

TransDrums：ドラムのフィルインとドラムパターン遷移確率に着目した2曲間のドラムパターン対応付け手法

澤田 隼^{1,a)} 深山 覚² 後藤 真孝² 平田 圭二³

受付日 2019年7月7日, 採録日 2020年1月16日

概要：本論文では、与えられた楽曲のドラムパターンを差し替えるシステム TransDrums を提案する。TransDrums に2つの楽曲（元となる楽曲：B 楽曲，置換するドラムパターンを提供するソース楽曲：S 楽曲）が与えられると，B 楽曲のドラムパターンが対応する S 楽曲のドラムパターンで置換され，音響信号として新しい楽曲が生成される。音楽的に一貫性のある置換を実現するために，ドラムパターンの大域的構造を考慮する。まず，各楽曲から1小節ごとにフィルインなどに着目してドラムパターンを抽出し，それぞれクラスタリングを行い，ドラムパターンの遷移行列を作成する。次に，B 楽曲と S 楽曲の間で最もドラムパターンの遷移が類似するよう，ドラムパターンどうしの最適な割当てを全解探索する。そして，B 楽曲のドラムパターンを S 楽曲の割り当てられたドラムパターンで置換する。ドラムパターン c_i から c_j への遷移の傾向は条件付確率と見なすことができるので，遷移行列どうしの類似度は確率分布どうしの距離から計算できる。評価実験では，遷移確率の類似度判定に Jensen-Shannon 情報量を用いた。実験結果より，TransDrums は B 楽曲のフィルインを含む小節を，S 楽曲のそれを含む小節で，B 楽曲のフィルインを含まない小節は S 楽曲のそれを含む小節で適切に置き換えることが示された。

キーワード：音楽情報処理，ドラムパターン，楽曲構造，能動的音楽鑑賞

TransDrums: Drum Pattern Transfer System Focusing on Fills and Transition Probability of Drum Patterns

SHUN SAWADA^{1,a)} SATORU FUKAYAMA² MASATAKA GOTO² KEIJI HIRATA³

Received: July 7, 2019, Accepted: January 16, 2020

Abstract: This paper presents a drum pattern substitution system, TransDrum. When given a drum-pattern source song (S-song) and a base song (B-song), TransDrum replaces drum patterns in B-song by drum patterns in S-song which are the counterparts for the drum patterns in B-song, and renders the audio output with the substituted drum patterns of S-song. To achieve a consistent replacement in the musical sense, TransDrums focuses on the fills of drum patterns. At first, based on the measures with and without fills, TransDrums analyzes the drum-pattern sequences for the two songs and then obtains state transition matrices of drum patterns, some of which contain fills. To identify the proper counterpart relationship between drum patterns in S-song and B-song, TransDrum uses the exhaustive search. Here, since the characteristic of transition from drum pattern c_i to c_j can be represented by a conditional probability of c_j given c_i , the similarity of two state transition matrices is obtained from the distances between two probability distributions. In the experiment, TransDrum uses the sum of the Jensen-Shannon divergence for the similarity of two matrices. Our experimental results show that TransDrums can successfully replace the measures with fills in B-song by those with fills in S-song. It can also replace the ones without fills in B-song by those without fills in S-song.

Keywords: music information retrieval, drum pattern, music structure, active music listening

¹ 公立はこだて未来大学大学院
Graduate School of Future University Hakodate, Hakodate,
Hokkaido 041-8655, Japan

² 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8568, Japan

³ 公立はこだて未来大学
Future University Hakodate, Hakodate, Hokkaido 041-8655,
Japan

1. はじめに

従来の音楽鑑賞がそのまま再生して楽しむ受動的な体験であったのに対し，能動的なインタラクションをともなって音楽を鑑賞する能動的音楽鑑賞 (Active music listening) [6]

^{a)} b1012046@gmail.com

が提案され、様々な研究が取り組まれてきた。たとえば、後藤らによって提案された能動的音楽鑑賞のための web サービス, Songle [9], [19] がある。音楽理解技術で解析した内容 (サビ, メロディ, コード, ビート) を俯瞰的に眺めながら、楽曲中のサビ区間や繰り返し区間へ頭出しして聴くことができる。また、自分の好みに合うように編集しながら音楽鑑賞をするといった、能動的音楽鑑賞の技術がこれまでいくつか提案されている。Nakamura らはドラムの音色を変更するシステムを提案した [11]。これは周波数特性やドラムの音色を別の楽曲のものに置き換えることができる。Ono らはリアルタイムイコライザーを提案した [12]。これは調波・非調波成分のバランスや、周波数特性を調節し音色を変更することができる。Davies らは複数の楽曲をミックスし、新たに楽曲を生成すること (マッシュアップ) ができる AutoMashUpper [2] を提案した。リズムの類似度や音響特性に着目し自然なマッシュアップを作成することができる。

Yoshii らはリアルタイムにドラムのリズムパターンや音色を変更することができるシステム Drumix [18] を提案した。これはあらかじめ用意されているドラムパターン集合から適当なパターンを選択するだけでドラムパターンを容易に変更することができる。ドラムパターンをランダムに変更する機能は、音楽経験者・未経験者ともに思いがけない音楽体験をもたらし、好評であったという結果が得られている。本論文で提案するシステムもドラムパターンを容易に変更し能動的音楽鑑賞を可能とするシステムである。

ポピュラー音楽などの典型的なドラムトラックは、複数のドラムパターンが連結されて形成される大域的なパターン構造を有している。たとえば、A メロ、B メロ、サビなどの単位のセクションや、2 小節、4 小節、8 小節などの比較的短いセクションなど、楽曲の構造の変わり目に、今まで繰り返していたドラムパターンとは異なるドラムパターン (フィルイン) がしばしば挿入される。しかしながら、ドラムトラックを小節ごとに置き換える従来のシステムでは、このような大域的な構造を形成することができなかつた。

これをふまえて本論文では、ポピュラー音楽を対象としたドラムパートの大域的な構造を保持しつつドラムパターンを差し替えるシステムを提案する。本システムの概要を図 1 に示す。本システムはドラムを差し替えたい 2 つの楽曲 (ベース楽曲: B 楽曲, ドラムパターンソース楽曲: S 楽曲) を入力し、B 楽曲のドラムパターンを S 楽曲のドラムパターンに差し替えるシステムである [14], [20]。その際、B 楽曲のドラムパターンの遷移に関する大域的な構造を保持しながら、B 楽曲のドラムパターンをそれと対応付けられた S 楽曲のドラムパターンに差し替える。

本システムを実現するためには以下の 2 つの問題を解決する必要がある。

- (a) 楽曲中のドラムパターンの大域的構造の認識。

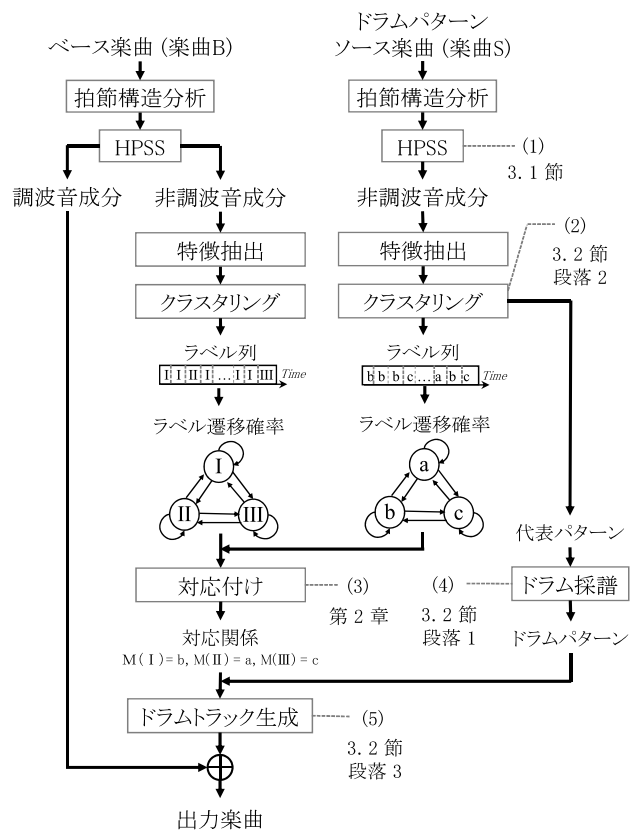


図 1 システム概要

Fig. 1 Overview of our method.

- (b) 楽曲中での役割が類似したドラムパターンどうしの対応付け。

適切なドラムパターンのペアを対応付けるために、我々は楽曲中のドラムパターン間の遷移確率を調べることで、ドラムパターンの大域的構造を抽出する。ドラムパターンの大域的構造とは、いい換えるとドラムパターンの長期依存性 (long-term dependency) のことである。たとえば 4 小節や 8 小節、16 小節などの単位でドラムパターンに繰り返しが生じることや、A メロや B メロなどのセクションによって異なる基本ドラムパターンが叩かれているなどである。また、4 小節や 8 小節、16 小節離れた箇所フレーズや A メロなどのセクションの最後の 1 小節に、同じようなフィルインを含むドラムパターンが現れるような構造を指す。それぞれの楽曲のドラムパターンのクラスタリングを行い、楽曲中のドラムパターン間の遷移確率を求め、それぞれの楽曲におけるドラムパターンの遷移確率分布間の距離が最小となるようにドラムパターンどうしのペアを決定する。これによりドラムパターンの楽曲中の大域的構造を保持し、音楽の一貫性を保ちながら、雰囲気異なる楽曲の生成が可能になる。ドラムパターンの大域的構造のモデル化は、定量的な楽曲構造分析に貢献する可能性がある。また、本システムの実現は、能動的音楽鑑賞に資するだけでなく、作曲・編曲支援技術への応用も考えられる。たとえば、モデル化された大域的なドラムパターンの構造を

利用することで、ドラム未経験者でも既存の楽曲を参考にしながら多種多様なドラムパターンを付加することが可能になるドラムパート作成支援ツールなどである。本システムのデモンストレーションは次の Web サイトにて公開されている (<https://shunsawada.github.io/transDrum/>)。

2. ドラムパターンの対応付けアルゴリズム

本システムは、B 楽曲のドラムパターンの遷移に関する大域的な構造を保持しながら、B 楽曲のドラムパターンをそれと対応付けられた S 楽曲のドラムパターンに差し替える。適切な対応関係を見つける問題は、B 楽曲に出現するドラムパターンの集合 B から S 楽曲に出現するドラムパターンの集合 S への対応関係 $M: B \rightarrow S$ を求める問題といえる (図 1 の (3))。ドラムパターンの大域的な構造は、B 楽曲と S 楽曲に出現する典型的なドラムパターン間の遷移確率を分析することで、確率モデルとしてモデル化する。

まず、入力された B 楽曲と S 楽曲の 2 つの楽曲のスペクトログラムを調波成分と非調波成分に分離し、それぞれの楽曲のスペクトログラムの非調波成分から複数の代表的な 1 小節単位のドラムパターンを抽出する。次に、B 楽曲と S 楽曲のそれぞれの代表的なドラムパターン間のそれぞれの楽曲中での遷移確率を確率モデルとして分析することによって、ドラムパターンの大域的な構造をモデル化する。そのモデルに従って B 楽曲と S 楽曲の間のドラムパターンのペアを作成し、対応するドラムパターンを使用して差し替え、最後に音響信号を合成する。

図 2 に B 楽曲と S 楽曲のドラムパターンの対応付けの処理を示す。まず、小節ごとのドラムパターンの特徴量はクラスタリングによってラベル付けされる。クラスタの数を N_{cluster} とするとクラスタの集合は $C = \{c_n\}_{n=1}^{N_{\text{cluster}}}$ となる。本システムにより、入力された楽曲はクラスタ集合 $C = \{c_n\}_{n=1}^{N_{\text{cluster}}}$ を用いて小節単位のラベル列 $l = \{l_n\}_{n=1}^{N_{\text{bar}}}$ に変換される。ここで各ラベル l_n はクラスタ集合 C の要素のいずれかに対応するラベルであり、 N_{bar} は楽曲中の小節の数を表す。次に、ラベル列 $l = \{l_n\}_{n=1}^{N_{\text{bar}}}$ の l_{n-1} から l_n へ遷移するラベルの遷移確率を計算する。楽曲 X (B 楽曲あるいは S 楽曲) の遷移確率行列 T_X ($N_{\text{cluster}} \times N_{\text{cluster}}$ 行列) の要素 t_{ij} は、楽曲中のドラムパターンのラベル c_i からラベル c_j への遷移確率 $P(c_j | c_i)$ を表している。行列 T_X の n 番目の行は c_n からの遷移についての離散確率分布となっている。確率 P_n は B 楽曲の遷移確率行列 T_B の n 番目の行を表し、確率 Q_m は S 楽曲の遷移確率行列 T_S の m 番目の行を表す。

次に、大域的な構造を保持したドラムパターンのペアを対応付けるために、我々は楽曲内でのドラムパターンの役割に関する距離の尺度を設定した。本研究でのドラムパターンの大域的な構造は遷移確率として表現されているため、Kullback-Leibler 情報量に基づいて計算される尺度

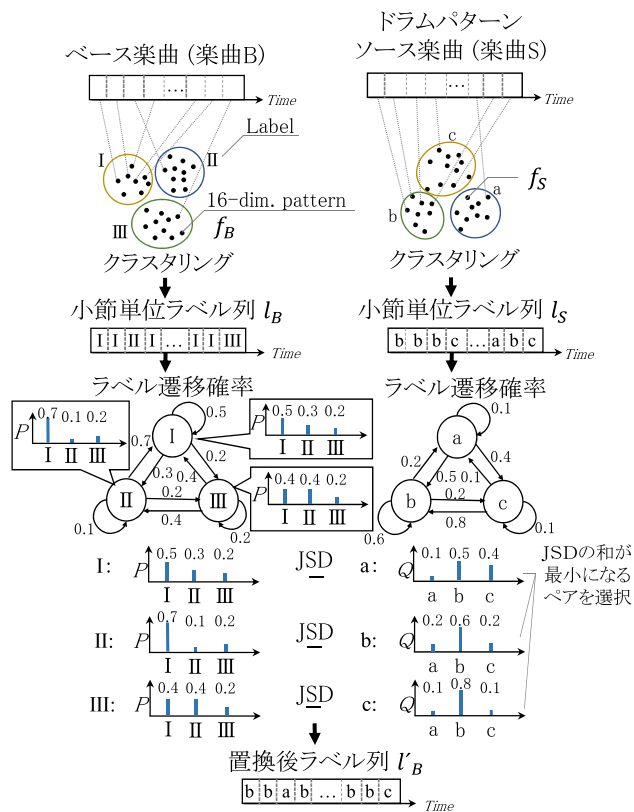


図 2 ドラムパターンの対応付け

Fig. 2 Drum patterns mapping algorithm of TransDrums.

を用いることができる。楽曲内で同じ役割を持つドラムパターンは類似した遷移確率を有すると考えられるため、ドラムパターンの遷移確率分布間の情報量の和を最小化することによって楽曲内で同じ役割を持つドラムパターンの対応付けが獲得できる。確率分布 Q_m から P_n への Kullback-Leibler 情報量 $KL(P_n || Q_m)$ は以下のように定義される。

$$KL(P_n || Q_m) = \sum_{i=1}^{N_{\text{cluster}}} P_n(c_i | c_n) \log \frac{P_n(c_i | c_n)}{Q_m(c_i | c_m)}. \quad (1)$$

本論文では、Kullback-Leibler 情報量に対称性を持たせた Jensen-Shannon 情報量 $JS(P_n || Q_m)$ を用いた。Jensen-Shannon 情報量は以下のように定義される。

$$JS(P_n || Q_m) = \frac{1}{2}(KL(P_n || M) + KL(Q_m || M)), \quad (2)$$

ただし、

$$M = \frac{1}{2}(P_n + Q_m). \quad (3)$$

本システムは、B 楽曲と S 楽曲の遷移確率におけるすべての可能なペアどうしの Jensen-Shannon 情報量を計算し、すべての対応関係の Jensen-Shannon 情報量の和を最小化するような対応関係を全探索する。この方法によって得られたドラムパターンの対応関係 M を用いて、B 楽

曲のラベル列 l_B は S 楽曲のラベルに置き換えられ、新しいラベル列 l'_B が生成される。B 楽曲と S 楽曲におけるドラムパターンは 1 対 1 対応をなすように対応付けるため、各楽曲 N_{cluster} 個のドラムパターンすべてを対応付ける組合せは $N_{\text{cluster}}!$ 通りある。いま各組合せにインデックス j ($1 \leq j \leq N_{\text{cluster}}!$) を割り振り、また、それぞれの組合せにおけるドラムパターンの対応付けにインデックス i ($1 \leq i \leq N_{\text{cluster}}$) を割り振る。さらに j 番目の組合せの i 番目の対応付けにおける B 楽曲のドラムパターンの遷移確率分布を P_i^j 、S 楽曲のドラムパターンの遷移確率分布を Q_i^j とする。このとき、 R_j を以下の式で定義し、

$$R_j = \sum_{i=1}^{N_{\text{cluster}}} \text{JS}(P_i^j || Q_i^j) \quad (4)$$

この R_j が最小となる j 番目のドラムパターンの組合せを最適なペアとする。

3. TransDrum の実装

本システムでは楽曲中にドラムパートが存在するポピュラー音楽を対象としており、テンポや拍子が途中で変化する楽曲は対象外である。ここでのドラムパートとはハイハットシンバル、スネアドラム、バスドラムによって演奏されるものを想定している。

3.1 ドラムパターン特徴量の抽出

入力された楽曲の 1 小節ごとのドラムパターンの特徴量の抽出方法について述べる。入力された楽曲は調波音成分と非調波音（打楽器音など）からなる。ここでいう調波音とは、整数次の倍音成分で構成される調波構造を持つ音で、ギターやピアノなどの明確な音高を有する楽器音に含まれる持続音に対応する。非調波音とは、そうした調波構造を持たない音で、それらの楽器の鳴り始め直後に含まれるアタック音や、本研究で対象とする打楽器音に対応する。スペクトログラム上で、調波音成分は時間軸方向、非調波音成分は周波数軸方向に連続する成分となることが多い。本論文では非調波音としてドラムセットに含まれるハイハットシンバル、スネアドラム、バスドラムなどの楽器を想定している。ドラムパターンは多くの非調波音成分を含む。まず、入力された楽曲のスペクトログラムを調波音成分と非調波音成分に分離し（図 1 の (1)）、ドラムパターン特徴量の抽出ではこの非調波音成分を扱う。ドラムパターンの特徴量として、オンセットの顕著性 (saliency) を用いた。オンセットの顕著性は、分離された非調波音成分のパワースペクトルに関して、着目するフレームと前のフレームとの差分（増加量）から求められる。16 分音符の間隔と一致するように 16 次元へと次元圧縮したものをドラムパターンの特徴量とする。本システムでは小節ごとに処理を行うため、事前に B 楽曲と S 楽曲のそれぞれに対して拍節構造分

析 (Beat tracking/Downbeat estimation) を行っておく。拍節構造分析は Python のライブラリである `madmom` を使用した [1]。

まず、入力された楽曲を調波音成分と非調波音成分に分離する。入力された 2 つの楽曲 (B 楽曲と S 楽曲) の音響信号のそれぞれに対して、窓幅 2,048 サンプルの分析窓を 512 サンプルずつシフトしながら短時間フーリエ変換 (Short-time Fourier transform: STFT) を行い、スペクトログラムに変換する。次に、得られたそれぞれのスペクトログラムに対して調波・非調波音分離 (Harmonic/Percussive sound separation: HPSS) を用い、入力スペクトログラム S を、調波音成分 H と非調波音成分 P に分解する。このとき、HPSS は、スペクトログラム上では調波音成分は時間軸方向に滑らかであり、打楽器などの非調波音成分は周波数軸方向に滑らかであるという特性を利用する。

HPSS は Python のライブラリである LibROSA (Version 0.6.2) [10] に実装されている `librosa.decompose.hpss` 関数を使用した。TransDrums で用いた HPSS [3] は、Fitzgerald [4] の手法が拡張されたものである。調波音成分と非調波音成分に分離するマスクの閾値に関するパラメタ (margins) が導入されており、本システムでは margins を 2.0 に設定した。

次に、分離されたそれぞれの楽曲の非調波音成分 P のスペクトルフラックス (Spectral flux) を求める。スペクトルフラックスはスペクトルの時間的変化を表す尺度として用いられる。時刻 t と時刻 $t-1$ における周波数 bin k における差分 $D_{k,t}$ を全周波数 bin n にわたって足し合わせて求める。ここで時間 t におけるスペクトルフラックスを \mathbf{SF}_t と表す。スペクトルフラックス \mathbf{SF}_t は次の式で与えられる。

$$\mathbf{SF}_t = \sum_{k=1}^n D_{k,t}, \quad (5)$$

$$D_{k,t} = \begin{cases} S_{k,t} - S_{k,t-1} & (S_{k,t} - S_{k,t-1} > 0) \\ 0 & (\text{otherwise}), \end{cases} \quad (6)$$

ここで $S_{k,t}$ はスペクトログラムにおける時刻 t の周波数 bin k における値を表している。今回はスペクトルの立ち上がりに着目しているため、増加分のみを計算している。

次に、 \mathbf{SF}_t を、16 分音符のビート位置の周り m フレームにわたって足し合わせて 16 分音符単位へと離散化する。これによって得られるオンセットの顕著性を表す 16 次元ベクトル $\mathbf{f} = \{f_i\}_{i=1}^{16}$ をドラムパターンの特徴量として用いる。フレーム数 m は 1~16 までをそれぞれ試し、ドラムパターンのクラスタリングの精度を比較する予備実験を行い、一番クラスタリングの精度が高かった $m = 8$ を採用した。このとき、RWC 研究用音楽データベース (RWC-MDB-P-2001) [7] の楽曲を使用した。ドラムパターン特徴量 f_i は次の式で与えられる。

$$f_i = \sum_{\tau=-m}^m \mathbf{SF}_{u_i+\tau}, \quad (7)$$

ここで、 $\mathbf{u} = \{u_i\}_{i=1}^{16}$ は 16 分音符単位のビート時刻列を表す。たとえば、10 フレームごとに均等に 16 分音符単位のビート位置がある場合、 $\mathbf{u} = \{10, 20, 30, \dots, 160\}$ となる。

3.2 ドラム採譜とドラムトラックの生成

2 章によって得られた新しいラベル列 \mathbf{l}_B に沿って、S 楽曲のドラムパターンのオーディオを生成し、B 楽曲の調波音成分と合成する。まず、S 楽曲から代表的なドラムパターンを抽出する。代表的なドラムパターンは、クラスタリングによって形成された S 楽曲のクラスタのセントロイドに最も近いドラムパターンとした。次に、代表パターン \mathbf{P}_S のドラム採譜を行う (図 1 の (4))。最後に、このドラム採譜の結果を用いてドラムトラックを生成する。HPSS によって得られる非調波音成分を使ってドラムトラックを合成した場合、歌声の子音が多く含まれていた。ノイズの含まれていないドラムパターンを生成するために、ドラムを採譜した結果からドラムパターンのオーディオを再生成している。また、ドラム採譜を行うことにより、入力される 2 つの楽曲のテンポの違いを音響信号のレベルで対処する必要がなくなる。

クラスタリングは Python のライブラリである `scikit-learn` (Version 0.19.1) [13] に実装されている `k-means` クラスタリングを使用した (図 1 の (2))。本論文の実装ではクラスタ数を自動で推定していないが、適切なクラスタ数を決めることは重要な課題である。クラスタ数を大きくすると、より細かいドラムパターンの違いを扱うことができる一方で、ドラムパターンの遷移確率が疎になるため、大きすぎると適切な対応付けは行えない。また、クラスタ数を小さくすると基本パターンとそれと類似したフィルインを含むパターンが同じクラスタに所属してしまうため適切な対応付けが行えない。

ドラム採譜として、Recurrent neural network (RNN) に基づいた手法を使用した。RNN に基づいた手法は、ポリフォニック楽曲のドラム採譜において多く使用されており [15], [16], [17], 高精度なドラム採譜が可能である。本システムでは自動ドラム採譜のライブラリ (ADTLib) [16] を使用した。このシステムはバスドラム、スネアドラム、ハイハットシンバルのオンセット時刻を出力する。これらのオンセット時刻を用いてドラムトラックを生成する。ドラムトラックは各オンセット時刻において、ドラム音源を発音することで生成する。ドラム音源として RWC 研究用音楽データベース：楽器音 (RWC-MDB-I-2001) を使用した [8]。使用した音源名は次のとおりである。スネアドラム：431SD7N3, バスドラム：432BD2N3, ハイハット：432HHCC3。最後に、こうして得られたドラムトラックと B 楽曲の調波音成

分が足し合わされ、出力される (図 1 の (5))。

4. ドラムパターン差し替えの評価実験

4.1 実験条件

大域的なドラムパターンを認識し、ドラムパターンが適切な対応付けがされているかを調べるため、以下の 2 つの実験を行う。

実験 1 B 楽曲, S 楽曲において、フィルインを含むパターンどうしが対応付けされているかを評価する実験。

実験 2 B 楽曲, S 楽曲において、頻出する遷移パターンどうしが対応付けされているかを評価する実験。

本システムを評価するために、RWC 研究用音楽データベース (RWC-MDB-P-2001) [7] の楽曲 (No.1 から No.25 の 25 曲) を使用した。正解データとしてドラムパターンのラベル列とフィルインを含むパターンであるか否かのラベルを用いて評価を行った。正解データは、ドラム演奏経験が 3 年以上ある 2 名が作成してクロスチェックした。実験 1 と 2 は、25 曲のうちクラスタ数が同じ楽曲のペア (122 通り) のみを用いて行った。使用した楽曲のクラスタ数の分布を図 3 に示す。クラスタ数は 6 が最も多くその割合は 0.44 であり、次いでクラスタ数 10 が多くその割合は 0.24 であった。また、今回の実験ではビートトラッキングやダウンビート検出の精度によって評価が左右されないように、AIST Annotation for the RWC Music Database [5] の拍節構造のラベルを用いた。ただし、16 分音符単位のグリッドは、各拍位置の間隔を均等に 4 つに分割することによって獲得した。

まずはじめに、実験 1, 2 に使用する評価尺度を定義するにあたって使用する記号の説明を以下に述べる。B 楽曲, S 楽曲のラベル集合をそれぞれ $\mathbf{B} = \{b_n\}_{n=1}^{N_{\text{cluster}}}$, $\mathbf{S} = \{s_n\}_{n=1}^{N_{\text{cluster}}}$ と表す。本手法によって得られる B 楽曲

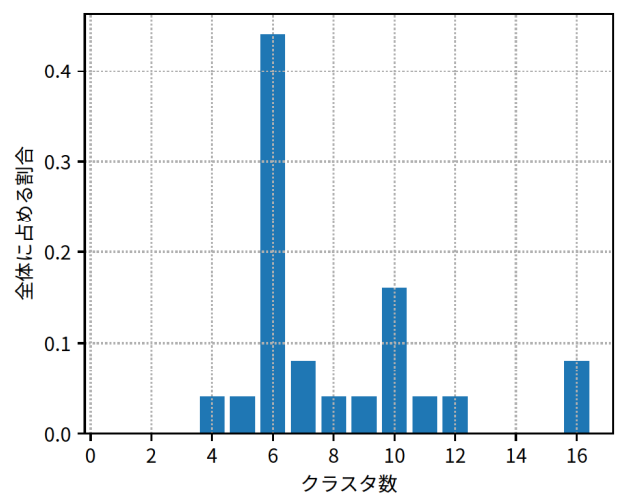


図 3 実験に使用した楽曲のクラスタ数の分布

Fig. 3 Distribution of the number of clusters of songs used in the experiment.

と S 楽曲の対応関係を $M(b_i) = s_j$ と表す。

実験 1 において、2 つの楽曲のフィルインを含むパターンどうしが対応付けられているかを調べる尺度としてフィルイン一致率 $F(M)$ を定義する。楽曲 B に対する楽曲 S の $F(M)$ は以下のように定義される。

$$F(M) = \frac{\sum_{i=1}^{N_{\text{cluster}}} f(c_i)f(M(c_i))}{N_{\text{fill-in}}}, \quad (8)$$

ここで、 $N_{\text{fill-in}}$ は楽曲 B 中で出現するフィルインを含むパターンの種類の数を表す。あるドラムパターンを p としたとき、 $f(p)$ は p にフィルインが含まれるかどうかを判定する関数であり、その値域は $\{1, 0\}$ である。たとえば、 $a \in B$ がフィルインを含むパターンであれば、 $f(a) = 1$ となり、そうでなければ $f(a) = 0$ となる。 $F(M)$ は入力された 2 つの楽曲のすべてのフィルインを含むパターンどうしが対応付いたときに最大値 1 をとる ($0 \leq F(M) \leq 1$)。

実験 2 において、楽曲中のドラムパターンラベルのバイグラム頻度に関して、楽曲どうしの整合性を表す尺度として、遷移整合性 $E(M)$ を定義する。たとえば、B 楽曲と S 楽曲のドラムパターンのラベル列のバイグラム頻度の上位がそれぞれ “12” と “ab” だった場合、対応関係は $M(1) = a$, $M(2) = b$ となることが期待される。ただし、パターンの対応付けの際、ドラムパターンの割当 M は全単射なので、必ずしも整合性がとれない可能性があるため、遷移整合性 $E(M)$ の尺度で評価する。遷移整合性 $E(M)$ を以下のように定義する。

$$E(M) = \sum_{r=1}^{10} \frac{1}{2} w(r)(R_0(r) + R_1(r)), \quad (9)$$

ここで $R_i(r)$ と $w(r)$ は以下の式で得られる。

$$R_i(r) = \begin{cases} 1 & (M(x_{ri}^b) = x_{ri}^s) \\ 0 & (\text{otherwise}), \end{cases} \quad (10)$$

$$w(r) = \frac{v_{x_{r0}x_{r1}}}{\sum_{r=1}^{10} v_{x_{r0}x_{r1}}}, \quad (11)$$

ここで、 $v_{c_i c_j}$ はドラムパターン c_i から c_j へ遷移する頻度を表し、 $\{x_{ri}\}_{i=0}^1 = x_{r0}x_{r1}$ は楽曲中のバイグラム遷移パターンを頻度順にソートしたものの r 番目に頻出するパターンを指す。たとえば、3 番目に頻出するバイグラム遷移パターンが ab だった場合、 $x_{30} = a$, $x_{31} = b$ となる。 $R_i(r)$ は B 楽曲の x_{ri}^b と S 楽曲の x_{ri}^s が対応付いているときに 1 の値をとる。 $E(M)$ は提案手法によって得られた対応関係が楽曲中のラベルのバイグラムの頻度と整合性がとれているかどうかを表す。 $E(M)$ は $0 \leq E(M) \leq 1$ の値をとり、対応関係が頻度と一致していたときに最大値 1 をとる。

4.2 実験結果

すべての楽曲のとりうるペア (122 通り) の $E(M)$ の平

均とは 0.37、標準偏差は 0.28 であり、 $F(M)$ の平均は 0.73、標準偏差は 0.23 であった。

表 1 に入力する楽曲のペアごとの遷移整合性 $E(M)$ の上位 5 つと下位 5 つの結果と、そのときのフィルイン一致率 $F(M)$ を示す。たとえば、表 1 (左) の 1 行目は RWC-MDB-P-2001 No.17 (B 楽曲) のドラムパターンを RWC-MDB-P-2001 No.15 (S 楽曲) のドラムパターンで差し替えた場合の結果を示しており、 $E(M)$ が 0.96 であることを示し、 $F(M)$ は 1.0 である。

表 2 にベース楽曲 (B 楽曲) を固定して、S 楽曲を変更した場合の結果を示す。RWC-MDB-P-2001 No.15 のドラムパターン構造を図 4 に示す。また、表 2 の中で最も $E(M)$ が低かった (0.39) RWC-MDB-P-2001 No.13 のドラムパターン構造を図 6 に、最も $E(M)$ が高かった (0.86) RWC-MDB-P-2001 No.17 のドラムパターン構造を図 5 に示す。横軸は小節番号を表し、縦軸はドラムパターンのラベルを表す。フィルインを含む小節は橙色 (□) で印付けし、フィルインを含まない小節は青色 (○) で印付けした。また、本システムのドラムパターン対応付けアルゴリズムによって得られた対応関係の例を表 3 に示す。RWC-MDB-P-2001 No.15 のドラムパターンのラベル 0, 1, は基本パターンであり、2, 3, 4, 5 がフィルインを含むパターンである。RWC-MDB-P-2001 No.17 も同様に 0,

表 1 実験結果：遷移整合性 $E(M)$ の値の上位 5 ペア (左) と下位 5 ペア (右) とそのときのフィルイン一致率 $F(M)$

Table 1 Experimental result: Top 5 results (left) and worst 5 results (right) with regard to transition consistency $E(M)$, and the corresponding fill-in pattern mapping rate $F(M)$.

楽曲ペア	$E(M)$	$F(M)$	楽曲ペア	$E(M)$	$F(M)$
No.17 - 15	.96	1.0	No.17 - 14	.01	.75
No.17 - 12	.90	1.0	No.8 - 18	.02	1.0
No.17 - 5	.89	.50	No.18 - 8	.02	.40
No.17 - 7	.89	.75	No.7 - 18	.02	1.0
No.15 - 17	.86	1.0	No.18 - 14	.03	.60

表 2 実験結果：遷移整合性 $E(M)$ とフィルイン一致率 $F(M)$

Table 2 Experimental results: Transition consistency $E(M)$, and the corresponding fill-in pattern mapping rate $F(M)$.

楽曲ペア	$E(M)$	$F(M)$
No.15 - 5	.79	.50
No.15 - 7	.79	.50
No.15 - 8	.40	.50
No.15 - 12	.78	.75
No.15 - 13	.39	.75
No.15 - 14	.75	.75
No.15 - 17	.86	1.0
No.15 - 18	.79	1.0

表 3 システムが出力した対応関係の例

Table 3 Example of mapping outputted by our system.

楽曲ペア	対応関係					
	$M(0)$	$M(1)$	$M(2)$	$M(3)$	$M(4)$	$M(5)$
No.15 - 13	1	2	4	0	3	5
No.15 - 17	0	1	5	3	2	4

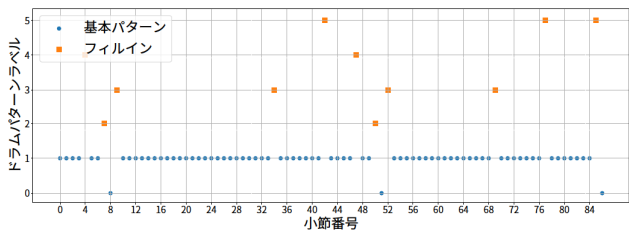


図 4 RWC-MDB-P-2001 No.15 のドラムパターン構造

Fig. 4 Drum pattern structure of RWC-MDB-P-2001 No.15.

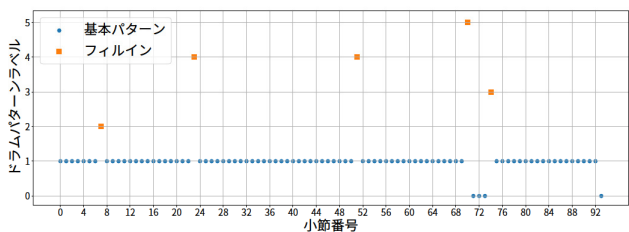


図 5 RWC-MDB-P-2001 No.17 のドラムパターン構造 (No.15 とドラムパターンの構造が類似している)

Fig. 5 Drum pattern structure of RWC-MDB-P-2001 No.17 (The drum pattern structure is similar to No.15).

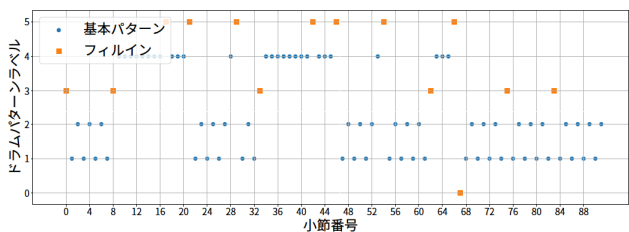


図 6 RWC-MDB-P-2001 No.13 のドラムパターン構造 (No.15 とドラムパターンの構造が異なる)

Fig. 6 Drum pattern structure of RWC-MDB-P-2001 No.13 (The drum pattern structure is different from No.15).

1 は基本パターンであり, 2, 3, 4, 5 がフィルインを含むパターンであるため, 基本パターンどうし, フィルインを含むパターンどうしが対応付けられていることが確認できる. 一方, RWC-MDB-P-2001 No.13 はドラムパターンのラベル 1, 2, 4 が基本パターンであり, 0, 3, 5 がフィルインを含むパターンである. RWC-MDB-P-2001 No.15 と No.13 のフィルインを含むパターンの数が異なるが, No.13 のフィルインを含むパターンはすべて No.15 のフィルインを含むパターンと対応付けられていることが確認できる.

5. 考察

全 122 ペアに対して, フィルインを含むドラムパターン

のラベルの割合 (フィルインが含まれるかどうかを判定する関数 $f(p)$ の平均) は 0.56 であることから, フィルイン一致率のチャンスレベルは 0.31 ($\approx 0.56 \times 0.56$) となる. また, ベースラインとして以下の 2 つの手法の $E(M)$ と $F(M)$ を示す.

BM1 B 楽曲と S 楽曲のドラムパターンのユニグラム確率の順序が一致するように対応付けを行う方法.

BM2 $E(M)$ を最大化するように対応付けを行う方法.

BM1 について, すべての楽曲のとりうるペア (122 通り) の $E(M)$ の平均は 0.56, $F(M)$ の平均は 0.65 であった. BM2 について, $E(M)$ の平均は 0.67, $F(M)$ の平均は 0.64 であった. 提案手法の $F(M)$ の平均 0.73 は, B 楽曲のフィルインを含むパターンに S 楽曲のフィルインを含むパターンがある程度正しく割り当てられていることを示している. 一方, 提案手法による遷移整合性 $E(M)$ の平均が 0.37 と低い理由を考察する.

まず, $F(M)$, $E(M)$ が高い例として, 表 1 中, No.15 (B 楽曲) と No.17 (S 楽曲) のペアを取り上げる (図 4, 図 5). 表 3 の ($M(1) = 1$) は, 図より, 楽曲中で頻繁に繰り返されているパターンどうしの対応付けであることが分かる. No.15 では, 基本パターン 0 からフィルインを含むパターン 3 への遷移が確認でき, No.17 でも, 同様の遷移が確認できる. つまり, $M(0) = 0$, $M(3) = 3$ である (表 3).

次に, $F(M)$ が低い例として, 表 1 中, No.17 (B 楽曲) と No.5 (S 楽曲) のペアを調べる. $F(M)$ が低い原因として, フィルインを含むパターンの数 $N_{\text{fill-in}}$ の差が考えられる. 実際, No.17 の $N_{\text{fill-in}}$ は 4 であるのに対し No.5 のそれは 2 であるが, 割当て M は全単射という制約のため, B 楽曲のフィルインを含むパターンの全部を S 楽曲のフィルインを含むパターンに割り当てることができないからである. $F(M)$ の値が低い他の楽曲のペアでも, $N_{\text{fill-in}}$ の差が大きい傾向にあることが確認された.

さらに $E(M)$ が低い例として, 表 2 中, No.15 (B 楽曲) と No.13 (S 楽曲) のペアを調べる. $E(M)$ が低い原因として, 基本パターンの構造の差が考えられる. たとえば, No.15 では基本パターンが 1 つのリズムパターンの繰り返しであるのに対し (図 4), No.13 では基本パターンとして 2 つのリズムパターンを交互に数回繰り返している (図 6). ここでも, 割当て M は全単射という制約のために, No.15 の基本パターンを適切に No.13 の基本パターンに割り当てられない. 以上より, フィルイン一致率, 遷移整合性を上げるには, 楽曲構造を反映した全単射ではない割当て M を考慮する必要がある.

ベースラインと提案手法との比較について述べる. BM1 について, ユニグラム確率の場合, 楽曲中でのパターンの出現頻度が類似しているものどうしの対応付けになり, フィルインを含むパターンの数やパターンの出現頻度などといった楽曲の構造が類似している楽曲どうしの場合は

$F(M)$ が高くなる．一方で，基本パターンとフィルインを含むパターンの出現頻度に差がない場合には適切な対応付けができない．BM2について， $E(M)$ を最大化する対応付けの場合，楽曲中でのパターンのバイグラムの出現頻度が類似しているものどうしの対応付けになる． $E(M)$ の特性上，出現する頻度が多いパターンどうしが優先して対応付け，自己遷移の多い基本パターンどうしなどが優先的に対応付けられることになる．一方で，フィルインを含むパターンなどの出現頻度が低いものどうしの対応付けはあまり重要視されない．提案手法の場合，特性上フィルインを含むパターンどうしが優先して対応付けやすく，フィルインを含むパターンであることや基本パターンであるといった，ドラムパターンの役割が適切に保持される傾向にある．

本提案手法でのドラムパターンのクラスタリングでは，3.1節によって得られたドラムパターン特徴量(式(7))しか考慮していないため，上のNo.13のような構造を持つドラムパターンに対して，フィルイン判定率やフィルイン一致率が上がらない可能性がある．たとえば00031115という正解ラベル列のドラムパターンを考える．ここで，ラベル0, 1が基本パターンのラベルを表し，ラベル3, 5はフィルインを含むパターンのラベルを表すとす．基本パターンであるラベル0にフィルインを加えたパターンがラベル3であり，ラベル1とラベル5も同様の関係にある．このとき，式(7)より得られるドラムパターン特徴量で考えると，ラベル0はラベル3と近く，同じ基本パターンであるラベル1とは遠いと判定されることが多いだろう．ラベル1とラベル5についても同様である．その結果，式(7)のドラムパターン特徴量だけでは00001111とクラスタリングされてしまう可能性がある．よって，ドラムパターンの構造を考慮するようなクラスタリングが望まれる．

6. おわりに

本論文では，フィルインに注目してドラムパターンの大域的な構造を分析し，元になる楽曲(B楽曲)のドラムパターンを他の楽曲(S楽曲)のドラムパターンで置換するシステムTransDrumsを提案した．ジャンルやスタイルが近い曲どうしの場合，実際に置換されたB楽曲を聴取すると，そのようなドラムパターンも可能であろうと感じられる遷移箇所を認知することができた．

より適切なドラムパターンどうしの照合を評価するため，N-gramの考え方を応用して，ユニグラムの評価尺度であるフィルイン一致率 $F(M)$ とバイグラムの評価尺度である遷移整合性 $E(M)$ を提案した．しかし多くの楽曲では，ドラムパターンに階層や繰返しなどの構造があり，N-gramの考え方だけでは不十分であることが示唆された．本手法は役割の類似するパターンを自動で対応付け入れ替えることはできるが，聴取した際の印象としての類似性は未検討である．また，ドラムパターンの大域的な

構造を知る手がかりとして，フィルイン以外にも定型リズムパターン，装飾フレーズ(キメやオカズなどと呼ばれる特徴的なフレーズ)，Aメロ，Bメロ，サビといった楽曲構造などがある．そのため，高速に連打される打楽器音の識別や分析手法を確立する必要があるだろう．

今後の課題として，ドラムパターンのクラスタリング結果とドラムパターンどうしの対応付け結果を互いにフィードバックさせる手法の検討がある．つまり，妥当なパターンどうしの対応付けが得られるようにクラスタリングを行い，妥当なクラスタリングが得られるようにパターンどうしの対応付けを行うことで，置換されたB楽曲の品質の向上を目指す．

謝辞 本研究の一部はJST ACCEL(JPMJAC1602)，JSPS科研費(19J15634)の助成を受けたものです．

参考文献

- [1] Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F. and Widmer, G.: madmom: A New Python Audio and Music Signal Processing Library, *Proc. 24th ACM International Conference on Multimedia*, pp.1174–1178 (2016).
- [2] Davies, M.E.P., Hamel, P., Yoshii, K. and Goto, M.: AutoMashUpper: Automatic Creation of Multi-song Music Mashups, *IEEE/ACM Trans. Audio, Speech, and Language Processing*, Vol.22, No.12, pp.1726–1737 (2014).
- [3] Driedger, J., Müller, M. and Disch, S.: Extending Harmonic-Percussive Separation of Audio Signals, *Proc. 15th International Society for Music Information Retrieval Conference (ISMIR)*, pp.611–616 (2014).
- [4] Fitzgerald, D.: Harmonic/Percussive Separation Using Median Filtering, *Proc. 13th International Conference on Digital Audio Effects (DAFx-10)* (2010).
- [5] Goto, M.: AIST Annotation for the RWC Music Database, *Proc. 7th International Conference on Music Information Retrieval (ISMIR)*, pp.359–360 (2006).
- [6] Goto, M.: Active Music Listening Interfaces Based on Signal Processing, *Proc. 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp.1441–1444 (2007).
- [7] Goto, M., Hashiguchi, H., Nishimura, T. and Oka, R.: RWC Music Database: Popular, Classical, and Jazz Music Databases, *Proc. 3rd International Conference on Music Information Retrieval (ISMIR)*, pp.287–288 (2002).
- [8] Goto, M., Hashiguchi, H., Nishimura, T. and Oka, R.: RWC Music Database: Music Genre Database and Musical Instrument Sound Database, *Proc. 4th International Conference on Music Information Retrieval (ISMIR)*, pp.229–230 (2003).
- [9] Goto, M., Yoshii, K., Fujihara, H., Mauch, M. and Nakano, T.: Songle: A Web Service for Active Music Listening Improved by User Contributions, *Proc. 12th International Society for Music Information Retrieval Conference (ISMIR)*, pp.311–316 (2011).
- [10] McFee, B., Raffel, C., Liang, D., Ellis, D.P.W., McVicar, M., Battenberg, E. and Nieto, O.: librosa: Audio and Music Signal Analysis in Python, *Proc. 14th Python in Science Conference (SciPy)*, pp.18–25 (2015).
- [11] Nakamura, T., Kameoka, H., Yoshii, K. and Goto, M.:

Timbre Replacement of Harmonic and Drum Components for Music Audio Signals, *Proc. 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.7470-7474 (2014).

- [12] Ono, N., Miyamoto, K., Kameoka, H. and Sagayama, S.: A Real-time Equalizer of Harmonic and Percussive Components in Music Signals, *Proc. 9th International Conference for Music Information Retrieval Conference (ISMIR)*, pp.139-144 (2008).
- [13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, J., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E.: Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, Vol.12, pp.2825-2830 (2011).
- [14] Sawada, S., Fukayama, S., Goto, M. and Hirata, K.: TransDrums: A Drum Pattern Transfer System Preserving Global Pattern Structure., *Proc. 44th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (2019).
- [15] Southall, C., Stables, R. and Hockman, J.: Automatic Drum Transcription Using Bi-Directional Recurrent Neural Networks, *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR)*, pp.591-597 (2016).
- [16] Southall, C., Stables, R. and Hockman, J.: Automatic Drum Transcription for Polyphonic Recordings Using Soft Attention Mechanisms and Convolutional Neural Networks, *Proc. 18th International Society for Music Information Retrieval Conference (ISMIR)*, pp.606-612 (2017).
- [17] Vogl, R., Dorfer, M. and Knees, P.: Recurrent Neural Networks for Drum Transcription, *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR)*, pp.730-736 (2016).
- [18] Yoshii, K., Goto, M., Komatani, K., Ogata, T. and Okuno, H.G.: Drumix: An Audio Player with Real-time Drum-part Rearrangement Functions for Active Music Listening, *IPSJ Journal*, Vol.48, No.3, pp.1229-1239 (2007).
- [19] 後藤真孝, 吉井和佳, 藤原弘将, Mauch, M., 中野倫靖: Songle: 音楽音響信号理解技術とユーザによる誤り訂正に基づく能動的音楽鑑賞サービス, *情報処理学会論文誌*, Vol.54, No.4, pp.1363-1372 (2013).
- [20] 澤田 隼, 深山 覚, 後藤真孝, 平田圭二: 大域的構造を考慮したドラムパターンの差し替えシステム, *研究報告音楽情報科学* (2018).



澤田 隼 (学生会員)

2018年公立はこだて未来大学大学院システム情報科学研究科博士前期課程修了。同年同大学大学院博士後期課程に進学。現在、博士後期課程3年。日本学術振興会特別研究員 (DC2)。2018年度情報処理学会山下記念研究

賞受賞。音楽情報処理に関する研究に従事。



深山 覚 (正会員)

2013年東京大学大学院情報理工学系研究科システム情報学専攻博士課程修了。博士 (情報理工学)。日本学術振興会特別研究員 (DC2), 産業技術総合研究所研究員を経て, 2017年より同研究所主任研究員。専門は音楽情報科学。2009年度情報処理学会山下記念研究賞受賞。情報処理学会音楽情報科学研究会計算論的生成音楽学 (GMI) ワーキンググループ2代目代表。



後藤 真孝 (正会員)

1998年早稲田大学大学院理工学研究科博士後期課程修了。博士 (工学)。現在、産業技術総合研究所首席研究員兼メディアコンテンツ生態系プロジェクトユニット代表。2016年からJST ACT-I「情報と未来」研究総括, 2017年から日本学術会議連携会員等を兼任。JST ACCEL 研究代表者。日本学士院学術奨励賞, 日本学術振興会賞, ドコモ・モバイル・サイエンス賞基礎科学部門優秀賞, FIT 船井業績賞, 科学技術分野の文部科学大臣表彰若手科学者賞, 星雲賞等, 51件受賞。



平田 圭二 (正会員)

1987年東京大学大学院工学系研究科情報工学専門課程博士課程修了。工学博士。同年NTT基礎研究所入所。1990~1993年(財)新世代コンピュータ技術開発機構(ICOT)に出向。2011年より公立はこだて未来大学教授。1993年音楽情報科学研究会初代主査。2005~2007年および2011~2013年理事。2010年~2015年デジタルプラクティス誌編集委員長。2001年度, 2011年度論文賞, 2003年度山下記念研究賞。現在, 音楽情報学に加え, MaaS, メンタルヘルスのICT支援の研究に従事。人工知能学会, 日本ソフトウェア科学会, サービス学会各会員。本会シニア会員。