

音素材の自動挿入機能を備えたループシーケンサ

岡田 美咲¹ 山下 雄史¹ 北原 鉄朗¹

概要：本稿では、テクノミュージックを対象に、ユーザがマウスで楽曲の「盛り上がり度」の時間軌跡を描くと、それに合わせて自動的に音素材を挿入するループシーケンサを提案する。ループシーケンサは、あらかじめ用意された音素材を組み合わせるだけで楽曲制作ができるため、音楽知識が不要という点では手軽であるが、膨大な音素材がある場合に、そこから自分が目的とするものを選び出すのは困難である。テクノミュージックでは、明確な主旋律がない代わりに、音素材の追加や削除を繰り返しながら楽曲が進行することで楽曲の盛り上がり方が変化し、これによって起承転結を表現することが多い。そのため、楽曲が進行とともにどのように盛り上がっていくかは、聴感上最も重要な要素の1つと言える。本研究では、この楽曲の盛り上がり方をマウスで描画するだけで、自動的に音素材を挿入する手法を提案する。これは、隠れマルコフモデルを簡略化したモデルによって実現する。システムを実装・試用したところ、ある程度妥当な楽曲が生成されることを確かめられた。

1. はじめに

楽曲制作を目的とするソフトウェアの1ジャンルに、「ループシーケンサ」というものがある。これは、1~2小節程度の音素材を重ねたりつないだりすることで楽曲を制作するソフトウェアである。音素材は通常オーディオデータとして用意されるため、MIDIシーケンサで1つ1つ音符を打ち込むのに比べて、特別な音楽知識を必要としないにも関わらず、完成度の高い楽曲を制作することができる。ループシーケンサの例としては、ACID PRO 7[1]やGarageBand[2]などが知られている。

ループシーケンサは、その特性上、幅広い種類の楽曲の制作を可能にするために、膨大な音素材を揃える必要がある。たとえば、前述のACID PRO 7では3000を超える音素材が用意されている。しかしながら、3000を超える音素材を一通り試聴し、用意されている音素材の傾向を把握し、どの音素材をどのように組み合わせるべきかを考え、実際に適した音素材を選ぶのは、決して容易ではない。そのため、音楽の専門知識を必要としないという特徴があるにも関わらず、音楽の非専門家がループシーケンサを使って楽曲制作を楽しむというのは、あまり一般的ではない。

これまで、自動作曲や作曲支援については、様々な研究がなされてきた。自動作曲に関しては、確率モデルを用いたもの[3]や対話的進化計算を用いたもの[4]など数多くの研究事例が存在する。作曲支援については、旋律の概形をマウスで描画することで旋律を生成するもの[5]や、複

数のメロディを混ぜ合わせて新たなメロディを生成するもの[6]などがあげられる。しかし、これらはいずれも音符レベルの楽曲を対象としており、ループシーケンサのようにオーディオの音素材を扱うものではない。また、近年になってオセロゲームや紐といった身近なものをメタファとしたり、(画面上ではなく)有形のオブジェクトを操作対象とすることで、音楽の非専門家でも容易に習得可能な音楽制作環境の実現を目指した研究も行われている[7], [8], [9]。しかし、これらは通常のループシーケンサとは異なる考え方に基づくものである。このように、ループシーケンサに着目し、膨大な音素材からの探索や選択を容易にすることを目的とした研究は非常に少ない。その少ない研究の例としては、創造学の一部である発想支援法を用いて音素材の選択を支援することを試みたもの[10]、音響的特徴に基づいて音素材の音響的類似度などを可視化して音素材の探索を支援するもの[11]などがあげられる。

本研究では、対象楽曲をテクノミュージックに絞り、ユーザが入力した「盛り上がり度」の軌跡に応じて自動的に音素材を挿入するループシーケンサを提案する。ユーザは、楽曲の開始から終了までにどのように楽曲が盛り上がるかをマウスを使って描画する。するとシステムは、その「盛り上がり度」の軌跡をできるだけ再現するように、音素材を自動的に決定する。テクノミュージックは、明確なメロディがない代わりに、音素材のオンオフや選び方で盛り上がりを制御し、起承転結を表現することが少なくない。そのため、「盛り上がり度」はテクノミュージックにおいて最も印象の強い音楽的要素の1つと考えられる。本システ

¹ 日本大学文理学部

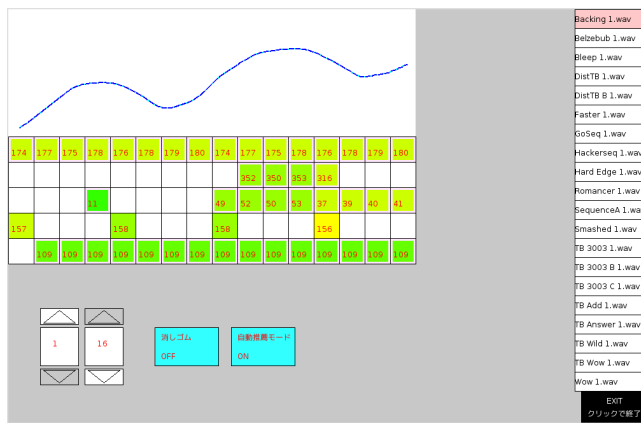


図 1 本システムの画面

ムにおいて入力項目として「盛り上がり度」を採用したのは、このような理由によるものである。本稿では、本システムの概要、音素材の自動挿入手法、本システムの試用結果について報告する。

2. 本システムの概要

本システムの作曲画面を図 1 に示す。この画面では、メインとなる音素材のリストが右側に表示され、左側に楽曲制作のためのパネルが存在する。パネルは上下に 2 分割されており、上部に「盛り上がり度」、下部に挿入された音素材が表示されるようになっている。

まず、曲のメインとなる音素材を画面右側のリストから試聴しながら選択する。選択を終えたら、パネル上部に、楽曲の「盛り上がり度」の時間変化を、マウスのドラッグ操作によって描く。すると、パネル下部の格子状の枠内に音素材が自動的に挿入される。この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(画面下の削除機能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。

3. 音素材の自動挿入手法

3.1 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に合わせて音素材を挿入する処理を、隠れマルコフモデル(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。あらかじめいくつかのパート(Drums, Percussion, Bass, Synth など)が設定されており、音素材があらかじめパートごとに分類されているものとする。

パート i 用の音素材の集合を \mathcal{M}_i と書くものとする。また、Drums, Percussion 以外のパートについては、(コード以外の)演奏内容が同一の音素材が、全ダイアトニックコード分用意されているものとする。これらは同一の音素材番号で表すものとする。そのため、これらのパートでは、音素材番号 $m(\in \mathcal{M}_i)$ とコード名 $c(\in \mathcal{C})$ (\mathcal{C} はダイアトニックコードの集合)の組 (m, c) で音素材が一意に特定されることとなるが、簡単のためコード進行を固定*1 とするため、コード名 c は省略する。

音素材を自動的に挿入する問題は、 n 小節めのパート i に挿入する音素材を $s_{n,i}$ とすると、各小節に対して、最適な $s_n = (s_{n,1}, \dots, s_{n,I})$ ($s_{n,i} \in \mathcal{M}_i$) を推定することである。ここで、どの音素材も挿入しないことを $s_{n,i} = 0$ と書くこととする。本研究では、楽曲の盛り上がり度の時系列がユーザによって入力されることを想定するので、各小節の盛り上がり度 $\mathbf{x} = [x_1, \dots, x_N]$ が与えられたときに、それに最適な音素材の組み合わせ $\mathbf{S} = [s_1, \dots, s_N]$ を推定すればよいこととなる。確率を用いて表せば、

$$\hat{\mathbf{S}} = \underset{\mathbf{S}}{\operatorname{argmax}} P(\mathbf{S}|\mathbf{x})$$

を求める問題と行うことができる。なお、楽曲の盛り上がり度は、図 1 のようにフリーハンドで描画されることを想定しているため、実際には 1 小節よりも細かい単位時間で与えられるものと考えられるが、小節ごとに平均を取るなどして 1 小節ごとの値の系列として扱うものとする。

ベイズの定理より、 $\hat{\mathbf{S}} = \underset{\mathbf{S}}{\operatorname{argmax}} P(\mathbf{x}|\mathbf{S})P(\mathbf{S})$ であり、また、 x_n が s_n にのみ依存して決まり、 s_n も s_{n-1} のみに依存して決まるとすると、

$$\hat{\mathbf{S}} = \underset{s_1, \dots, s_N}{\operatorname{argmax}} \prod_{n=1}^N P(x_n|s_n) \cdot P(s_1) \prod_{n=2}^N P(s_n|s_{n-1})$$

となる。これは、 x_1, \dots, x_N を観測系列(出力系列)、 s_1, \dots, s_N を状態遷移系列とみなした HMM に他ならない。そのため、 $P(x_n|s_n)$ および $P(s_n|s_{n-1})$ を適切に設定または学習することができれば、Viterbi アルゴリズムを用いて最ももつともらしい s_1, \dots, s_N を決定することができる。

3.2 モデルの簡略化

前節では音素材の自動決定問題を HMM によって定式化した。これを直接実装するには、膨大なパラメータを決定する必要がある。状態遷移確率 $P(s_n|s_{n-1})$ だけでも、 s_n は $\prod_{i=1}^N (|\mathcal{M}_i| + 1)$ 通りの値を取りうるので、 $\left\{ \prod_{i=1}^N (|\mathcal{M}_i| + 1) \right\}^2$ 個のパラメータを持つこととなる。これだけ多くのパラメータを人手で適切に決定することも

*1 原理的にはコード進行をユーザが選択あるいは入力できるようにするのは可能であるが、現在の実装では 1 通りのコード進行で固定されている。

データから学習することも、現実的とはいえない。そこで、モデルの簡略化を行い、パラメータ数を削減する。

導入する簡略化を以下に示す。

- (1) 盛り上がり度 x_n を離散化し、分解能をパート数 I (現在の実装では 5) と等しくする。
- (2) 音素材を挿入するパートの個数は、その小節の盛り上がり度と等しいものとする。たとえば、ある小節の盛り上がり度を「3」とすると、その小節には 3 つのパートに音素材を挿入するものとする。このとき、どのパートに音素材を挿入するかは、直前の小節の状態に依存して決まるものとする。
- (3) 各パート用の音素材を盛り上がり度ごとに分類する。たとえば、パート i , 盛り上がり度 k 用の音素材の集合を $\mathcal{M}_{i,k}$ とすると、 x_n が k である小節のパート i に挿入する音素材は $\mathcal{M}_{i,k}$ から選ばれるものとする。
- (4) パート間における音素材の選び方の依存関係は考慮しない。

以上の簡略化を確率に基づいて議論するため、 n 小節めのパート i に挿入する音素材の決定を、当該小節・当該パートに「音素材を挿入するかどうか」と「音素材を挿入するならば何を挿入するか」に分割して考え、前者を $q_{n,i}$ (挿入するなら 1, 挿入しないなら 0), 後者を $s'_{n,i} (\in \mathcal{M}_i)$ という確率変数で表す。簡単のため、これらを独立とみなすと、 n 小節めのパート i に $m (\in \mathcal{M}_i)$ という音素材が挿入される確率は、

$$\begin{aligned} P(s_{n,i} = m) &= P(q_{n,i} = 1, s'_{n,i} = m) \\ &= P(q_{n,i} = 1) P(s'_{n,i} = m) \end{aligned}$$

と表される。

この確率変数を用いて $P(x_n | s_n)$ を展開すると、簡略化 (4) より $s'_{n,1}, \dots, s'_{n,I}$ はそれぞれ独立であるので、

$$P(x_n | s_n) = \alpha P(x_n | \mathbf{q}_n) \prod_{i=1}^I P(x_n | s'_{n,i})$$

と表される (ここで、 $\mathbf{q}_n = (q_{n,1}, \dots, q_{n,I})$ と置いた。また、 α は定数)。

$P(x_n | \mathbf{q}_n)$ は、音素材を挿入するパートと盛り上がり度の関係を表しており、簡略化 (1) および (2) より、

$$P(x_n = k | \mathbf{q}_n) = \begin{cases} 1 & (q_{n,1}, \dots, q_{n,I} \text{ のうち } k \text{ 個が } 1 \text{ で、残りが } 0) \\ 0 & (\text{それ以外}) \end{cases}$$

と定義する。

$P(x_n | s'_{n,i})$ は、挿入される音素材の内容と盛り上がり度の関係を表し、簡略化 (3) より、

$$P(x_n = k | s'_{n,i} = m) = \begin{cases} 1 & (m \in \mathcal{M}_{i,k}) \\ 0 & (m \notin \mathcal{M}_{i,k}) \end{cases}$$

と定義する。これにより、たとえば指定された盛り上がり度が「3」だとすると、3 つのパートにのみ音素材が挿入され、かつ、そのすべてが、盛り上がり度「3」用に用意された音素材であるときのみ、 $P(x_n | s_n)$ が 0 より大きい値を取る。

$P(s_n | s_{n-1})$ も同様に展開すると、

$$P(s_n | s_{n-1}) = P(\mathbf{q}_n | \mathbf{q}_{n-1}) \prod_{i=1}^I P(s_{n,i} | s_{n-1,i})$$

となる。ここで、 $P(\mathbf{q}_n | \mathbf{q}_{n-1})$ は、音素材をどのパートに挿入するかを制御するパラメータで、たとえば時刻 $n-1$ において Drums と Bass に音素材が挿入されており、時刻 n に、それに加えてもう 1 パートに音素材を挿入する場合には、Percussion に挿入するのがよいか Synth に挿入するのがよいかを計算するのに使用する。ここでは簡単のため、 \mathbf{q}_{n-1} および x_n が与えられたときに最ももらしい \mathbf{q}_n を書き下した「状態遷移表」(次章で具体例の一部を示す) と呼ばれる表を手で作成し、決定論的に状態遷移するものとする。すなわち、

$$P(\mathbf{q}_n | \mathbf{q}_{n-1}) = \begin{cases} 1 & (\text{状態遷移表に記述された } \mathbf{q}_n \text{ と一致}) \\ 0 & (\text{それ以外}) \end{cases}$$

である。

$P(s_{n,i} | s_{n-1,i})$ は具体的にどの音素材を挿入するかを定めるパラメータであるが、これについては、次節で述べる。

3.3 一貫性のある楽曲生成のための工夫

前節にて $P(s_{n,i} | s_{n-1,i})$ 以外は定義したので、与えられた盛り上がり度の時系列 $[x_1, \dots, x_N]$ に対して、どのパートに音素材を挿入するか $[\mathbf{q}_1, \dots, \mathbf{q}_N]$ は決定できるようになった。ここでは、具体的にどの音素材を挿入するかを決定する方法について議論する。ここで重要なことは、単調さを避けつつも一貫性のある音素材を選択することである。最初から最後まで全く同じ音素材が続けば、音素材のオンオフがあるとしても、単調さは免れない。一方、すべての小節に全く特徴が異なる音素材が挿入されれば、一貫性のない楽曲になってしまう。そこで、我々は次の 5 つのルールを導入する。

- (1) 楽曲を 4 小節ごとに区切り、4 小節をひとまとまり (以下、セクションと呼ぶ) として扱う。
- (2) 各パートにおいて、同一セクション内では同じ番号の音素材が用いられる。
- (3) 聴感上最も目立つ高音部のパート (パート番号を 1 とする) については、ユーザが最初に音素材を選択するものとし、楽曲の最初から最後までその音素材が挿入されるものとする。
- (4) Percussion (パート番号を I とする) については、次

の特別ルールを適用する。

- スネアドラムの8部音符連打のようにフィルインとしての役割を担うものは、各セクションの最終小節にのみ挿入できる。
 - 小節の先頭にクラッシュシンバルが鳴るものは、各セクションの最初の小節にのみ挿入できる。
(これらのいずれかに当てはまらないものは扱わないものとする)
- (5) 以上のルールによって音素材の候補が一意に決まらない場合は、当該候補からランダムに決定する。その際の確率は等確率とする。

以上のルールを満たすように $P(s_{n,i}|s_{n-1,i})$ を定義すると、次のようになる。

- $i = 1$ のとき

$$P(s_{n,1} = m|s_{n-1,1}) = \begin{cases} 1 & (m \text{ が, ユーザが選んだ音素材と等しい}) \\ 0 & (\text{それ以外}) \end{cases}$$

- $1 < i < I, n = 4n' + 1$ (n' は任意の自然数) のとき

$$P(s_{n,i} = m|s_{n-1,i}) = \begin{cases} 1/|\mathcal{M}_i| & (m \in \mathcal{M}_i) \\ 0 & (m \notin \mathcal{M}_i) \end{cases}$$

- $1 < i < I, n \neq 4n' + 1$ のとき

$$P(s_{n,i} = m|s_{n-1,i} = m') = \begin{cases} 1 & (m = m') \\ 0 & (m \neq m') \end{cases}$$

Percussion パート ($i = I$) についても同様に、上のルールを満たすように定義することができる (省略)。

3.4 実際の音素材挿入アルゴリズム

以上のようにモデルを簡略化し、一定のルールを導入することで、パラメータ数を大幅に削減することができる。具体的には、 $I \times 2^I$ 個の項目からなる状態遷移表を作り、すべての音素材を I 段階の盛り上がり度ごとに分類するだけで、処理を行うことができる。また、各セクションの最初の小節の音素材がランダムに選ばれる以外は、小節毎に解が一意に定まるので、Viterbi アルゴリズムなどを用いずに解を求めることができる。具体的なアルゴリズムは、次の通りである。

- (1) 入力された盛り上がり度 $\mathbf{x} = [x_1, \dots, x_N]$ から、小節毎に挿入する音素材の個数を決定する。
- (2) 状態遷移表から、小節ごとに音素材を挿入すべきパートを決定する。具体的には、 x_1 から \mathbf{q}_1 を決定したのち、状態遷移表における (\mathbf{q}_{n-1}, x_n) の項目を参照して、 \mathbf{q}_{n-1} と x_n から \mathbf{q}_n を決定する処理を、最後の小節まで再帰的に繰り返す。
- (3) 3.3 節で導入した5つのルールに従って、各パート・各小節に挿入する音素材を決定する。

```
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、Drums, 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 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。

4. 本システムの実装・試用結果

4.1 実装

2~3章で提案したシステムをProcessing[12]上で実装した。WAVの読み込み・合成には、CrestMuse Toolkit[13]を用いた。音素材は「Sound Pool」[14]という素材集から抜粋した。パートはSequence, Synth, Bass, Percussion, Drumsの5つとし、Sequenceを3.3節における「聴感上最も目立つ高音部のパート」とした。状態遷移表(図2)および各音素材の盛り上がり度(図3)は、第2著者が手動で作成・付与した。コード進行は、[Cm, Fm, Ddim, Gm, Eb, Gm, Ab, Bb]の繰り返しとした。図4にシステムの流れを示す。

4.2 試用結果

本節では、実装したシステムを実際に試用して得られた楽曲について考察する。以下、楽曲の長さは16小節とする。

4.2.1 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した。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。

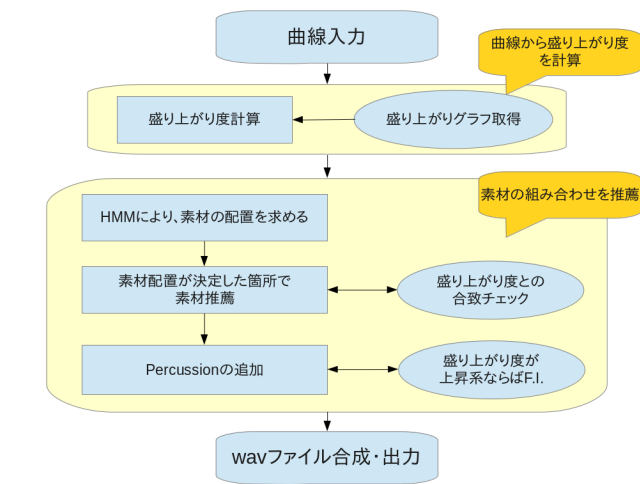


図 4 システムの流れ

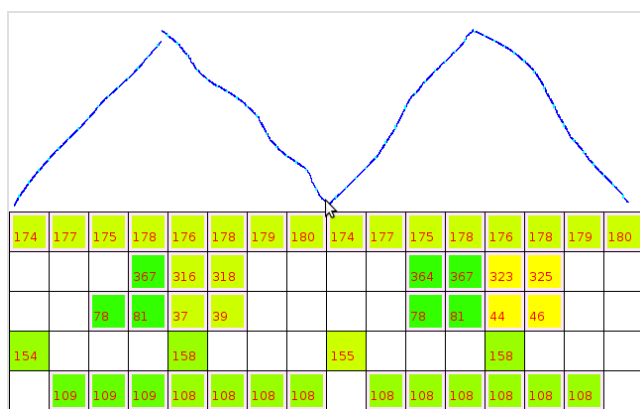


図 5 実行例 1 の出力結果画面

- 残り 8 小節でも同様に繰り返す。

その出力結果を図 5 に示す。図 5 の下半分に表示されている番号が、音素材の通し番号である。2 章で定義した M_i や $M_{i,k}$ と異なり、全音素材に対する単純な通し番号であるので、コードだけが異なる音素材に対しても異なる番号が振られている。たとえば、一番上のパート (Sequence) の最初の 4 小節は [174, 177, 175, 178] となっているが、これらはコード以外の演奏内容はいずれも同一である。

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 のみ最初から最後まで同じ音素材が再生されるようにしたことで、一貫性のある曲に仕上がった。

4.2.2 実行例 2

次に、盛り上がりの変化に緩急がある例を示す。この例において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。

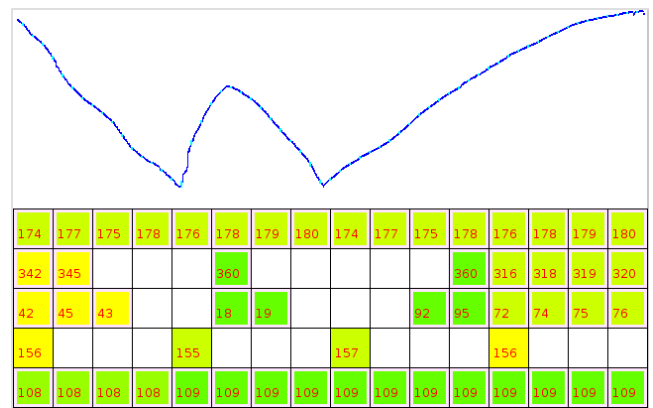


図 6 実行例 2 の出力結果画面

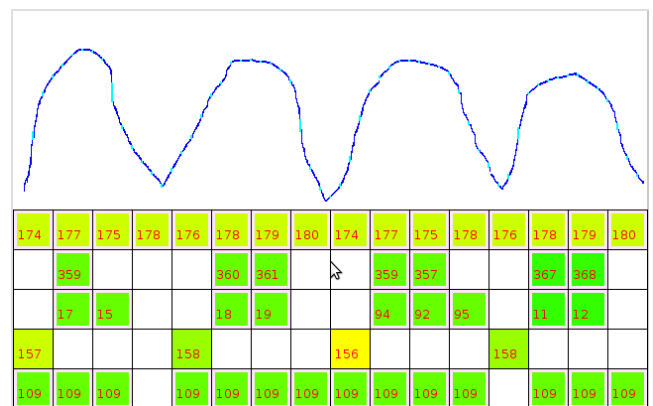


図 7 実行例 3 の出力結果画面

その出力結果を図 6 に示す。盛り上がり度が細かく変化する 5~8 小節め以外については、連続で挿入される素材が多く、変化の少ない曲となった。一方、5~8 小節めについては、入力した曲線のとおり音素材が挿入されたため、唐突に 1 小節だけ再生される素材が出現した。これ自体はユーザの指定通りの結果であるものの、その部分のみが曲の流れにそぐわず若干不自然であった。今後は、1 小節のみ挿入する場合には、フィルイン的な特徴の素材を優先するなど、大局的な盛り上がり度の変化を考慮して音素材を選ぶ仕組みを導入することが必要といえる。

4.2.3 実行例 3

次に、急激に盛り上がり度が上下するのを繰り返すようにした場合について述べる。4 小節中で盛り上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、それ自体が特徴となり、決して不自然ではない楽曲になることが分かった。

5. おわりに

本稿では、テクノミュージックを対象を限定し、「曲の盛り上がり度」という新たな指標をユーザに入力させることで、それに合わせて自動的に音素材を挿入するループシー

ケンサを提案した。曲の盛り上がり度から挿入すべき音素材を決定する処理は、隠れマルコフモデル (HMM) によって定式化し、その後、パラメータ数削減のために簡略化したモデルを提案した。実際に楽曲生成を試したところ、ユーザが入力した盛り上がり度が反映され、一定範囲で妥当な楽曲が生成された。

本稿で提案した手法は、パラメータ数の削減に重きを置いたために、ほとんど決定論的に動作しており、確率モデルの良さが活かされているとは言い難い。今後は、パラメータ数を抑制しながら確率モデルの利点を活かせるよう、モデルの簡略化方法などを改良する。また、本システムを Web サービス化することで使用履歴を取得・収集してパラメータの学習に用いる試みも進めていきたい。

参考文献

- [1] <http://www.sonycreativesoftware.com/acidpro>
- [2] <http://www.apple.com/jp/ilife/garageband/>
- [3] 深山 寛, 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, 2008.
- [4] 安藤 大地, P. dahlstedt, M. G. Nordaxhl, 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] 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 (テクニカルノート), Vol.54, No.4, pp.1302-1307, 2013.
- [6] 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.
- [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-MUS-83-12, 2009.
- [8] 青木 惇季, 宮下 芳明: SeekRopes: 複数スライダとシークロープによる音楽制作, インタラクション 2011, 2011.
- [9] E. Constanza, S.B. Shelley, and J. Robinson: D-touch: A Consumer-grade Tangible Interface Module and Musical Applications, *Proceedings of International Conference on Human-Computer Interaction*, 2003.
- [10] 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 Vol.101, pp.131-134, 2006.
- [11] S. Sreich and B. S. Ong: A Music Loop Explorer System, *Proceedings of International Computer Music Conference*, 2008.
- [12] <http://processing.org/>
- [13] <http://cmx.sourceforge.jp/>
- [14] 著作権フリー音楽ループ素材集「SoundPool」 vol.1, <http://www.ah-soft.com/soundpool/>