Vol.2018-MUS-119 No.43
Vol.2018-SLP-122 No.43
2018/6/17

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

澤田 隼[1,a)]　深山　覚[2]　後藤 真孝[2]　平田 圭二[3]

概要：本稿では楽曲のドラムトラックを大域的構造を考慮しながら差し替えるシステムを提案する．典型的なドラムトラックは，一定のドラムパターン演奏を繰返すなかで繰返しの最後や楽曲構造のつなぎ目でパターンに変化を加えたり（フィルイン），A メロ，B メロなどの楽節ごとに異なるドラムパターンを叩いたり，大域的な構造に依存して演奏が変化する．しかし，ドラムトラックを小節毎に別のドラムパターン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に依存した変化を実現することが難しかった．そこで，本システムでは大域的構造に依存した複数のドラムパターンを楽曲から抽出し，その楽曲の持つコンテキストを保存しながらドラムパターンの差し替えを行う．まず，システムは A と B の 2 つの楽曲を入力として受け付け，各曲から複数の代表的なドラムパターンを各曲それぞれから抽出する．次に，楽曲 A と B の間でコンテキストにおける相対位置が類似している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同士を置き換える．適切な対応ペアを見つけるために，まず抽出されたドラムパターン間の遷移確率分布によって，大域的ドラムパターン構造をモデル化する．対応するペアを決定するために，楽曲 A と B のそれぞれのドラムパターンの遷移確率分布間のカルバック・ライブラー情報量の合計が最小になるものを選ぶ．提案システムを用いて，楽曲のコンテクスト内での役割を適切に保持したドラムパターンの差し替えが可能になり，聴覚的に自然でかつ，雰囲気が大きく異なる楽曲を生成するできることを確認した．

1. はじめに

従来の音楽鑑賞はそのまま再生して楽しむだけの受動的な体験であった．近年，能動的なインタラクションを伴って音楽を鑑賞する能動的音楽鑑賞（Active music listening）[7] の需要が広がっている．例えば，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] 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．中村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に置き換えることができる．小野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．吉井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事ができるシステム Drumix [15] を提案した．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・未経験者共に思いがけない音楽経験をもたらし，好評であったという結果が得られている．本稿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．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．例えば，楽曲の構造の変わり目には，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）が挿入される．しかしながら，ドラムトラックを小節毎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った．

これを踏まえて本稿では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．本システムを実現するためには大域的なドラムパターンの認識と，適

[1] 公立はこだて未来大学大学院
Graduate School of Future University Hakodate
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)
[3] 公立はこだて未来大学
Future University Hakodate
[a)] b1012046@gmail.com

© 2018 Information Processing Society of Japan

Vol.2018-MUS-119 No.43
Vol.2018-SLP-122 No.43
2018/6/17

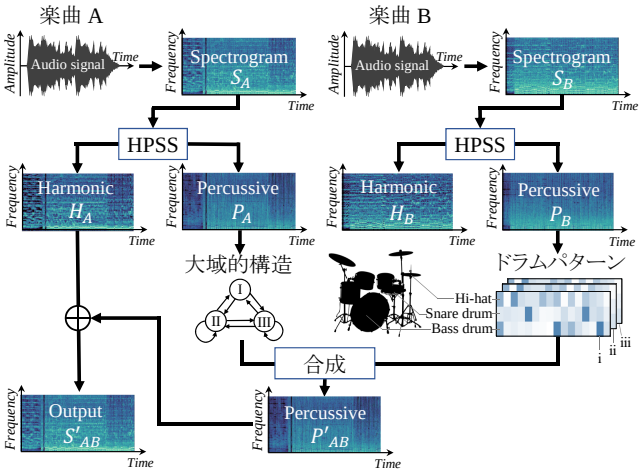

図 1　システム概要

切なドラムパターン同士の対応付けが必要になる．これらを実現するために，楽曲のそれぞれ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ドラムパターンのペアを決定する．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，聴覚的に自然でかつ，ドラムパターンが差し変わったことで雰囲気が大きく異なる楽曲の生成が可能になる．

2. ドラムパターンの差し替えシステム

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（楽曲 A と楽曲 B）を入力し，楽曲 A のドラムパターンを楽曲 B のドラムパターンに差し替えるシステムである．楽曲 A 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，楽曲 A のドラムパターンをそれと対応付けられた楽曲 B のドラムパターンに差し替える．まず，入力された A と B 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ドラムパターンを抽出する．次に，楽曲 A と B 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楽曲 A と B 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．本システムでは小節毎に処理を行うため，事前に楽曲 A と B 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．

2.1 ドラムパターン特徴量の抽出

入力された楽曲の 1 小節毎のドラムパターンの特徴量の抽出方法について述べる．まず，入力された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する．分離された非調波成分の各時刻ごと 1 時刻前とのスペクトルの差分（増加量）を求め，16 次元へと量子化することで次元圧縮したものをドラムパターンの特徴量とする．小節ごとに分割する際，AIST Annotation for the RWC Music Database [6] の Beat structure を使用した．抽出方法の概要を図 **2** に示す．

まず，入力された 2 つの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）を行う．次に，得られた 2 つのスペクトログラムのそれぞれに対して調波・非調波音分離（Harmonic/Percussive sound separation: HPSS）を用い，調波成分と非調波成分のスペクトログラムへと分離する．スペクトログラム上では調波成分は時間軸方向に滑らかであり，打楽器などの非調波成分は周波数軸方向に滑らかであるという特性を持つ．HPSS はこの特性を利用して，入力スペクトログラム $\boldsymbol{S}$ を，調波成分 $\boldsymbol{H}$ と非調波成分 $\boldsymbol{P}$ に分解する処理である．HPSS は librosa [10] の `librosa.decompose.hpss` 関数を使用した．`librosa.decompose.hpss` 関数は Driedger [4] と Fitzgerald [5] のアルゴリズムを基に実装されている．

次に，分離されたそれぞれの楽曲の非調波成分 $\boldsymbol{P}$ 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 $\boldsymbol{D}_{k,t}$ を全周波数 bin 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．

$$\mathbf{SF}_t = \sum_k \boldsymbol{D}_{k,t} \tag{1}$$

$$\boldsymbol{D}_{k,t} = \begin{cases} \boldsymbol{S}_{k,t} - \boldsymbol{S}_{k,t-1} & (\boldsymbol{D}_{k,t} > 0) \\ 0 & (otherwise) \end{cases} \tag{2}$$

ここで $\boldsymbol{S}_{k,t}$ は時刻 t におけるスペクトログラムの周波数 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 16 次元ベクトル $\boldsymbol{F}_i = \{f_i\}_{i=1}^{16}$ をドラムパターンの特徴量として用いる．予備実験より，$m = 8$ とした．ドラムパターン特徴量 $\boldsymbol{F}_i$ は次の式で与えられる．

$$\boldsymbol{F}_i = \sum_{\tau=-m}^{m} \mathbf{SF}_{u_i+\tau}, \tag{3}$$

ここで $\boldsymbol{U} = \{u_i\}_{i=1}^{16}$ は 16 分音符レベルのビート時刻系列を表している．例えば, 10 フレーム毎に均等に 16 分音符単位のビート位置がある場合，$\boldsymbol{U} = \{10, 20, 30, \cdots, 160\}$ となる．

© 2018 Information Processing Society of Japan

Vol.2018-MUS-119 No.43
Vol.2018-SLP-122 No.43
2018/6/17

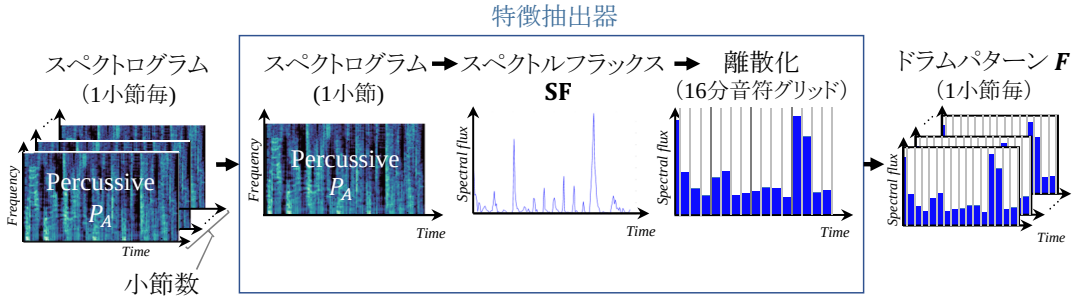

図 2　ドラムパターン特徴量の抽出

2.2　ドラムパターンの対応付けアルゴリズム

楽曲 A と楽曲 B の差し替えるドラムパターン同士の対応付けを行う概要を図 3 に示す．第 2.1 節で述べたドラムパターン特徴量を用いて，それぞれの楽曲についてクラスタリングを行う．クラスタリングの結果得られた 2 つの楽曲におけるクラスタ同士の適切な対応付けを行う．

まず 2.1 節によって得られた 16 次元のドラムパターン特徴量を楽曲中のすべての小節について求めて，それらの k-means クラスタリングを行う．k-means クラスタリングは python のライブラリ `scikit-learn` [13] として実装されている Lloyd のアルゴリズムを用いた．小節ごとに分割された楽曲のドラムパターンは，クラスタリングによってラベル付けされる．クラスタ数を N_{cluster} とし，ラベル集合を $\boldsymbol{C} = \{c_n\}_{n=1}^{N_{\mathrm{cluster}}}$ と表す．このラベルを用いて，小節単位のラベル列 $\boldsymbol{L} = \{l_n\}_{n=1}^{N_{\mathrm{bar}}}$ が求まる．ここでラベル l_n はクラス集合 $\boldsymbol{C}$ に属し（$l_n \in \boldsymbol{C}$），N_{bar} はその楽曲内の小節数を表す．たとえば，$N_{\mathrm{cluster}} = 4$ の場合，$\boldsymbol{C} = \{0, 1, 2, 3\}$ となる．楽曲の長さが 4 小節分で，ドラムパターンが 1 小節ごとに変わる場合は $\boldsymbol{L} = \{0, 1, 2, 3\}$ となり，ドラムパターンが全て同じ場合は $\boldsymbol{L} = \{0, 0, 0, 0\}$ のようになる．

上述した方法によって得られたラベル列の 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 のパターン間遷移確率行列 $\boldsymbol{T}_X$（$N_{\mathrm{cluster}} \times N_{\mathrm{cluster}}$ 行列）は以下のようになる．

$$\boldsymbol{T}_X = \begin{pmatrix} t_{11} & t_{12} & \dots & t_{1j} \\ t_{21} & t_{22} & \dots & t_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ t_{i1} & t_{i2} & \dots & t_{ij} \end{pmatrix}$$

ここで，$\boldsymbol{T}_X$ の要素である t_{ij} はラベル c_i から遷移する際の遷移先 c_j についての遷移確率 $P(c_j \mid c_i)$ を表している．$\boldsymbol{T}_X$ の n 行目が c_n の離散確率分布になっており，楽曲Aの遷移確率行列 $\boldsymbol{T}_A$ の n 行目を $P_n(i)$ と表し，楽曲Bの遷移確率行列 $\boldsymbol{T}_B$ の m 行目を $Q_m(i)$ と表す．

次に，大域的な構造を保持したドラムパターンのペアを見つけるために，2 つの確率分布間のカルバック・ライブラー情報量（Kullback - Leibler divergence: KLD）を元に計算される JensenShannon divergence（JSD）を計算し，その和が最小となるペアを決定する．これは，楽曲内で同じ役割を持つドラムパターンは同じような遷移確率を有するという仮説に基づくものである．確率分布 P の Q に対する KLD は以下のように与えられる．

$$\mathrm{KLD}(P \,||\, Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{4}$$

今回は以下のように対称化した KLD である JSD を使用した．

$$\frac{1}{2}(\mathrm{KLD}(P \,||\, Q) + \mathrm{KLD}(Q \,||\, P)) \tag{5}$$

楽曲 B の遷移確率行列の各行を入れ替えながら JSD を求め，その全ての対応付けの和が最小になるペアを探索する．こうして得られたドラムパターンのペアを使用して，楽曲 A のラベル列 $\boldsymbol{L}_A$ は対応づいた楽曲 B のラベルに差し替えられたラベル列 $\boldsymbol{L'}_A$ が作成される．

2.3　ドラム採譜とドラムトラックの生成

第 2.2 節によって得られたラベル列 $\boldsymbol{L'}_A$ に沿って，楽曲 B のドラムパターンのオーディオを生成し，楽曲 A の調波成分と合成する（図 4）．まず，クラスタリングによって形成された楽曲 B のクラスタのセントロイドに最も近いドラムパターンを代表ドラムパターンとして抽出する．次に，代表パターン $\boldsymbol{P}_B$ のドラム採譜を行う．HPSS によって得られる非調波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ため，ドラムを採譜しその結果からドラムパターンのオーディオを再生成することでノイズの含まれていないドラムパターンを生成している．

ドラム採譜は非負値行列因子分解（Non-negative Matrix Factorization: NMF）を用いて行う．NMF は非負値行列を 2 つの非負値行列に分解するアルゴリズムであり，本研究では非調波成分のスペクトログラム $\boldsymbol{P}$ を基底行列 $\boldsymbol{W}$ とアクティベーション行列 $\boldsymbol{H}$ へと分解する（$\boldsymbol{P} \approx \boldsymbol{W} \cdot \boldsymbol{H}$），$\boldsymbol{W}$ は基底を表し，$\boldsymbol{H}$ は基底をどれだけの重みで結合するか（アクティベーション）を表す．NMF は Cichocki らのアルゴリズムを参考に実装した [1]．多くのドラム採譜の研究では NMF が使用されており [3], [14], [16]，これらにおいては基底の初期値としてドラムセットの音源のスペク

© 2018 Information Processing Society of Japan

Vol.2018-MUS-119 No.43
Vol.2018-SLP-122 No.43
2018/6/17

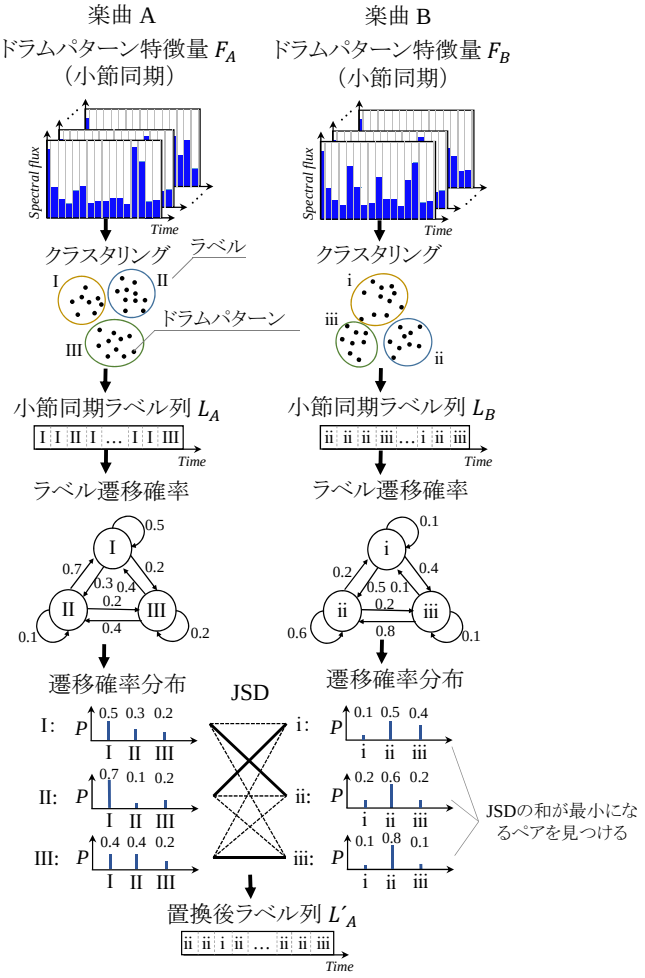

図 3　ドラムパターンの対応付け

トログラムを与えることでドラムの各楽器音に対応する基底を学習している．これらにならい，ドラムだけの音源のスペクトログラムを初期値とし，バスドラム，スネアドラム，ハイハットに対応する基底が学習されるようにした．今回は RWC 研究用音楽データベースの楽器音データベース（RWC-MDB-I-2001）を初期値として使用し，予備実験の結果基底数は 4 にした．それぞれの 3 つの基底が各ドラムセットの楽器に対応するが，4 つ目はノイズの基底として学習されるため，このノイズの基底以外を用いる．そうして得られたアクティベーション $\boldsymbol{H}_B$ をドラムパターンに変換するために，式 (3) と同様に $\boldsymbol{H}_B$ を 16 分音符のビート位置の周位 m フレームにわたって足し合わせて 16 分音符グリッドで離散化する．この離散化されたアクティベーションを正規化したものを，各音源の強さ（ベロシティ）としてそれぞれの音源（バスドラム，スネアドラム，ハイハット）に掛け合わせ，代表ドラムパターンの音響信号を生成する．最終的に，このように得られたドラムパート $\boldsymbol{P'}_B$ をラベル列 $\boldsymbol{L'}_A$ に沿って楽曲 A の調波成分の音響信号と足し合わせたものを出力する．

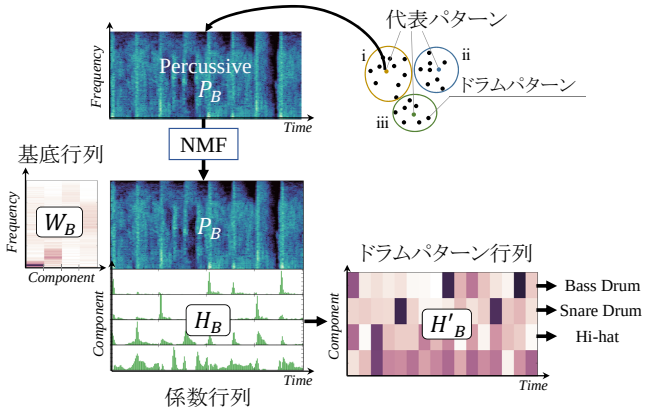

図 4　NMF を用いたドラム採譜

3. ドラムパターン差し替えの評価実験

3.1 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）2 つの楽曲においてフィルイン同士が正しく対応付けができているかを評価する実験．

実験（2）頻出する遷移パターンを認識し，2 つの楽曲においてそれらが対応付けら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）[8] の楽曲を使用した．（No. 1, No. 2, No. 3）ドラム演奏経験が 3 年以上ある 2 名が作成しクロスチェックしたラベル列を正解データとして評価を行う．正解データを図 **5** に示す．図 5 中の数字は 1 小節ごとのリズムパターンのラベルを表しており，フィルインの小節には数字の上にドットを付与した．また，同じ種類の線で囲まれたラベル列は楽曲構造上の区分であることを意味する．例えば，楽曲 No. 1 の 2 重線はサビを，点線はイントロを表している．ドラムパターンのクラスタ数を 6 として実験を行った．

2 つの実験（1）と（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．楽曲 A のラベル集合を $\boldsymbol{C} = \{c_n\}_{n=1}^{N_{\text{cluster}}}$，楽曲 B のラベル集合を $\boldsymbol{D} = \{d_n\}_{n=1}^{N_{\text{cluster}}}$ とあらわす．このとき，提案手法によって得られた対応関係を $M(c_i) = d_j$ とあらわす．たとえば，楽曲 A のラベル $0 \in \boldsymbol{C}_{\text{set}}$ と楽曲 B のラベル $2 \in \boldsymbol{D}_{\text{set}}$ が対応付けされた場合は $M(0) = 2$ とあらわされる．

実験（1）において，2 つの楽曲のフィルイン同士が対応づい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．まず，2 つの楽曲のフィルイン位置を表す正解データ $f(c_i)$ を作成する．$f(c_i)$ はフィルインであるかそうでないかの 2 値 $\{0, 1\}$ をとる．たとえば，$0 \in \boldsymbol{C}_{\text{set}}$ がフィルインであれば，$f(0) = 1$ となり，そうでなければ $f(0) = 0$ となる．フィルイン一致率 $F(M)$ を以下のように

© 2018 Information Processing Society of Japan

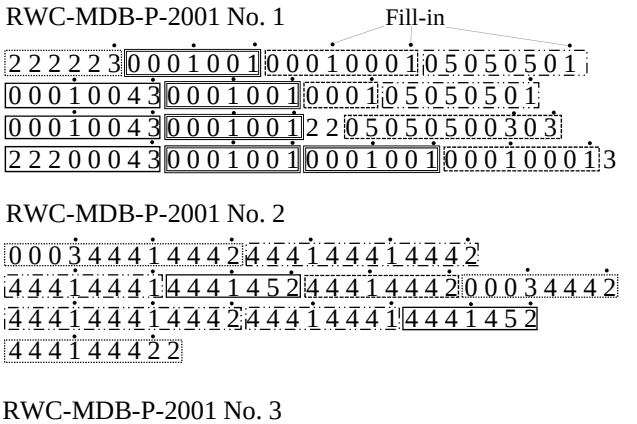

図 5　使用した正解データ

定義する．

$$F(M) = \frac{\sum_i^{N_{\text{cluster}}} f(c_i) f(M(c_i))}{N_{\text{fill-in}}} \qquad (6)$$

ここで $N_{\text{fill-in}}$ は楽曲中に含まれるフィルインの種類の数をあらわす．$F(M)$ は $0 \leq F(M) \leq 1$ の値をとり，2 つの楽曲の全てのフィルイン同士が対応づいている場合に最大値 1 をとる．

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と対応付けの整合性を調べる尺度として，頻度整合性 $E(M)$ を定義する．たとえば，楽曲 A と楽曲 B 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，すべてのパターンが 1 対 1 で対応付くという制約が理由で必ずしも整合性が取れない可能性があるため，整合性の尺度で評価する．頻度整合性 $E(M)$ を以下のように定義する．

$$E(M) = \sum_r^{10} w_x(r) \frac{1}{2} (R_0(r) + R_1(r)) \qquad (7)$$

ここで $R_i(r)$ と $w_x(r)$ は以下の式で与えられる．

$$R_i(r) = \begin{cases} 1 & (M(x_{ri}) = y_{ri}) \\ 0 & (otherwise) \end{cases}$$

$$w_x(r) = \frac{v_{x_{r0}x_{r1}}}{\sum_r v_{x_{r0}x_{r1}}} \qquad (8)$$

c_i から c_j への遷移する頻度を $v_{c_i c_j}$ と表す．頻度順にソートしたバイグラムの遷移パターンを $\{x_{r0}x_{r1}\}_{r=1}^{10}$ と表す．ここで r は頻度の順位を表している．たとえば，頻度の順位が 3 位の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．この尺度は提案手法によって得られた対応関係がバイグラムの頻度ランキングと整合性が取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，対応関係が頻度のランキングと一致していた時に最大値 1 をとる．

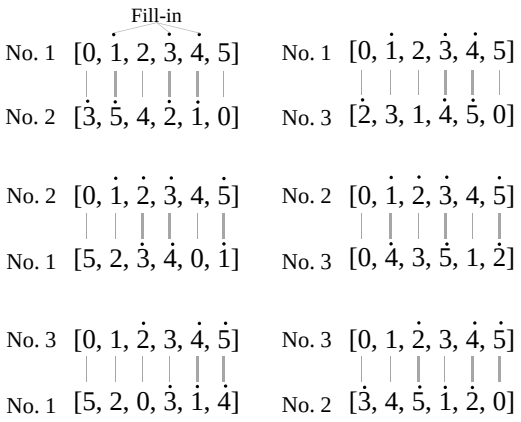

図 6　対応関係の結果

表 1　ドラムパターンのバイグラム出現頻度．括弧内が頻度を表す．

No. 1	No. 2	No. 3
00 (34)	44 (38)	11 (18)
01 (19)	41 (13)	33 (18)
10 (17)	14 (13)	00 (12)
50 (9)	42 (6)	34 (9)
05 (9)	24 (6)	43 (8)
22 (7)	00 (4)	14 (5)

3.2　実験結果

システムが出力した対応関係の結果を**図 6** に示す．図左上は RWC の楽曲 No. 1 に No. 2 を対応付けした結果をあらわしており，No. 1 の 0 と No. 2 の 3 が対応づいている（$M(0) = 3$）ことを意味する．

ドラムパターンラベルのバイグラム出現頻度を**表 1** に示す．楽曲毎に頻度が高い順にそのバイグラムパターンを並べ，括弧内にその頻度を示した．たとえば，楽曲 No. 1 ではバイグラムの 00 が最も頻出しており，その頻度が 34 回あったことを意味する．これはドラムパターンの自己遷移が最も多いことを意味する．

頻度整合性 $E(M)$ とフィルイン一致率 $F(M)$ を**表 2** に示す．たとえば，（a）は RWC の楽曲 No. 1 を No. 2 のドラムパターンで差し替えた場合の $E(M)$ が 0.05 で，$F(M)$ が 1.0 であることを表す．（d）の No. 2 を No. 3 のドラムパターンで差し替えたときが最も $E(M)$ が高く（0.53），（a）の No. 1 を No. 2 で置き換えたものが最も $E(M)$ が低かった．

最後に，システムが出力したクラスタリング結果を**図 7** に示す．正解データと比べて，No. 1 と No. 3 ではフィルインが正しく検出できていない．No. 2 ではラベル 0 がフィルインの位置に来ることが多く，比較的良くフィルインとラベルが一致している．

© 2018 Information Processing Society of Japan

表 2　頻度整合性とフィルイン一致率．括弧内はフィルインの全ての個数のうち何個一致したかを表す．

楽曲のペア	$E(M)$	$F(M)$
(a) No. 1 - No. 2	.05	1.0 (3/3)
(b) No. 1 - No. 3	.17	.67 (2/3)
(c) No. 2 - No. 1	.44	.75 (3/4)
(d) No. 2 - No. 3	.53	.75 (3/4)
(e) No. 3 - No. 1	.06	.67 (2/3)
(f) No. 3 - No. 2	.39	.67 (2/3)

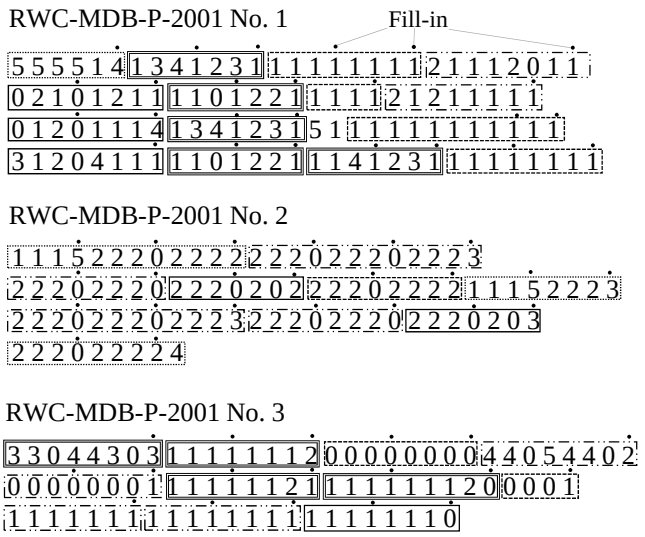

図 7　クラスタリング結果

4. 考察

表 2 における（c），（d）の楽曲のペアでは No. 2 のフィルインの数が 4 なのに対し，No. 1 と No. 3 のフィルインの数は 3 であり，良くパターン同士が対応づいていると言える．ドラムパターンの大域的な構造を保持しながら，ドラムパターンを差し替えた楽曲が生成できたと言える．

ただし，このようにフィルインやパターンの数は楽曲によって違うため，1 対 1 対応ではフィルインと違うものが対応づいてしまう問題がある．ドラムパターンは階層的なパターンのクラスタを形成すると考えられ，たとえば 1112111233343334 というラベル列があった場合，ラベル 2 とラベル 4 をフィルインの小節だとすると，ラベル 2 はラベル 1 のドラムパターンに変化を加えたパターンであることが多い．同様にラベル 4 もラベル 3 のドラムパターンに変化を加えたパターンであることが多いため，ラベル 1 とラベル 3 の類似度よりもラベル 1 とラベル 2 の類似度の方が高いと考えられる．そのため，階層的なクラスタリングを行い，1 対多対応にする必要があると考えられる．また，楽曲 No. 2 と No. 3 のように，リズムパターンの構造が似ている曲は $E(M)$ の値が比較的高かったが，楽曲 No. 1 のように 0505 とパターンを交互に繰り返す様な曲ではこの尺度では低い値を示し，パターンの構造の違いに応じたパターンの対応付けが必要であるとわかった．

5. おわりに

本稿では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を提案した．ドラムパターンの遷移確率に着目し，遷移確率分布の距離が小さくなるように 2 つの楽曲のドラムパターンを対応付けることで，大域的な構造を保持しながら別のドラムパターンに差し替えることが可能になった．今後の課題はドラムパターンのクラスタリングと音響生成結果の品質を向上することと，ユーザからのフィードバックを得て定性的な評価を行うことである．

謝辞

本研究の一部は JST ACCEL (JPMJAC1602) の支援を受けた．

参考文献

[1] Cichocki, A. and Phan, A.-H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations, *IEICE transactions on fundamentals of electronics, communications and computer sciences*, Vol. 92, No. 3, pp. 708–721 (2009).
[2] Davies, M. E., Hamel, P., Yoshii, K. and Goto, M.: AutoMashUpper: Automatic creation of multi-song music mashups, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 22, No. 12, pp. 1726–1737 (2014).
[3] Dittmar, C. and Gärtner, D.: Real-Time Transcription and Separation of Drum Recordings Based on NMF Decomposition., *DAFx*, pp. 187–194 (2014).
[4] Driedger, J., Müller, M. and Disch, S.: Extending Harmonic-Percussive Separation of Audio Signals., *ISMIR*, pp. 611–616 (2014).
[5] Fitzgerald, D.: Harmonic/percussive separation using median filtering, *13th International Conference on Digital Audio Effects (DAFX10)* (2010).
[6] Goto, M.: AIST Annotation for the RWC Music Database., *ISMIR*, pp. 359–360 (2006).
[7] Goto, M.: Active music listening interfaces based on signal processing, *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, Vol. 4, IEEE, pp. IV–1441 (2007).
[8] Goto, M., Hashiguchi, H., Nishimura, T. and Oka, R.: RWC Music Database: Popular, Classical and Jazz Music Databases., *ISMIR*, Vol. 2, pp. 287–288 (2002).
[9] Goto, M., Yoshii, K., Fujihara, H., Mauch, M. and Nakano, T.: Songle: A Web Service for Active Music Listening Improved by User Contributions., *ISMIR*, pp. 311–316 (2011).
[10] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E. and Nieto, O.: librosa: Audio and music signal analysis in python, *Proceedings of the 14th python in science conference*, 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, *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, IEEE, pp. 7470–7474 (2014).
[12] Ono, N., Miyamoto, K., Kameoka, H. and Sagayama, S.:

© 2018 Information Processing Society of Japan

A Real-time Equalizer of Harmonic and Percussive Components in Music Signals., *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. et al.: Scikit-learn: Machine learning in Python, *Journal of machine learning research*, Vol. 12, No. Oct, pp. 2825–2830 (2011).

[14] Wu, C.-W. and Lerch, A.: Drum Transcription Using Partially Fixed Non-Negative Matrix Factorization with Template Adaptation., *ISMIR*, pp. 257–263 (2015).

[15] 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, *Information and Media Technologies*, Vol. 2, No. 2, pp. 601–611 (2007).

[16] Yoshii, K., Goto, M. and Okuno, H. G.: Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15, No. 1, pp. 333–345 (2007).

© 2018 Information Processing Society of Japan