベース

ラインにも対応できるようにするために、ミ♭やシ♭の出現確率を高め設定したからである。今後は、スケールごとに状態遷移確率を設定し、ユーザが簡単に切り替えられる仕組みの導入が望まれる。

右の例では、後半に音が上がりきってから急峻に音が下がって再び音が上がるように編集した。編集後の旋律は、この編集意図が反映されたものになっているが、上と同様にラ♭やミ♭が出力され八長調としては違和感が残る旋律となった。また、2 小節めの終わりのレ♭が、左右どちらの例でもレに変化していた。これは、編集箇所極めて近い位置の音だったため、編集による歪みの影響を受けたものと考えられる。

4.2 試用結果 2

図 5 は図 4 と同様に、同じベースラインを異なる 2 つの方針で編集した結果を表すものである。図 5 の (a) に入力されたベースラインを表す。ブルースケールのベースラインであり、ミ♭やシ♭のブルーノートを含んだ旋律である。(b) は、(a) を旋律包絡に変換したものである。図 4 の旋律包絡に比べて図 5 の旋律包絡は形状が細くなっており、図 4 で用いた旋律よりも動きが細かい旋律であることが見て取れる。(c) は (b) をマウスで編集し、(d) はそれを音符表現に再変換したものである。左の例では、3 小節めの後半で少し音を高くし、そこから最後の 1 小節で徐々に音を低くするように編集を行った。出力された旋律では 3 小節めの後半で音が上がって、最後の 1 小節で音が徐々に低くなっており、ユーザの意図を反映した旋律が出力されたと言える。出力された旋律にはブルーノートであるミ♭やシ♭の音が複数ふくまれていた。これは HMM の状態遷移確率を設定する際にブルースケールに対応するように設定したからと言える。

右の例では、後半の 2 小節で低めの音から、徐々に音が高くなるように編集を行った。その結果、一度少し低くなってから徐々に旋律が上昇するイメージに近い旋律が出力された。ただし所々で上がり下がり繰り返した箇所があった。これは編集前の高次の係数を保存し編集後の低次の係数と結合したからである。本手法は、元の旋律の細かい動きを保存したまま旋律の大まかな流れをユーザが描画した曲線に合わせて変化させることを特徴としており、その特徴が現れた結果であると言える。

4.3 考察

試用結果から、旋律包絡から旋律のおおまかな流れや特徴がうまく表現されていたことがわかる。特に図 5(右)では、旋律包絡を徐々に音が高くなるように編集した。その結果徐々に旋律が上昇し、さらに所々で元の旋律の情報でもある音の上がり下がり繰り返した箇所があった。ここは高次のフーリエ係数を保存することで細かい部分を残し

大まかな編集をする狙いが現れていた。しかし、一方図4では編集箇所ではない部分でレリがレに変化した。これは、編集箇所に極めて近い位置の音で編集による歪みの影響を受けたものと考えられる。これを解決するために編集していない箇所の標準偏差を低く設定することで、歪みによる音の変化を改善する方法が考えられる。

さらに本研究の改善点として、状態遷移確率の改善があげられる。本手法では旋律包絡を用いて旋律を表現し、マウスのドラッグ操作で簡単に編集を行っている。その編集をした曲線から音楽的に適切な曲を作成する必要がある。本研究ではHMMでそれを実現しているが、出力された旋律に違和感が残る場合が少なくない。これを改善するために、HMMのように1つ前の音だけでなく、2つや3つ前の音から状態遷移確率をデータから得ることで出力結果に不協和音を生じる事が少なくなるだろう。そして編集後の曲に自由度を持たせるために、状態遷移確率を直感的に制御できるようなGUIを検討する必要がある。

5. 終わりに

本研究では、旋律を旋律包絡という一本の曲線に例え、包絡上で新たな線を描くことで旋律を編集する初心者向けの旋律編集手法を提案した。これを実現するために旋律分析手法としてフーリエ変換を用いた。旋律のおおまかな部分を表現する低次の次数を用いて旋律包絡を作成し、包絡上をマウスのドラッグ操作で新たな線を描くことにより、音符表現を使用せずに旋律編集することを可能にした。編集した旋律包絡から音楽的に適切な旋律を出力させるためにHMMを用いてノートナンバーを推定することで、音楽的に不適切な値を避けた。これにより、音楽知識がない人でも音楽的に適切に旋律編集することを実現した。今後は、自動作曲システムとの統合を進めていき、初心者でも1曲作成できるシステムにする予定である。

謝辞 有益なご意見をくださった松原正樹氏(筑波大学)に感謝する。

参考文献

- [1] L. Hiller, L. Isaacson, "Musical composition with a high-speed digital computer", *Journal of Audio Engineering Society*, 1958.
- [2] C. Ames, M. Domino, "Cybernetic composer: An overview," in *Understanding Music with AI*, M. Balaban, K. Ebcioglu, O. Laske, Eds. AAAI Press, 1992, pp.186-205.
- [3] D. Cope, "Computers and Musical Style", Oxford University Press, 1991.
- [4] 安藤 大地, P. dahlstedt, M. G. Nordahl, 伊庭 斉志, "対話型 GP を用いたクラシック音楽のための作曲支援システム", *芸術科学会論文誌*, Vol.4, No.2, pp.77-86(2005).
- [5] J. A. Biles, "Genjam: A genetic algorithm for generating jazz solos", in *Proc. ICMC*, 1994.
- [6] 深山 覚, 中妻 啓, 米林 雄一郎, 酒向 慎司, 西本 卓也, 小野

- 順貴, 嵯峨山 茂樹: "Orpheus: 歌詞の韻律に基づいた自動作曲システム", *情報処理学会研究報告*, 2008-MUS-76-30, pp.179-184, Aug. 2008.
- [7] 北原 鉄朗, 深山 覚, 片寄 晴弘, 嵯峨山 茂樹, 長田 典子: "OrpheusBB: Human-in-the-loop 型の自動作曲システム", *インタラクシオン* 2011, pp.57-64, Mar. 2011.
- [8] 池田 輝政, 菱田 隆彰: "劇伴の作成を支援する楽曲作成システム「Lazy Composer」", *情報処理学会研究報告*, 2011-MUS-91-10, pp.1-4, Jul. 2011.