

平成 24 年度 卒業論文

手書き入力によって盛り上がりをコントロール
するループシーケンサ

指導教員 北原鉄朗専任講師

日本大学文理学部情報システム解析学科

山下雄史, 岡田美咲

2013 年 2 月 提出

概 要

現代社会において、動画投稿サイトや DTM 等技術の発達により、作曲や編曲はより身近なものとなっている。そんな中で、作曲を始めてみようとする人も多く出てくるようになった。しかし、楽曲制作を本格的に始めるには、音楽の専門知識や楽曲制作をコンピュータ上で行うために専用ソフトウェアを使いこなす技術が求められる。

楽曲制作におけるテクノのようないわゆるクラブミュージックでは、パターンのループによってバックトラックが構成されていることが少なくない。このような楽曲をつくるには、ループシーケンサが有効である。これは、あらかじめ用意された音楽素材を組み合わせることで作曲が行えるものである。

しかし、作曲初心者が扱うとなると、問題となるのは複雑化した操作である。膨大な素材から自身が望むものを選び出すのは簡単ではない。その他にも、音楽理論の知識や、煩雑なパラメータ設定などにより、敷居が高いと考えられる。

ループシーケンサを用いた作曲で初心者が扱うことを想定すると、ユーザはおおまかに作りたい曲をイメージしていると考えられるが、その情報の一つに曲の盛り上がりが挙げられる。

そこで本研究では、初心者向けの作曲システムとして、曲の盛り上がりをグラフでユーザに入力させることで、音素材を自動的に探索し、挿入するループシーケンサを提案する。システムを実装し、検証した結果、グラフに沿った盛り上がりの曲が出力されることを確認した。

目 次

目 次	iii
図目次	v
第1章 序 論	1
1.1 本研究の背景	1
1.2 目 的	2
1.3 本論文の構成	3
第2章 作曲システムのアプローチ	5
2.1 関連作曲システム	5
2.1.1 関連作曲ソフト	5
2.1.2 関連研究	6
2.2 まとめ	8
第3章 システム構成	11
3.1 システム概要	11
3.2 音素材の決定方法	13
3.2.1 隠れマルコフモデルを用いた定式化	13
3.2.2 モデルの簡略化	14
3.2.3 実際の音素材決定アルゴリズム	14

第4章 実装と試用結果	19
4.1 実装	19
4.2 試用結果	20
4.2.1 曲線入力による試用結果	20
4.2.2 編集機能による試用結果	25
第5章 結論	33
参考文献	35

目 次

3.1	作曲画面	12
3.2	状態遷移表. 左から, 状態の通し番号, 現在の配置, 次の時刻の盛り上がり度が1の場合の遷移先を示す. 同様にして2~5の各遷移先を示す. 1は配置する, 0は配置しないことを指す. 5桁の番号は左から, Drum, Percussion, Bass, Synth, Sequenceを表している.	16
3.3	各素材の盛り上がり度を記載したリスト. 左から, 素材の通し番号, ファイルへのパス, 盛り上がりの値1~5を示している.	17
4.1	システムの流れ	19
4.2	実行例1の出力結果画面	21
4.3	実行例2の出力結果画面	23
4.4	実行例3の出力結果画面	24
4.5	実行例4の出力結果画面	27
4.6	実行例5の出力結果画面2	30
4.7	実行例7の出力結果画面	32

第1章 序 論

本章では、研究の背景、目的、従来研究との違いなどを、過去の論文を引用しながら述べる。最後に、本論文の構成を述べる。

1.1 本研究の背景

現代社会において、動画投稿サイトや DTM 等技術の発達により、作曲や編曲はより身近なものとなっている。そんな中で、作曲を始めてみようとする人が多く出てくるようになった。しかし、そういった活動を本格的に始めるには、音楽の専門知識が必要であり、また楽曲制作をコンピュータ上で行うために専用のソフトウェアを使いこなす技術も求められる。

そんな初心者が作曲活動を始めるにあたって、敷居が低く扱いやすいと言われているのが、ループシーケンサである。ループシーケンサソフトはあらかじめ用意された音楽素材を組み合わせることによって作曲が行えるものである。例として、Acid Pro[1] など市販されているものを含めて様々なものがある。多くの機能を使いこなし、細部に渡って編集することにより、普段耳にする楽曲と何ら変わらないクオリティで、0 から作り出すよりもはるかに容易な楽曲制作が可能となる。

しかし、初心者がこのようなソフトを扱うとなると、問題となるのは複雑化した操作である。膨大な素材集から自身が望むものを選び出すのは簡単ではない。その他にも、音楽理論の知識や、難解で煩雑なパラメータ設定、その他多くの機能の混在により、作曲初心者が扱うには敷居が高いと考えられる。

これまでに作曲初心者を対象としたソフトウェアのアプローチとして、「遊び」

を題材として視覚的に作曲を行えるシステムに関する研究 [2] など、様々な研究が行われてきた。

そこで我々は、新しいアプローチで初心者扱いやすいループシーケンサのシステムを提案する。

1.2 目 的

ループシーケンサを用いた作曲で初心者が扱うことを想定すると、ユーザは明確で詳細なメロディーよりも、曲の流れにおける盛り上がりといった、曲の雰囲気や想定していると考えられる。

曲の盛り上がりは、特別知識を持たずとも聴いているだけで比較的分かりやすく、楽曲製作時も素材の組み合わせ、重ね合わせにより再現が容易である。

そこで我々は曲の盛り上がりに着目し、曲の盛り上がりユーザに入力させることで、音素材を自動的に探索し、挿入するループシーケンサを提案する。

しかしここでは、2つの問題点がある。

1つ目は、曲の盛り上がりという曖昧なものをどのように表現し、入力させるかである。これは、曲の流れを直感的に捉えられると思われる曲線グラフをマウスのドラッグ操作で入力することで解決する。そして入力された盛り上がり情報から、音素材の配置のオン・オフを決定することによって盛り上がりコントロールする。

2つ目は、盛り上がり考慮した音素材の推薦をどのように行うかである。これは、各音素材に盛り上がり値を付与し、自動出力の際に選ばれる音素材を制限することで解決する。

1.3 本論文の構成

本論文は次の構成からなる。第2章では従来の作曲支援ループシーケンサソフトや研究を紹介し、従来システムの課題及び本研究の方針について述べる。第3章では、盛り上がりの曲線グラフから音素材を自動出力する際に用いるデータベース、ルールベースについてと、システムの概要について説明し、ルール設定の方法と設定値について述べる。第4章では、試用実験と考察を行う。第5章では本研究の結論と今後の課題について考察する。

第2章 作曲システムのアプローチ

本章ではまず従来の作曲支援システムを紹介する。これらのシステムの課題について述べた後、それらを踏まえて本研究のアプローチと特徴を述べる。

2.1 関連作曲システム

ここでは自動作曲システム、特にループシーケンサーが主となるものについて関連研究を述べ、そのシステムの課題を示す。

2.1.1 関連作曲ソフト

- ACID Pro[1]

ACID Pro は、ドラッグドロップを中心とした操作により楽曲制作が出来る。使用できる素材は 3000 以上と豊富にあり、種類は、ドラム、管楽器、弦楽器やユニークな効果音など幅広くある。これらの素材を様々に組み合わせることによってオリジナルの音楽を作成できる。また、ピッチやテンポの自動的マッチ(ピッチやテンポの合わないループ素材でも、自動的にマッチさせる機能)ができる。ループシーケンサの機能だけでなく、MIDI 打ち込み、マイクやピアノを繋いで歌や楽器の演奏を録音、編集することができ、またそれらとループシーケンサの機能を組み合わせた楽曲制作も可能である。

- MusicShake[3]

MusicShake は SNS とループシーケンサを組み合わせたシステムである。マス目状に並んだボタンをクリックしていくだけで楽曲制作が出来る。ユーザー同士、インターネット上で楽曲作品や新規に制作した素材を公開し、コミュニケーションをとることが出来る。最大の特徴は、そのソーシャルシステムを用いて共有され、日々増えていく素材の膨大さにあり、ソーシャルネットワークを利用したデータベースにより、現在使用できるものは 550,000 以上までのぼる。

- Band in a Box[4]

Band in a Box は、コードの入力、伴奏スタイルの選択から作曲を行うシステムである。伴奏スタイルとして、ジャズスイング、ブルース、シャッフル、ミディアムロック、ワルツ、ポップバラードなどが用意されている。目的のジャンルのスタイルを選択するだけで、入力したコード進行通りの演奏を作成することができる。また、コードの知識がなくても、システムが自動でコード進行を作成する。

2.1.2 関連研究

- 音楽で「遊ぶ」ことを目的とした作曲システムの構築に関する検討 [2]

初心者向き作曲ソフトウェアの特徴を捉えながら、遊びの感性を取り入れていく研究である。遊びの感性は初心者でも視覚的に分かりやすいように「パターンコーディング」を用いることによって実現されている。パターンコーディングではピッチの高さやメロディの音調を表現することができる。これによって複雑なパラメータ設定を必要としなくてもある程度の曲を作曲することが可能になる。ただし、パターンコーディングの特性上、1音しか再現できないことが難点である。

- 発想支援法を用いた作曲支援システムについて：発想支援法 [5]

創造学という学問の中の、発想支援法という考え方がある。その中からいくつかの発明技法をループシーケンサーの作曲システムと結びつけることにより、音楽の専門知識が無い人でも、いかに感性を発揮出来るようなシステムを作る事が目的である。システム内では、ユーザー独自のデータベースを構築することから始まり、使用する素材を何小節目で鳴らすかなどのカテゴリ分けを行う。その後、曲を生成し、どのタイミングでループさせるか、組み合わせを変えてみるか、また新しく素材を考える必要があるのかをシステムが提示する。この繰り返しにより、ユーザーが納得出来る曲を作っていく。

- Seek Rope：曲げて切って結べるシークバー [6]

音楽プレーヤに見られるシークバーを紐のように曲げたり結合したりすることで、自由に曲のプレイリストをシークバー（シークロープ）の観点から作成することができるという研究である。

- Au-thello：ゲームを「オセロ」をモデルにした音楽演奏コントローラ [7]

オセロゲームを題材としたシーケンサソフトである。各駒に音素材を割り当て、白の駒には丸い音の印象の音を、黒の駒には角張った印象の音をそれぞれ当てはめる。著者は一般的に知られているゲームのロジックを用いることでわかりやすいシーケンサソフトのシステム設計を目指したが、同色に挟まれた色は反転するというロジックにより作曲の自由が効かなくなることでシステムが複雑化した。

- 音楽表情を担う要素と音高の分割入力による容易な MIDI シーケンスデータ作成システム [8]

楽器の演奏経験が乏しい人でも MIDI シーケンスデータを作成することと、従来の楽器と比べて遜色のない音楽表情づけのポテンシャルを持ち合わせる

「2段階式作成方法」を提案している。第1段階ではノートナンバーのみを入力する。第2段階では、個々の音符に含まれる音楽表情に関する情報を、リアルタイム入力のようにMIDI楽器を演奏しながら統合的に入力する。ここでの音楽表情は、「細かい速度の変化」「細かい強弱の変化」「アーティキュレーション」の3つの要素と、MIDIで出力される「ヴェロシティ」「発音時刻」「消音時刻」の3つの情報との関係を考察することで表現している。

- D-touch : A consumer-Grade Tangible Interface Module and Musical Applications[9]

作曲ソフトの音源の入力方法として、有形のオブジェクトをWebカメラで読み込ませることで実現する手法を提案している。最大の特徴はコンピュータのモニターを作曲に使用しない点である。モニターを介すことなく、有形オブジェクトをWebカメラで認識して音源を入力することで音がスピーカーから出力される。作曲は用意された有形オブジェクトを並べることで実現する。

2.2 まとめ

2.1.1節で、従来のループシーケンサをいくつか取り上げたが、初心者がこれを扱おうとする上で、以下の2つの点が問題になると考える。

1. 基本的な操作は簡単ではあるが、その他付随して、音響効果等のパラメータを調整する機能が複雑であり、その操作には専門知識が要求される。
2. 使用出来る素材が膨大過ぎて、どの素材を使い、どのように組み合わせれば良いかが分からず、意図する楽曲を制作するのが困難になる。

本研究では、1に対しては複雑なパラメータ調整は実装せず、簡単な操作のみで楽曲制作を行えることに重点を置くこととする。簡単な操作を実現するにあたり、

ユーザの意図する楽曲制作の1要素として、曲の盛り上がりの様子に着目する。曲線として可視化、入力させることで、マウスのクリックやドラッグなど、最低限の操作で楽曲制作が行えるようにする。

2に対しては、ユーザが任意に素材を選択する補助として、使用する素材や組み合わせ、曲中の配置に関して推薦するシステムを提案する。

第3章 システム構成

本章では、システムの概要を述べ、隠れマルコフモデルについて説明し、出力が行われる際にどのような手法を取っているかについて述べる。

3.1 システム概要

作曲画面は、一般的に親しみがあると思われる「パズル」をモチーフにした作曲画面を採用する(図 3.1)。以下に作曲におけるシステムの流れを示す。

- 曲のメインとなる主旋律の音源を画面右側のリストから選ぶ。
- 画面左上の枠内に意図する曲の盛り上がりの曲線を、マウスのドラッグ操作によって描く。すると、枠内下の格子状の枠内に音素材が自動的に挿入される。
- 音素材は変更可能で、各々の素材もしくは空のマスをクリックすると、音素材を選択する画面が表示される。そこから任意の音素材を選択すると、素材の変更及び挿入することができる(画面下のツールで素材の削除も可能)。
- 音素材を変更すると、その次の小節にふさわしい音素材を探索し、自動的に変更する。この処理が再帰的に繰り返される。ただし、この自動変更機能は無効にすることもできる。
- また、1小節ごとにループする回数、曲全体の長さの変更(4小節ごと)は、画面左下で変更できる。

The screenshot displays a music composition interface. At the top, a blue waveform represents the audio signal. Below it is a grid of notes, with some cells containing numerical values. The right side of the interface shows a list of audio files, including 'Backing 1.wav', 'Belzebub 1.wav', 'Bleep 1.wav', 'DistTB 1.wav', 'DistTB B 1.wav', 'Faster 1.wav', 'GoSeq 1.wav', 'Hackerseq 1.wav', 'Hard Edge 1.wav', 'Romancer 1.wav', 'SequenceA 1.wav', 'Smashed 1.wav', 'TB 3003 1.wav', 'TB 3003 B 1.wav', 'TB 3003 C 1.wav', 'TB Add 1.wav', 'TB Answer 1.wav', 'TB Wild 1.wav', 'TB Wow 1.wav', and 'Wow 1.wav'. At the bottom, there are navigation arrows, a '1' button, a '16' button, and two toggle switches: '消しゴム' (Erase) set to 'OFF' and '自動推薦モード' (Auto Recommendation Mode) set to 'ON'. An 'EXIT' button with the text 'クリックで終了' (Click to end) is located at the bottom right.

174	177	175	178	176	178	179	180	174	177	175	178	176	178	179	180
									352	350	353	316			
			11					49	52	50	53	37	39	40	41
157				158				158				156			
	109	109	109	109	109	109	109	109	109	109	109	109	109	109	109

図 3.1: 作曲画面

3.2 音素材の決定方法

ユーザが盛り上がりグラフを入力した後、音素材の配置方法は状態遷移によって決める。これは、音素材の配置を時系列ごとに見たとき、一意に状態遷移先が決まらないため、隠れマルコフモデルで表現することができる。

3.2.1 隠れマルコフモデルを用いた定式化

マウスカーソルのドラッグによって描画された盛り上がり度の時系列から、それに合う音素材を推定する処理を、HMM を用いて定式化する。

ここでは、用いる音素材はすべて同じテンポで、1小節分の長さになっているものとする。楽曲の小節数 (N とする) は既知、パート数 (M とする) は固定とする。

いま、 n 小節目の盛り上がり度を x_n 、パート m における n 小節目の音素材 (当該パートに音素材を挿入しない、つまり無音の音素材も含む) を $s_{n,m}$ とし、 n 小節目の音素材の組み合わせを $s_n = (s_{n,1}, \dots, s_{n,M})$ とすると、ここで取り組む問題は、 $[x_1, \dots, x_N]$ から $[s_1, \dots, s_N]$ を推定することである。盛り上がり度 x_n は音素材の組み合わせ s_n に依存して決まり、 s_n は1つ前の組み合わせ s_{n-1} に依存して決まると考えることができる。そこで、 m パート分の音素材の組み合わせを「状態」とみなすことで、HMM における典型的な、出力系列を推定する問題に帰着することができる。

3.2.2 モデルの簡略化

上の定式化では、音素材のすべての組み合わせに対して出力確率と状態遷移確率を設定または学習する必要があり、パート数や素材数が多い場合には現実的ではない。そこで、次のように簡略化を行う。

- パート数を5に固定し、盛り上がり度1から5の整数に離散化する。
- 挿入されている音素材の個数を、その時点での盛り上がり度とみなす。つまり、 $s_n = (1, 11, 0, 20, 0)$ (各自然数は素材番号を表し、「0」は当該パートに素材を挿入しないことを表す) のとき、この状態が出力する盛り上がり度は常に「3」とする。
- ただし、音素材は盛り上がり度1のものから5のものまで手動で分類しており、盛り上がり度3の小節では、盛り上がり度3に分類されている素材のみを用いることができるものとする。
- 状態遷移については、各パートに音素材を挿入するかどうかと、挿入するならどの素材を挿入するかの2つに分けて考える。前者については、手動で作成した状態遷移表に基づいて決定する。後者については、4小節ごとに区切った各々における最初の小節ではランダムに決め、それ以外では原則として直前の小節のものと同じものにする。
- ただし、パーカッションについては、スネアドラムの8分連打のようにフィルイン的な役割を担うものは、4小節ごと区切った各々における最後の小節のみ挿入できるなど、素材の性質ごとに挿入箇所を制限する。

3.2.3 実際の音素材決定アルゴリズム

上の簡略化を行うことで、次のアルゴリズムで音素材を決定できる。

1. 入力された盛り上がり度から小節ごとに挿入する音素材の個数を決定する.
2. 状態遷移表 (図 3.2) から, 小節ごとに音素材を挿入すべきパートを決定する.
3. 各音素材に付与されている盛り上がり度 (図 3.3) から, 挿入できる素材を絞り込む.
4. 絞り込まれた素材から, 上で述べた制約をみたすように, 実際に挿入する素材を決定する.
5. この際, 音素材ごとの制約をみたすように, パーカッションパートに挿入する素材を決定する.

```

0, 00001, 00001, 00011, 00111, 01111, 11111
1, 00010, 00001, 00101, 00111, 01111, 11111
2, 00011, 00010, 00011, 01011, 01111, 11111
⋮      ⋮      ⋮      ⋮      ⋮      ⋮      ⋮
31, 11111, 00001, 00011, 01011, 01111, 11111

```

図 3.2: 状態遷移表. 左から, 状態の通し番号, 現在の配置, 次の時刻の盛り上がり度が 1 の場合の遷移先を示す. 同様に 2~5 の各遷移先を示す. 1 は配置する, 0 は配置しないことを指す. 5 桁の番号は左から, Drum, Percussion, Bass, Synth, Sequence を表している.

0,TechnoTrance/Bass/DanceOff 1.wav,4
1,TechnoTrance/Bass/DanceOff 2.wav,4
2,TechnoTrance/Bass/DanceOff 3.wav,4
3,TechnoTrance/Bass/DanceOff 4.wav,4
4,TechnoTrance/Bass/DanceOff 5.wav,4
5,TechnoTrance/Bass/DanceOff 6.wav,4
6,TechnoTrance/Bass/DanceOff 7.wav,4
7,TechnoTrance/Bass/EightBass 1.wav,1
:
362,TechnoTrance/Synth/Stack A 7.wav,2
363,TechnoTrance/Synth/TB Dist 1.wav,1
364,TechnoTrance/Synth/TB Dist 2.wav,1
365,TechnoTrance/Synth/TB Dist 3.wav,1
366,TechnoTrance/Synth/TB Dist 4.wav,1
367,TechnoTrance/Synth/TB Dist 5.wav,1
368,TechnoTrance/Synth/TB Dist 6.wav,1
369,TechnoTrance/Synth/TB Dist 7.wav,1

図 3.3: 各素材の盛り上がり度を記載したリスト. 左から, 素材の通し番号, ファイルへのパス, 盛り上がり度の値 1~5 を示している.

第4章 実装と試用結果

以下の節では、本稿で提案するシステムがどのように実装されているか、また、これを用いた作曲の使用例と試用結果について述べる。

4.1 実装

本稿で提案するシステムは Processing[10] を用いて実装した。WAV の読込、合成ファイルの生成には、CrestMuse Toolkit[11] を用いた。音素材は「Sound Pool」[12] という素材集から抜粋した。3.2.3 節で述べた状態遷移表および各音素材の盛り上がり度は、第1著者が手動で作成・付与した。図 4.1 にシステムの流れを示す。

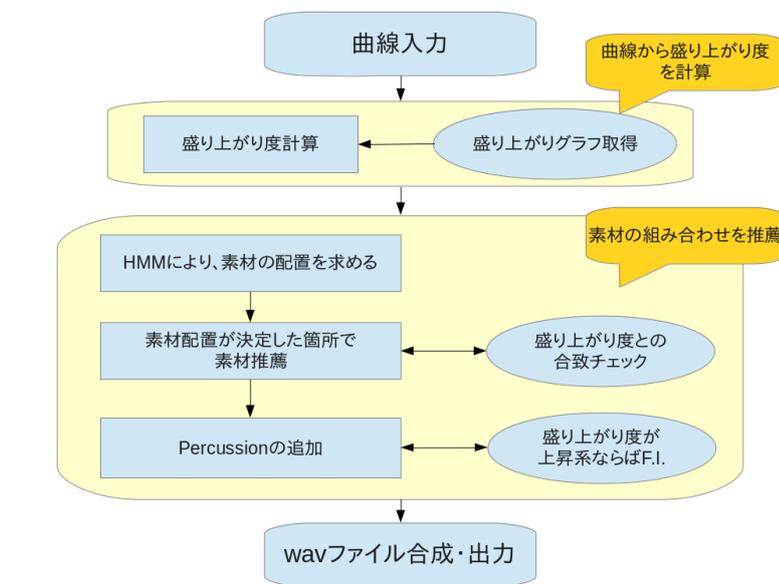


図 4.1: システムの流れ

4.2 試用結果

ユーザは実行の際にどのような曲を生成したいかを表した、盛り上がりの推移条件と実行結果と比較する。長さは16小節(素材を16個繋げたもの)、1小節分の繰り返しは1回とする。

4.2.1 曲線入力による試用結果

実行例1

最初に、4小節ごとに役割がはっきりと分かる曲線を入力する。その際に設定した条件は以下の3点である。

- 最初の4小節で盛り上がり度を最低値から最高値に上げる。
- 次の4小節中で最高値から最低値まで下げる。
- 残り8小節でも同様に繰り返す。

その出力結果を図4.2に示す。入力した曲線グラフの値が大きいほど1小節ごとに配置される音素材が多くなり、挿入される素材も盛り上がりの大きいものになる。Synth, Bassでは曲中2ヶ所に素材がかたまって配置された。

試聴した結果、盛り上がりが大きくなる部分でSynth, Bassがアクセントとなり、盛り上がりが高低と変化する様子が分かりやすい楽曲となった。また、Sequenceを主旋律に据え常に再生されるようにしたことで、一貫性のある曲が出力されていた。

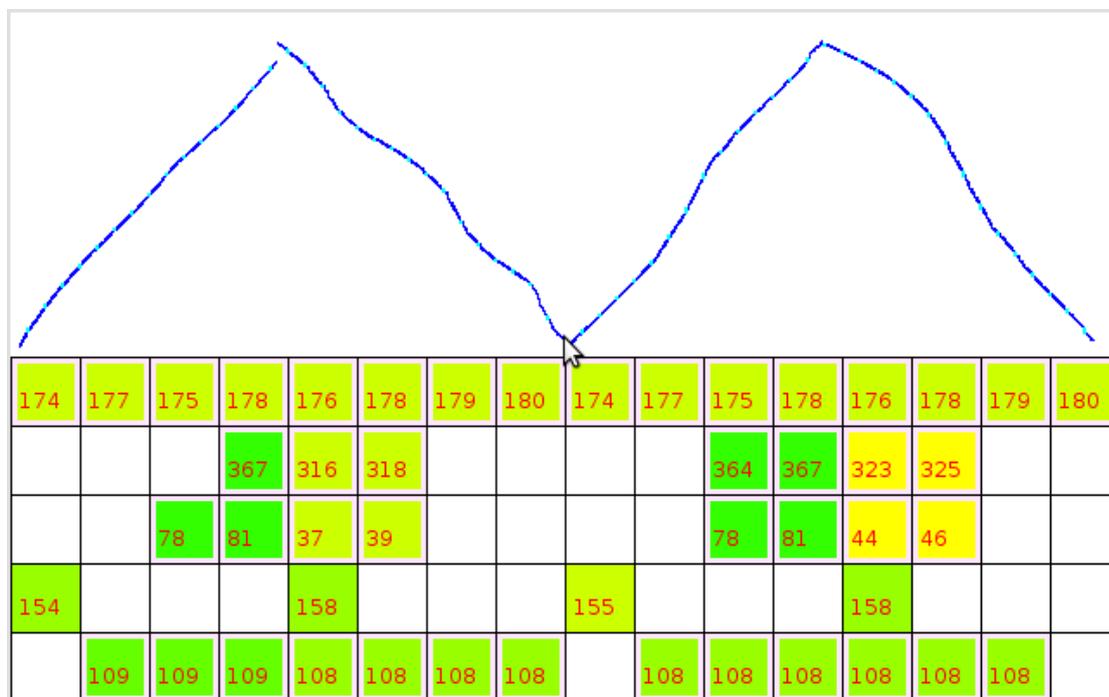


図 4.2: 実行例 1 の出力結果画面

実行例2

ある4小節でのみ盛り上がりが細かく変化した曲線を入力する。設定した条件は以下の3点である。

- 最初の4小節で盛り上がり度を最高値から中程度に下げる。
- 次の4小節中で盛り上がり度を少し上げてから最低値まで下げる。
- その後8小節かけて盛り上がり度を最大まで上げる。

その出力結果を図4.3に示す。実行例1の結果に比べ、連続で挿入される素材が多く、変化の少ない曲となった。入力した曲線のとおり、盛り上がり度が3/4/3などになるような一部が突出した形のグラフが入力された場合、比較的一个の楽器が連続して流れることが多い曲中において、唐突に1小節だけ再生される素材が挿入された。そのため、その部分のみが曲の流れにそぐわず不自然になることがあった。

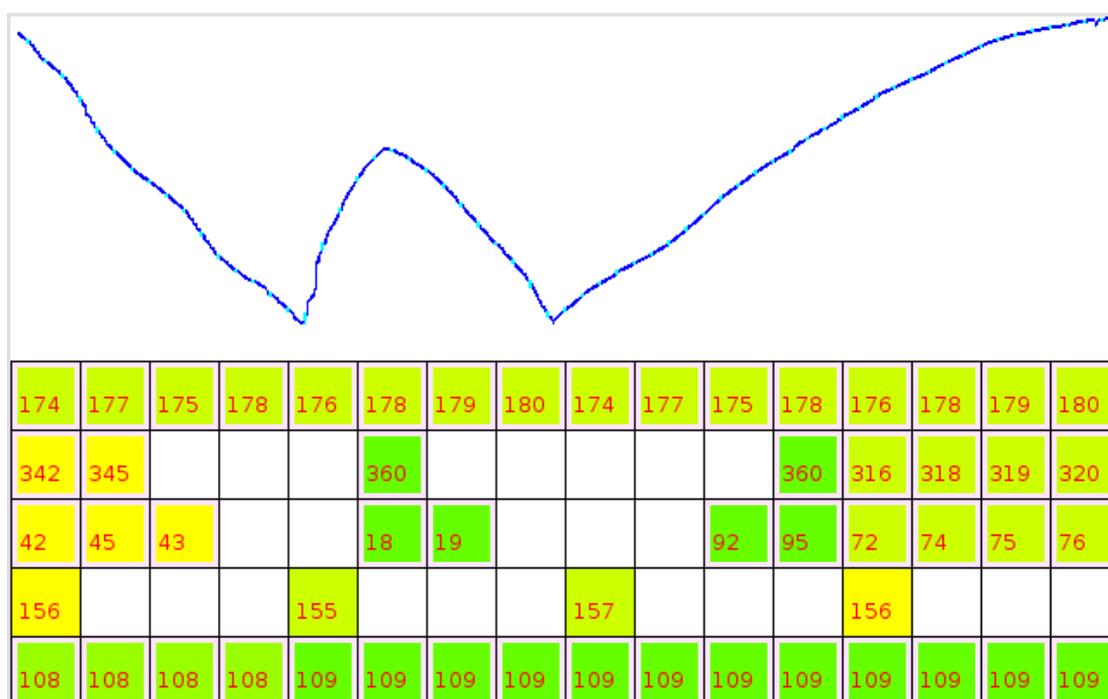


図 4.3: 実行例 2 の出力結果画面

実行例3

実行例2で不自然だった部分を更に検証する. 4小節中で盛り上がりを低い状態から中程度まで上げ, また低い状態にすることを4回繰り返し替えた.

その出力結果を図4.4に示す. 1小節だけ素材が挿入されてもそれが曲中で何度も繰り返されれば, 単発の素材が次々と再生される特徴的な曲となった. 実行例2のような変化の少ない曲中では不自然であったが, 曲線を入力する際に, 前後4小節で似た構成になるよう考慮すれば, 曲中に不自然な部分が出力されなくなることが分かった.

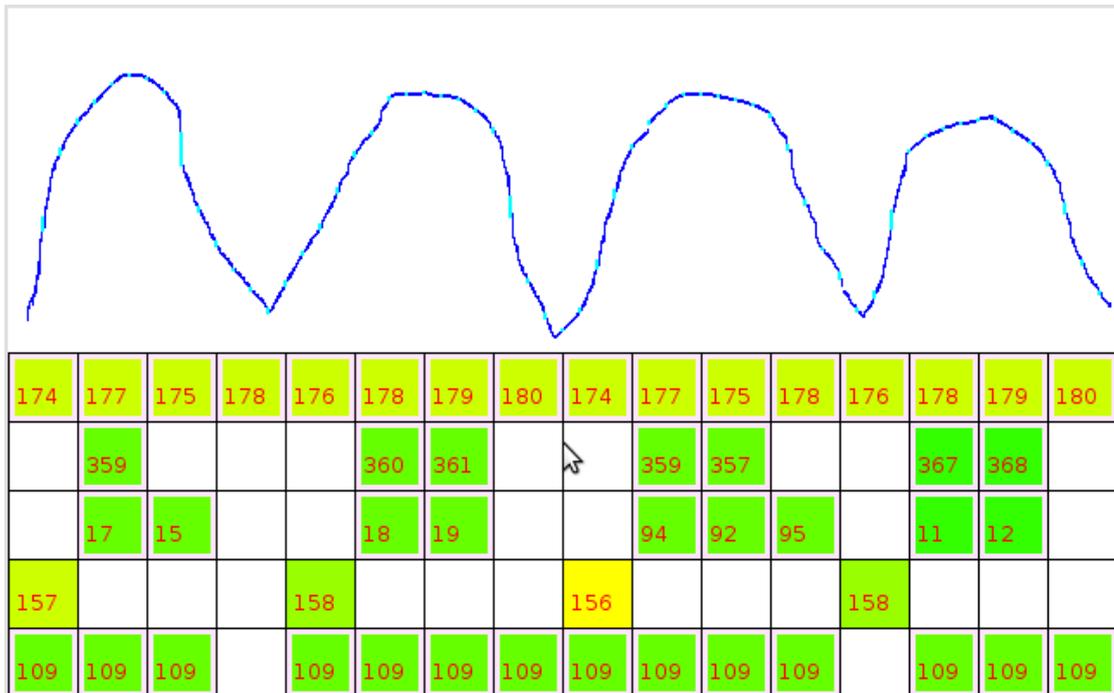


図 4.4: 実行例3の出力結果画面

4.2.2 編集機能による試用結果

実行例 4

ループシーケンサーの編集機能を使用する。編集機能の一つ、配置された素材を消去する消しゴムを用いて検証する。

まず、曲線を入力し、素材が配置された状態の図 4.5(a) を用意した。(a) を編集し、9 小節目、Bass の部分を消去したものが図 4.5(b) である。編集を行ったところ、10 小節目以降に対して自動推薦が行われた。Bass の 10~12 小節、13~16 小節で推薦された素材が変化していた。

更に編集を行い、(b) の 5 小節目、Bass の部分を消去したものが図 4.5(c) である。6 小節目以降を見ると (b) と同じく自動推薦が行われ、4 小節ごとに挿入される素材が変化していた。10 小節目では、(b) の編集によって素材配置数が 5 → 4 となっていたが、(c) の編集後に素材数は 4 のまま、自動推薦によって挿入される素材やその配置が変化した。消しゴムでの編集によって盛り上がり度も同時に変化したため、編集した小節が含まれる 4 小節内でより盛り上がりの低い素材が挿入されるよう、自動推薦が行われていた。

(c) さらに 5 小節目の Bass を削除した結果

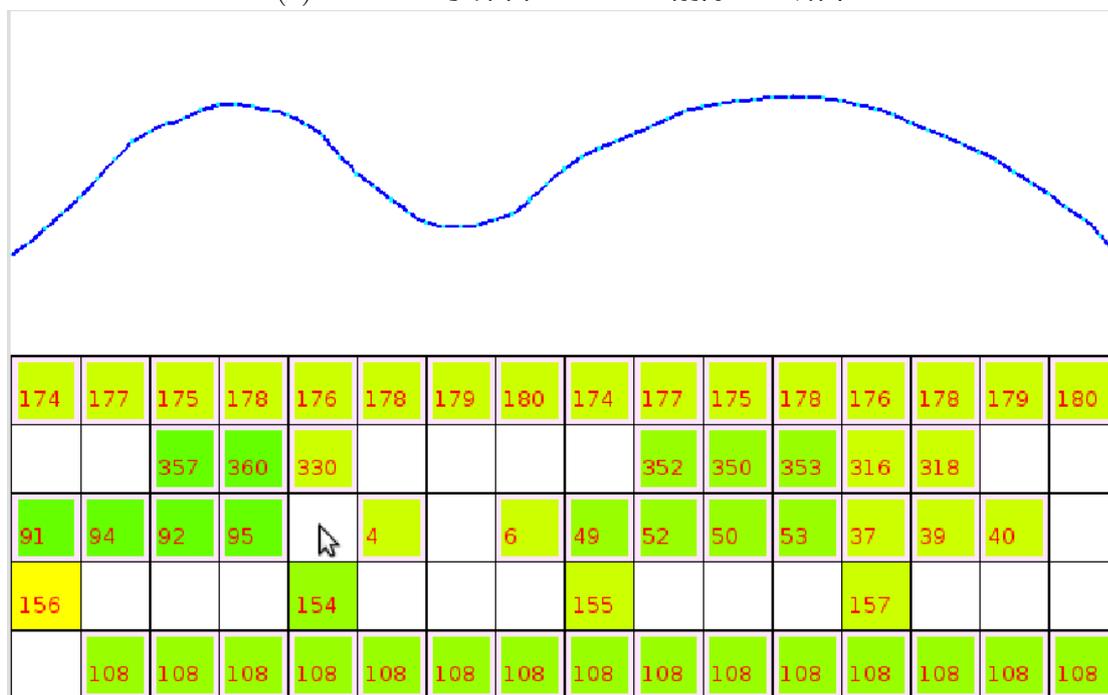


図 4.5: 実行例 4 の出力結果画面

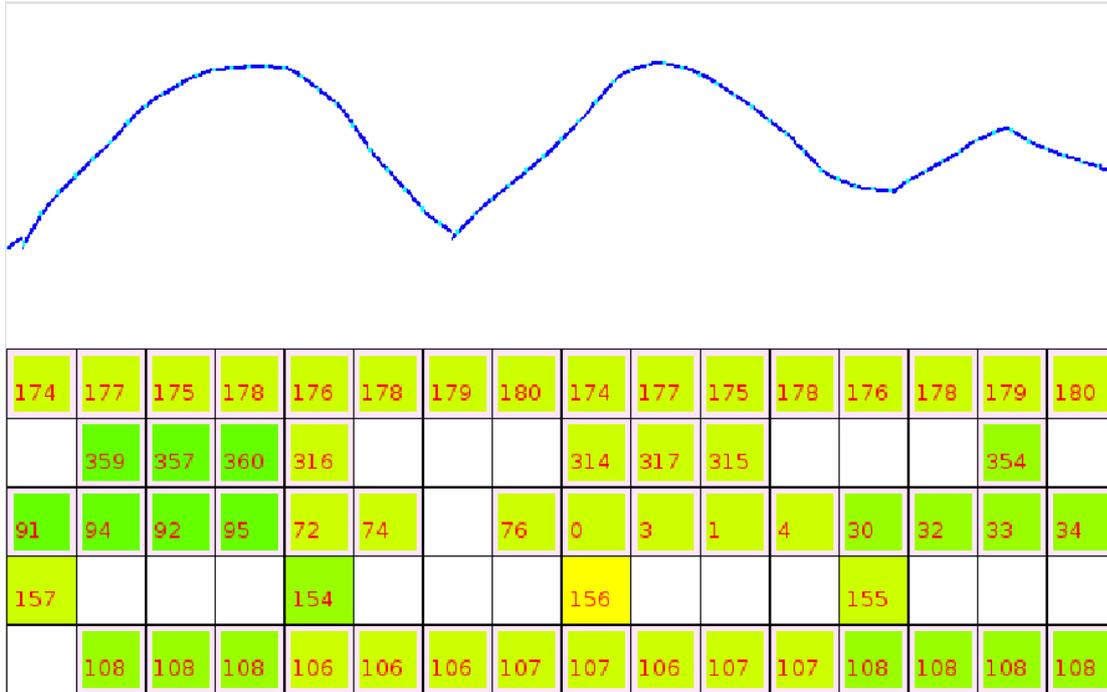
実行例5

もう一つの編集機能, 素材の変更を行う. 配置場所を選択し, ユーザの選んだ素材を挿入する事が出来る.

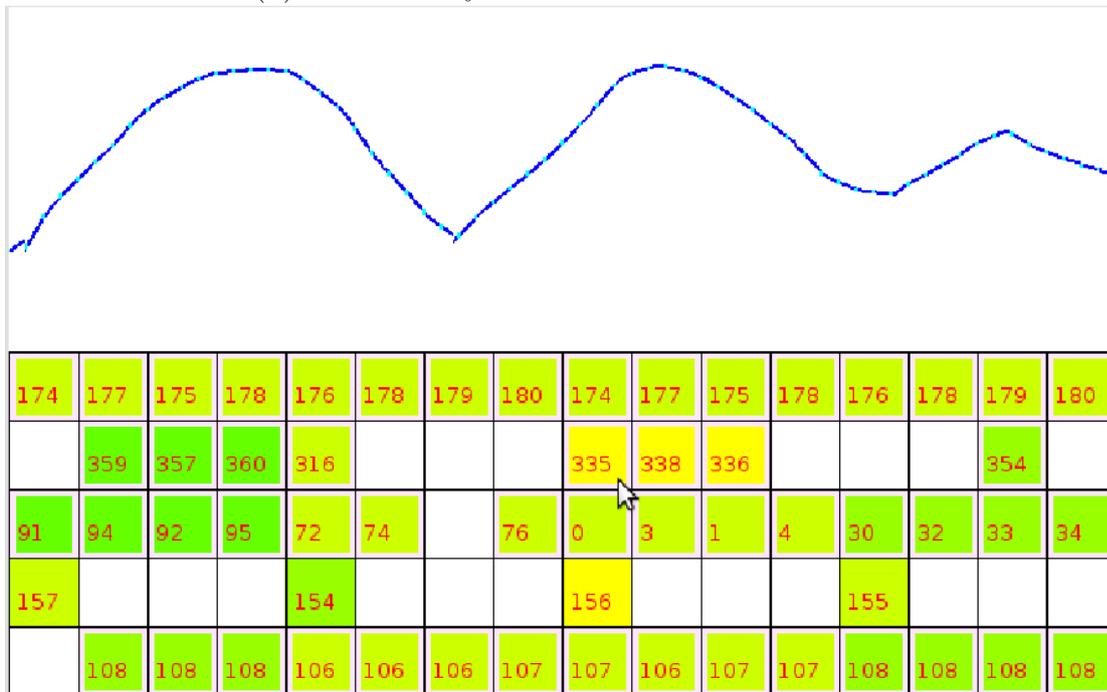
9小節目, Synthの部分に挿入されている「314: AlarmLead 1」を違う素材「335: Lead 1」に変更した. 編集前後を比較したものが図4.6(a), (b)である. 実行例4にて消しゴム編集した結果と同様に, 編集した小節以降で自動推薦が行われた. 編集に用いた2つの素材は, メロディは異なるが盛り上がり度が同じ値のため, 同じ種類の素材が推薦された.

次に, 10小節目, Bassの「3: DanceOff 3」を「42: OfflineB 7」に変更した. 編集したものが図4.6(c)である. この場合, 11小節目から自動推薦が行われたにもかかわらず, 9~12小節のなかで編集を行ったSynthのみ素材の種類が変わっており, それ以外は13小節目から変化した. よって, 区切られた4小節中で最初の1小節目が編集されない場合でも, 他のパートに影響を及ぼすことなく編集する事が出来た.

(a)9小節目のSynthの音素材を変更する前



(b)9小節目のSynthの音素材を変更した結果



(c) さらに10小節目の Bass の音素材を変更した結果

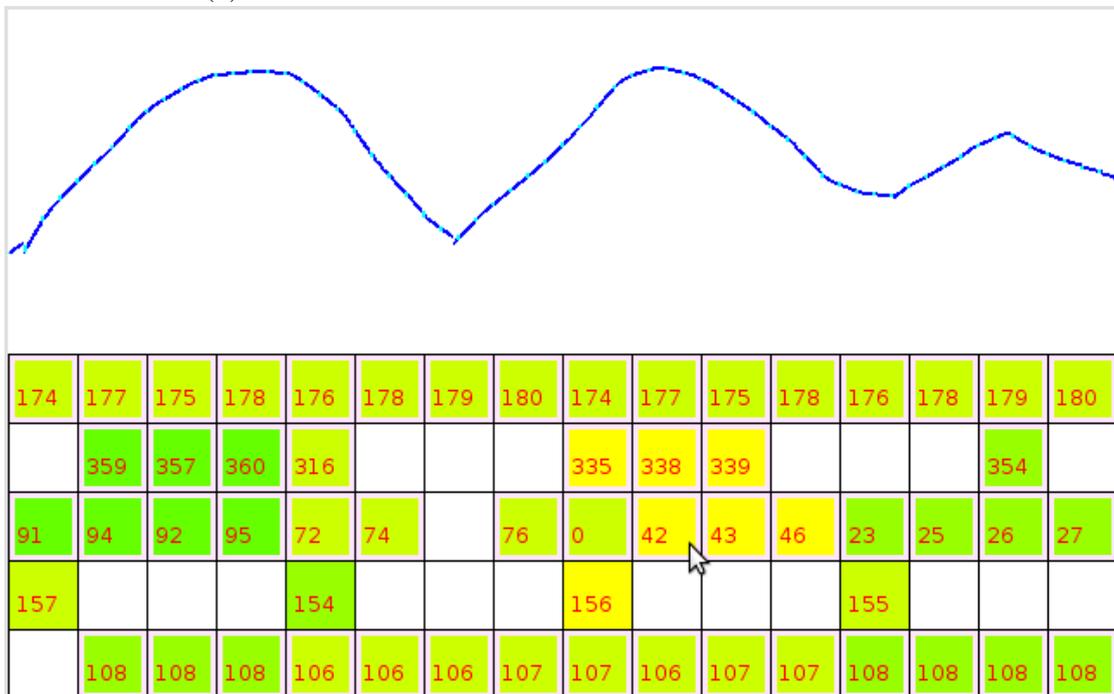
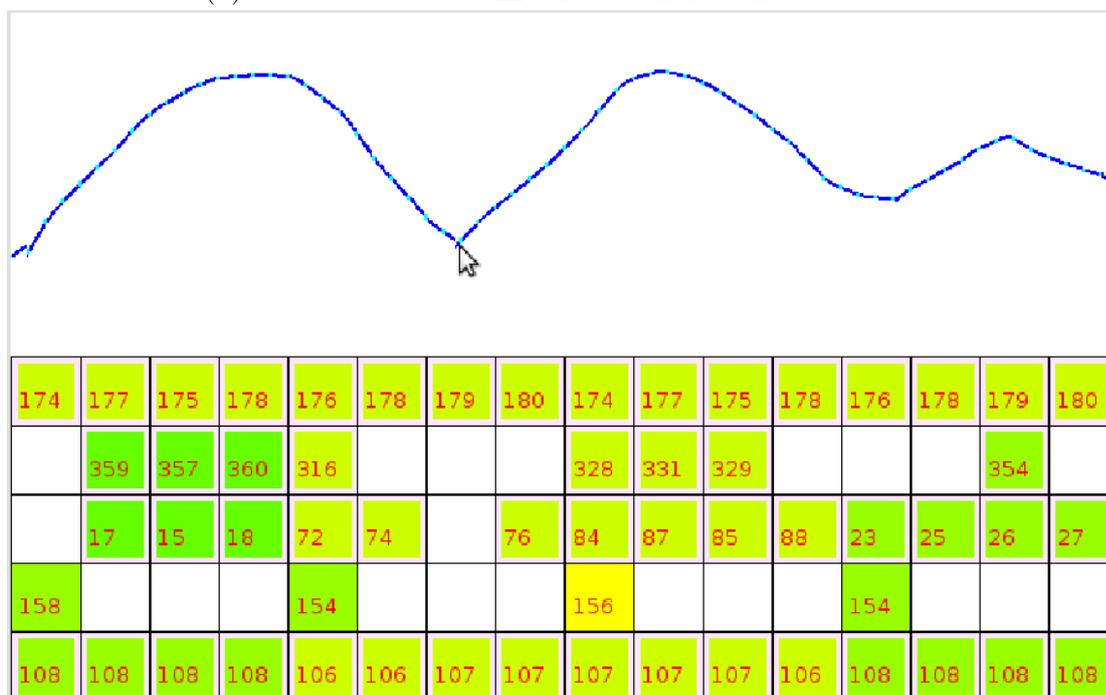


図 4.6: 実行例5 の出力結果画面 2

実行例 6

実行例 5 の編集機能から発展させ、素材の追加挿入を行う。素材が配置されていない空の部分を選択しても、ユーザの選んだ素材を挿入出来る。曲線を入力した状態から、6 小節目、Bass の部分に「22: MaasedBass 1」を挿入した。編集したものが図 4.7 である。この場合も実行例 5 と同様に、編集した 6 小節目以降に対して自動推薦が行われた。

(a)6 小節目の Bass の空の部分に音素材を挿入する前



(b)6小節目の Bass の空の部分に音素材を挿入した結果

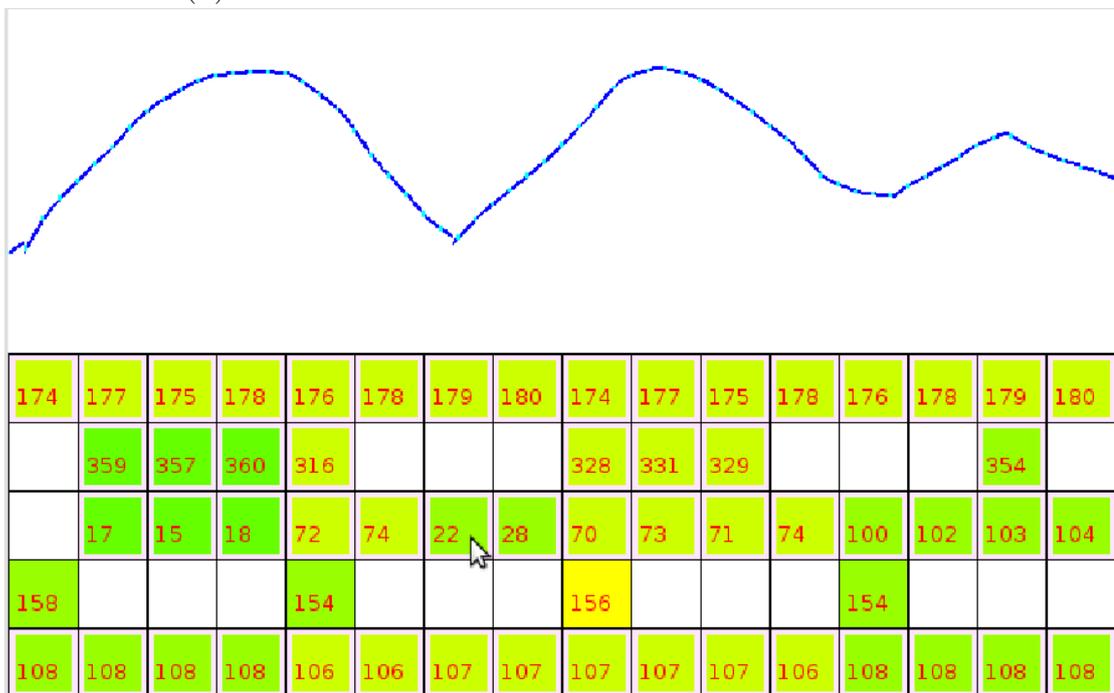


図 4.7: 実行例 7 の出力結果画面

第5章 結 論

作曲初心者を対象としたループシーケンサを実装するにあたって、作曲初心者は明確で詳細なメロディーよりも、曲の盛り上がりといった、曲の雰囲気や想定していると考えられる。そこで本研究では、曲の盛り上がりに着目し、これをマウス入力によって曲線のグラフでユーザが描くことによって、その盛り上がりに沿った曲が出力されるループシーケンサを提案した。

対象とするユーザは作曲初心者とするため、音楽理論の知識や煩雑なパラメータ設定を必要とせず直感的に作曲を行ってもらう必要がある。そこで、ユーザが曲線グラフを描いた段階で用意した音素材を自動的に推薦し出力することで、最初に曲を完成をさせることにした。完成された曲を出力してしまうことで、どこから手をつけて作曲すればよいのか分からない状態を回避する。

本研究のシステムで取り組む問題は、ユーザが入力したグラフの曲の盛り上がりやどのようにより反映させるかである。これは、まず曲の盛り上がりは音源の個数の重なりによって表現ができると考え、状態遷移表を用いることで、音素材の配置と重なる個数を決定した。しかし、各音素材を単体でみたとき盛り上がりの度合はそれぞれ違っているため、音源の重なり個数の決定だけでは曲の盛り上がりや表現するのは困難であると考えた。そこで、各音素材に盛り上がり度合いの値を付与し(データベースを作成した)、自動出力の際に選ばれる音素材を制限することで、曲の盛り上がりや表現することを実現した。

現段階では、この自動出力の際に使用している状態遷移表や各音素材の盛り上がり度のデータベースは、手動で作成している。そのため、ユーザによっては満足で

きない遷移や音素材が出力される可能性がある。今後の課題は、評価実験を通じてデータベースの改良を重ね、自動出力の段階での曲の完成度を上げていきたい。

また、ユーザが気に入った音素材や組み合わせを記録して、自動推薦のときに優先的に出力されるシステムを考えている。これによって、自動出力される曲はユーザが望むものに一層近づくと考えられる。さらに、このお気に入りの音素材の組み合わせをインターネットを通じて他者と共有し、自身のデータベースを充実させることで、自動出力される曲の更なる完成度の向上、作曲を行う上での新たな発見が期待できる。

参考文献

- [1] プロフェッショナル用デジタルオーディオワークステーション「Acid Pro」
<http://www.sonycreativesoftware.com/acidpro>
- [2] 徳丸正孝, 大竹孝昌, 村中徳明, 今西茂: 音楽で「遊ぶ」ことを目的とした作曲システムの構築に関する検討, 感性工学研究論文集 vol5, pp.45-52, 2005.
- [3] Musicshake, Inc. : Musicshake
- [4] 株式会社イーフロンティア: Band in a Box.
- [5] 諸星和明, 椎塚久雄: 発想支援法を用いた作曲支援システムについて : 発想支援法, 工学院大学研究報告 101, pp.131-134, 2006-10-30.
- [6] 佐藤剛, 宮下芳明: Seek Rope : 曲げて切って結べるシークバー, インタラクシオン 2010 論文集, 2010.
- [7] 松村智弘, 中村滋延: Au-thello : ゲーム「オセロ」をモデルとした音楽演奏コントローラ, 情報処理学会研究報告, 2009-11-28.
- [8] 大島千佳, 西本一志, 宮川洋平, 白崎隆史: 音楽表情を担う要素と音高の分割入力による容易なシーケンスデータ作成システム, 情報処理学会論文誌 44, 2003-7-15.
- [9] E Costanza, SB Shelley, J Robinson: D-touch : A consumer-Grade Tangible Interface Module and Musical Applications, Proceedings of Conference on Human-Computer Interaction, 2003.

[10] <http://processing.org/>

[11] 音楽情報処理システム開発のための Java ライブラリ「CrestMuse Toolkit」
<http://cmx.sourceforge.jp/>

[12] 著作権フリーの音楽素材集「Sound PooL」
<http://www.ah-soft.com/soundpool/>

謝 辞

本研究は日本大学文理学部情報システム解析学科北原研究室の卒業研究の研究テーマとして実施されました。本研究を進めるにあたり、ご指導を頂いた卒業論文指導教員の北原鉄朗先生に感謝致します。また、日常の議論を通じて多くの知識や示唆を頂いた北原研究室の同期・後輩に感謝します。