Vol.2019-MUS-122 No.2
Vol.2019-EC-51 No.2
2019/2/22

多面体を用いた楽曲の自動生成について

濱岡　智恵[†]，　川島　正行[‡]，
岡山理科大学大学院 総合情報研究科[†]，岡山理科大学 総合情報学部[‡]

1　はじめに

本研究では, 楽曲から多面体を構成した先行研究をもとに, 音の数から多面体を構成し, そこから楽曲の自動生成を試みる.

楽曲と多面体を繋げる重要な道具として、トライアド・コードによって構成される Tonnetz を用いる. 本研究の目標は音楽理論に詳しくない人でも多面体の形から楽曲の生成を可能とすることである.

2　先行研究について

本研究ではこれまでに、以下の手順で MIDI ファイルから $\mathbb{R}^3$ 内の多面体を構成した.

1. Tonnetz を $\mathbb{R}^3$ 内の z 座標が 0 の xy 平面におく.
2. MIDI ファイルから音の数・長さを読み込む.
3. Tonnets の各音の z 座標を、読み込んだ音の数と長さとして表現する.
4. このようにしてできた $\mathbb{R}^3$ の部分集合を変形 Tonnetz という.

以下にベートーベン作曲、ピアノソナタ第 14 番:月光の冒頭部分を Tonnetz で表した例を示す. 図 1 は、月光の冒頭部分の楽譜を示し、図 4 は、それを変形 Tonnetz で表したものある.

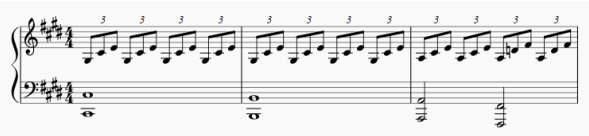


図 1: 月光の冒頭部分

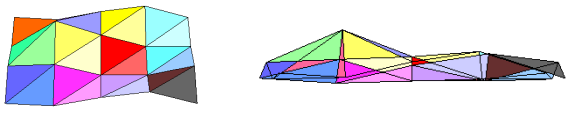


図 2: 上と横から見た月光の変形 Tonnetz

On composition of using polyhedra

[†]Tomoe Hamaoka・Graduate School of Informatics, Okayama University of Science
[‡]Masayuki Kawashima・Faculty of Informatics, Okayama University of Science

3　設定

今回はこの逆問題を考える. 即ち、任意の多面体 $S \subset \mathbb{R}^3$ から MIDI ファイルを生成することを目標とする. MIDI ファイルを生成するためには

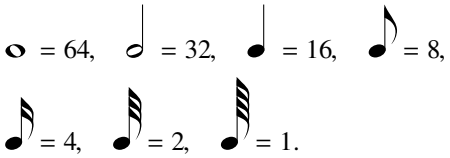
1. 「どの音」
2. 「何回」
3. 「どのくらいの長さ」鳴っているか

のデータが必要である. 1. のデータは各頂点の座標 (x, y) 座標が Tonnetz のどの音に対応するかを与えている. 今回の問題は 2 と 3 についてで、音の数と長さ、即ち、楽曲のリズムをどう表現するかである. 音の長さには様々なものがあるが、今回は以下の約束のもと考えるとする.

1. 扱う音符は, 全音符から 64 分音符まで.
2. 付点やスラーなど音の長さを調整する記号などは使用しない.
3. 休符については議論しない.

上記条件のもとでは、今回使用する音符は以下の 7 種類である.

また 4 分の 4 拍子 (4 分音符が 4 つ分で 1 小節) で考えることにすると、音の組み合わせは 64 の分割の個数に対応することがわかる. 整数値で音の長さを表すため, 64 分音符を 1, 全音符を 64 として表す:



このように考えると一小節に入る音の長さは 16×4 = 64 となる. 例えば

は 64 を (32, 16, 8, 8) と分割したものと考える.

Vol.2019-MUS-122 No.2
Vol.2019-EC-51 No.2
2019/2/22

4 リズムの構成

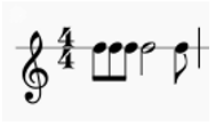
$(m_1, m_2, \ldots, m_k) \in \mathbb{N}^k$ が自然数 $n \in \mathbb{N}$ の k 分割であるとは

$$n = \sum_{i=1}^{k} m_i$$

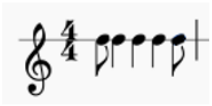
を満たすものをいう．またこのときの k を分割の長さという．上の $(32, 16, 8, 8)$ は 64 の 4 分割になっている．

即ち、64 の k 分割を指定することは 1 小節の中に音が k 個ある状態ををを指定することに対応する．例えば 64 の 64 分割は 64 分音符を 64 個並べた状態である．さらに $k = 5$ としてランダムに生成すると以下のようなリズムが作ることができる．

(8, 8, 8, 32, 8) :



(8, 16, 16, 16, 8) :



(16, 32, 8, 4, 4) :

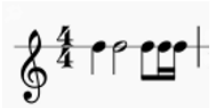


図 3: $k = 5$ として生成したリズムの例

5 自動生成アルゴリズム

5.1 準備

ここまでの準備をもとに Tonnetz を用いた楽曲の自動生成を実装する．

1. $\mathcal{N} := \{C, C^\#, D, \ldots, A, A^\#, B\}$ を音の名前の集合とする．まずは音の高さは考慮せずに名前だけで考える．

2. 楽曲を生成するためには, コード進行を指定することが必要である．コードとは音の集合のことで, 3 つの音からなるコードをトライアド･コードという．例えばメジャー･コード C とは $C = \{C, E, G\} \in 2^{\mathcal{N}}$ のことである．

3. $\mathcal{C} \subset 2^{\mathcal{N}}$ をトライアドコードの集合とする．本研究では 8 小節を 1 つのパートして考えることにする．このときコード進行とは直積集合 $\mathcal{C}^8$ の元を定めることに対応する．

4. 楽曲には A メロ、B メロ, サビ用によく使用されるコード進行が用意されており、それらの集合をそれぞれ $\mathcal{C}_A, \mathcal{C}_B, \mathcal{C}_H$ で表す．ここでそれぞれ $\mathcal{C}^8$ の部分集合であることに注意．

5. 楽曲を生成するためにはコード進行を決めることが必要になるがこれは

   $$\mathfrak{C} := \mathcal{C}_A \times \mathcal{C}_B \times \mathcal{C}_H$$

   の元 $c = \{c_A, c_B, c_H\}$ を 1 つ指定することで決定できる．

6. コード進行を決定するとはあるデータセット $\mathcal{D}$ から $\mathfrak{C}$ への写像

   $$\varphi : \mathcal{D} \to \mathfrak{C}$$

   を構成することに他ならない．

   音楽の知識がある人はこのデータセットが知識や経験によって構成されている．さらに天才と呼ばれている人たちの中にはこのデータセットそのものが用いていないとも考えられる．

   本研究ではこのデータセットを簡単に、即ち、音楽理論や経験にとらわれない物にしたい．

5.2 実際の作業

ここではデータセット $\mathcal{D}$ を指定し、実際の楽曲の自動生成と繋げる．まず音楽理論に詳しくない人からみると、前述したコード進行を指定することも一苦労である．このような背景のもと本研究ではより直感的に作曲を体験してもらうために以下の手順を取ることにする．

1. 各音の個数を入力してもらう． それを $\mathbb{R}^{12}$ の元 $\boldsymbol{x} = (x_1, \ldots, x_{12}) \in \mathbb{R}^{12}$ と捉える．このとき番号に対応して

   $$C \text{ の個数} \to x_1, \quad \ldots, \quad B \text{ の個数} \to x_{12}$$

   と $\mathcal{N}$ の元の順番に対応つける．このようにデータセット $\mathcal{D}$ として $\mathbb{R}^{12}$ を採用する．

2. 先行研究に基づき, 入力された音の個数に対して変形 Tonnetz を構成する．構成された変形 Tonnetz の形をもとに、入力者が気に入ったかそうでないか判断し、良いと判断されたとき、次のステップに進む．

3. コード進行 $c \in \mathfrak{C}$ を決めるために、各音に割り当てられた数字を加える．このとき, どこかの音に 0 が割り当てられていたらそのコードは使用しない．

Vol.2019-MUS-122 No.2
Vol.2019-EC-51 No.2
2019/2/22

4. 次に用意されているコード進行の音を全て加え、その中でどのコード進行が選ばれるかを重み付き確率を用いて決定する.

5. コード進行が決まったら、それぞれのコードのなかで、どの音どのくらい使用されるかを, 入力された $\boldsymbol{x}$ を元に先ほどと同様に重み付き確率を用いて決定する.

6 楽曲の生成

上述に従い、楽曲の生成を行う．例として生成した楽曲は、$k \leq 5$、テンポは 120 とした．Tonnetz のそれぞれの頂点の高さとして入力した値は以下の通りである．

C : 76, $D^{\flat}$: 56, D : 82 $E^{\flat}$: 71, E : 99, F : 87
$F^{\sharp}$: 56, G : 45, $A^{\flat}$: 63, A : 77, $B^{\flat}$: 68, B : 34

これを元に生成した変形 Tonnetz と楽曲を以下に示す.

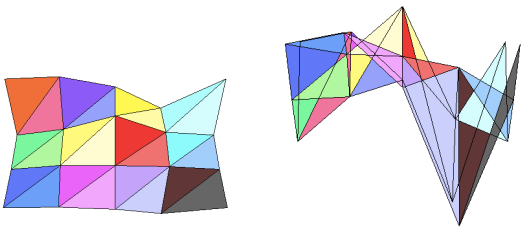


図 4: 生成した変形 Tonnetz

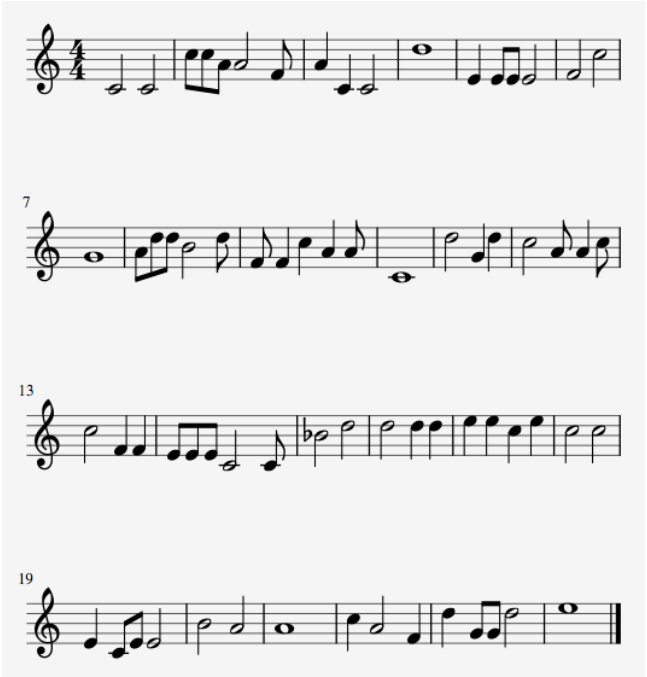


図 5: 生成した楽曲

7 まとめ

本研究では、Tonnetz を用いて楽曲の生成を行った．今回の問題点として 4 つ挙げられる．

1 つ目は、音を選ぶ際に使われているコードの音しか選ばれないことである．コードで使われている音を中心に他の音も選ばれるようになれば楽曲に広がりがみられるのではないのだろうか．2 つ目は、リズムの生成がランダムであることである．

k の値が大きくなればなるほどリズムが複雑になりすぎる．リズムに規則性を持たせればより楽曲らしくなるのではないだろうか.

3 つ目は、休符が使われていないことである．楽曲を作成するにあたって休符は楽曲の要素として重要である．リズムの生成の際に休符を取り入れることができればリズムの幅も広がるのではないだろうか.

4 つ目は時間について考慮されていないことである．楽曲において時間の流れ（音の進行）は楽曲を作成する上で重要になってくるのではないかと考えられる．しかし、それを Tonnetz 上に表現することが難しくできていないというのが現状だ．時間という要素を取り入れることができればまた違った楽曲ができるのではないだろうか.

また、このシステムをアプリケーションとして作成しようと考えている．そうすることにより、音楽理論を知らない人はもちろん、小さい子供など様々な人に気軽に作曲を体験することができる．作曲をより身近に感じてもらうことができ、作曲に対するハードルを低くすることができるのではないかと考えられる.

参考文献

[1] 自動作曲システム Orpheus, http://www.orpheus-music.org/v3/index.php

[2] Magenta, https://magenta.tensorflow.org

[3] 高木 将一 中村 啓佑 沼尾 正行, 機械学習の手法を用いた感性の抽出と作曲・編曲への応用, The 15th Annual Conference of Japanese Society for Artificial Intelligence, 2001

[4] 枡田 幹也, 講座 数学の考え方〈15〉代数的トポロジー, 朝倉書店,2002

[5] ピアノ Midi ファイル集, http://piano.s20.xrea.com/midi/