

## PlaylistPlayer:

### 再生終了時刻と再生方針が指定できる音楽プレイリスト再生インターフェース

中野 倫靖    濱崎 雅弘    後藤 真孝\*

**概要.** 本論文では、実世界での再生終了時刻を指定すると、その時刻までに音楽プレイリストの全楽曲が部分再生されてちょうど終了する再生インターフェース PlaylistPlayer を提案する。終了時刻の指定は、時間制約下における音楽鑑賞や時計の代替手段等の音楽活用として有用である。部分再生の方針は、楽曲の自動分析の結果や Web マイニングの結果、再生回数を活用して簡単に指定できるだけでなく、再生中にユーザの介入（一時停止、別の曲の再生）に伴って残り時間が減少しても、動的に再生されて終了時刻を超えることはない。従来、総再生時間を指定できるプレイリストの自動生成や隙間時間に合ったコンテンツ推薦に関する研究があったが、既存プレイリストの総再生時間は扱えず、また、ユーザの介入が考慮されていなかった。さらに PlaylistPlayer では、従来の楽曲単位の音楽可視化（音楽再生に同期した視覚効果）を拡張し、楽曲同士の類似度やアーティスト同士の関係性の強さに応じた視覚効果を生成する機能も持つ。

## 1 はじめに

音楽は時系列の形態をもったメディアコンテンツであり、楽しむために時間の経過を伴う。音楽プレーヤ等のデバイスを用いた「音楽再生」では、ライブ等の「音楽演奏」とは異なり、音楽の再生時間を実世界での時間から切り離すことができるため、原理的にはいつでもどこでも再生可能である。しかし、聴取者の置かれた状況によっては音楽再生が許されないこともあるため、実際はいつでもどこでも再生することはできない。したがって、音楽鑑賞には常に時間的な制約が伴うため、音楽における再生時間を実世界での時間と「あえて」関連づけることが有用な場合がある。例えば、時間の決まっているパーティ等で複数の楽曲全てを指定時刻に終了するように再生したり、隙間時間で最近の人気曲（週間ランキング等）をチェックする応用が考えられる。さらに、作業における終了時刻を知る手段等、実生活における音楽活用の可能性も広がる。

このような問題意識に関連した従来研究やアプリケーションとして、ユーザによる楽曲毎の評価を実世界の時刻と対応付けて記録しておくことで、現在から指定した時刻までの時間に合った音楽プレイリストを生成する研究 [10] や、指定した合計再生時間のプレイリストを生成するアプリケーション [15] があった。しかし、時間に合うプレイリストを生成できても、既存のプレイリスト（聴きたい曲を自分で指定したプレイリスト）を時間に合わせて再生することはできなかった。また、メディアコンテンツの再生を待ち時間に重ねたり [18]、空き時間に挟み

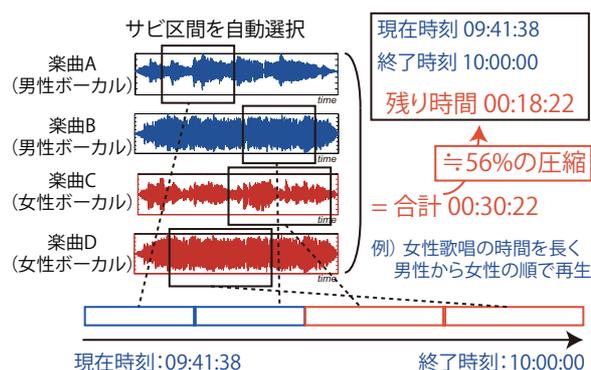


図 1. 終了時刻を指定して楽曲を部分的に再生。自動指定された楽曲の特性（男女度など）に基づいて、個々の楽曲の長さや再生順序を制御できる。

込んだり [19] する、隙間時間の活用に関する研究があった。動画の高速鑑賞 [17] や音楽要約（例えば [14] やダイジェスト再生機能<sup>1</sup>）も、そうした隙間時間におけるコンテンツ活用へ応用可能である。しかし、プレイリストのような複数コンテンツの総再生時間を考慮した研究はなかった。既存のプレイリストを時間に合わせて再生するためには、従来技術・インタラクションのみでは不十分で、個々の楽曲を部分的に再生させる必要がある。

さらに、以上で述べた従来研究やアプリケーションでは、再生中のユーザの行動が考慮されていなかった。つまり実際には、一時停止して席を外したり、別の曲を今すぐ再生したりしたくなることもある。これは、デジタル化された音楽コンテンツならで楽しみ方、ユーザの行動といえる。しかし、実世界の

Copyright is held by the author(s).

\* Tomoyasu Nakano, Masahiro Hamasaki, Masataka Goto, 産業技術総合研究所 (AIST)

<sup>1</sup> SONY ウォークマンの「ZAPPIN 再生」など。

時間は止まることなく進み続けるため、もともとの再生方針を動的に変更しない限り、このようなユーザの介入に対応することができない。

そこで本研究では、複数の楽曲の再生時間を実世界の時間と関連づけて制御できる、音楽プレイリスト再生インタフェース PlaylistPlayer を提案する (図 1)。PlaylistPlayer では、プレイリストを「いつまで」再生し、各曲の「どこから」「どれくらい」「どのように」再生するか (再生する箇所と時間、順序) の方針を指定できる。これは、自動推定された楽曲の音楽要素や特性、ユーザの曲への親密度に基づいて指定する。ユーザはその再生方針に対していつでも介入でき、そのたびに PlaylistPlayer が再生方針を動的に再生する。

さらに、再生中の楽曲が他の楽曲と比較してどういう位置付けにあるかを、自動推定された楽曲間の類似度やアーティスト間の関係性の強さに基づいた、音楽再生に同期した視覚効果 (本論文では以降、ビジュアライザと呼ぶ) によって可視化する。従来、プレイリストの再生であっても、ビジュアライザ等の可視化は個々の楽曲のみを対象としていたが、複数の楽曲同士が連動した可視化の可能性を提案する。

## 2 PlaylistPlayer: 終了時刻を指定するインタラクティブな複数音楽の再生

本章では、まず PlaylistPlayer の機能に基づいた運用シナリオ例について述べた後、それを実現するための課題と、実際のインタラクションを示す。そして、インタフェースについて説明する。

### 2.1 PlaylistPlayer の運用シナリオ

個人が音楽聴取を目的とした場合とそうでない場合、イベント等の運用を想定したシナリオを述べる。

個人 (音楽聴取が目的) 同一のプレイリストは複数回、複数日聞き続けることも多い。そのような状況において、個々の曲の再生箇所や時間を変えることで、プレイリストの雰囲気を変えて新鮮な気持ちで聴くことができる。

また、毎週更新されるランキングを隙間時間でチェックできる。その際、既に好きな曲と新曲の比率を操作し、徐々に慣れ親しむことができる。それに加え、類似度や関係性などに基づいて、新曲の特性を客観的に把握できる。

個人 (音楽聴取以外が目的) 作業や勉強・トレーニング等において「終了時刻付 BGM」を作成できる。この際、自分の今の気分や作業内容に合った楽曲だけを利用できる。

イベント 時間の決まっているパーティで、進行に特化したプレイリストを流したい、といった

状況で利用できる。プレイリスト中の特定の曲をいつでも再生できる。

また、思い出の曲をもれなく再生して流したい、というイベントで利用できる。途中、挨拶や乾杯などで再生を中断する状況は起こりうるが、そういった介入にも柔軟に対処できる。

### 2.2 PlaylistPlayer 実現の課題

PlaylistPlayer を実現するためには、まず既存のプレイリストをユーザの介入にも対応しながら時間通りに再生する (課題 1) 必要がある。さらに、前述した運用シナリオのように PlaylistPlayer をより実用的とするためには、プレイリストの再生方針を指定する (課題 2) ことで雰囲気を変えたり、楽曲の特性や楽曲間の関係性を提示する (課題 3) ことでユーザが曲を理解する支援も重要である。

課題 1 においては、再生方針の決定と、それを動的に再生するインタラクションによって解決する。また、課題 2 および課題 3 においては、再生回数など楽曲に直接的に付与されるメタデータに加え、音楽音響信号解析や Web マイニングなどを活用して自動的に分析する。多様な観点から課題を実現することで、ユーザ毎の違いを吸収することを考える。

具体的には、まず、多くの人にとって聞き分けやすいボーカルに着目し、その性別に基づいて再生方針を決定するために「男女度」(男声・女声らしさ) を用いる。また、プレイリスト中で個性的な曲に着目したり、個々の楽曲の属性を把握したりするためにプレイリストと個々の楽曲の類似度として「ありがち度」[31] を用いる。ありがち度とは、ある楽曲に含まれる音楽要素が確率的にどれだけ出やすいかを表現し、本研究ではボーカルの歌声、楽曲中の音色、リズムを対象とする。例えば、女性ボーカルの楽曲が多いプレイリストにおいては、女性ボーカルの楽曲に対する歌声のありがち度が高くなる。

さらに、音響信号処理により楽曲間類似度を、Web マイニングによりアーティスト間の関係の強さを推定する。関係の強さは、例えば「同年代の歌手」「出身地が同じ」「どちらもシンガーソングライターである」といった意味的な情報に基づいて推定する。

### 2.3 音楽プレイリスト再生インタラクション

PlaylistPlayer は、以下の 4 つのインタラクションを実現する。また、プレイリスト再生中に、別の曲を指定したり一時停止しても、指定した時刻で再生終了するよう動的に再生方針が再生される。

いつまで (終了時刻の設定) 「直接入力」で決定できる。再生終了時刻が、プレイリスト中の全楽曲の長さよりも短い場合は、ユーザによって設定された楽曲再生箇所に基づいて部分的に再生される。逆に、再生終了時刻が、プレ



図 2. PlaylistPlayer の実行画面

プレイリスト中の全楽曲の長さよりも長い場合は、プレイリストを2回以上繰り返して再生する。

どこから（楽曲再生箇所の設定）「イントロ」「サビ」「ランダム」を選択できる。再生終了時刻がプレイリストの長さよりも短い場合は、サビを選択すれば効率的に全曲聴くことができる。

どのように（楽曲再生順序の設定）ランダム再生に加え、自動推定された楽曲の特性に基づいて再生順序を変更できる。具体的には、ボーカルの男女度とありがち度が利用できる（後述）。

どれだけ（個々の楽曲の再生時間の調整）「男性ボーカルの楽曲を多めに聴きたい」「あまり聴いたことがない曲の比率を増やしたい」という、プレイリストの多様な聴き方を実現する。具体的には、再生回数（親密度）、男女度、ありがち度に基づいて再生時間の比率を変更できる。

## 2.4 楽曲同士が連動する音楽再生ビジュアルライザ

プレイリストや個々の楽曲の特性を理解する支援のために、楽曲特性の可視化に加えて、楽曲間の類似度・関係性も可視化する。

楽曲間類似度・関係性の可視化 音楽と連動した視覚効果で、楽曲間の類似度やアーティスト間の関係性を可視化する。具体的には、再生中の音楽にはビートに合わせた視覚効果がかかるが、再生されていない楽曲にも、再生中の曲との類似度・関係性に応じて視覚効果がかかって関連の強さを表現する。ユーザは、着目する類似度もしくは関係性を選択できる。

## 2.5 インタフェース

図 2 に、PlaylistPlayer の実行画面を示す。簡単には、まず再生終了時刻を指定することで、プレイリストが再生できる。画面には現在時刻および、その終了時刻までの残り時間も表示される。プレイリストは各楽曲が球体で表現され、その曲に関する分析結果が棒グラフで表される。

各楽曲の球体は、そのボーカルの「男女度」が自動推定されて色づけされる。男性ボーカルの楽曲は男声らしさが高くなって色が青くなり、女性ボーカルの楽曲は女声らしさが高くなって色が赤くなる。また、ビートに合わせた視覚効果がかかる。

次に、各楽曲のありがち度が、ボーカルの歌声、楽曲中の音色、リズムに関して表示され、その楽曲の再生時間（楽曲の長さ）と再生回数が表示される。終了時刻を定めて複数の楽曲を再生する場合、個々の楽曲の長さは重要な要素であり、ここには再生方針に基づいた再生時間が表される。例えば、もとの曲の長さは長くても、ユーザが各曲をどれだけ再生するか方針を決める（2.3 節を参照）と、その長さが短くなることもある。また、再生回数が多いほど、その楽曲に対して親近性を感じると仮定し、「親密度」と呼んで表示する。その他、再生方針の指定やビジュアルライザを操作できる。

## 3 実装

前述した音楽プレイリストの再生インタラクションを実現する方法について説明する。具体的には、再生終了時刻と再生方針の指定、音楽音響信号処理技術、Webマイニング技術について説明する。

ビート及びサビ区間は、能動的音楽鑑賞サービス Songle [25] のモジュールで推定した。

### 3.1 再生終了時刻と再生方針の指定

プレイリストの総再生時間と終了時刻までの残り時間との割合に応じて、各曲の再生時間を決定する。例えば、60分のプレイリストを30分で再生する必要がある場合には、各曲の長さを50%に圧縮する。

再生方針において、再生箇所が「サビ」と指定された場合には、複数の候補が存在するため、ランダムに選択する。また、再生時間は、それぞれパラメータ（再生回数や男女度など）について、プレイリスト中の全曲の総和が1となるように正規化した後、その比例配分によって決定した。

プレイリスト再生中、ユーザによる曲の変更が行われた場合には、変更された曲からプレイリストの最後の曲で、再生時間を上述のように割り振り直すことで対処した。一時停止も同様で、再生が開始された際に、個々の楽曲の再生時間を割り振り直した。

### 3.2 楽曲間類似度の推定

楽曲間類似度やありがち度を推定するために、「ボーカルの歌声」「楽曲中の音色」「リズム」に関する音響特徴量を自動推定し、それらを確率的生成モデルでモデル化した。分析のための個々のパラメータや分析条件は、基本的には文献 [31] と合わせた。

歌声に関しては、混合音中で最も優勢な音高を推定する手法 PreFEst [6] によってメロディーの  $F_0$  (基本周波数) を推定する。その後、20 次倍音までを用いて正弦波合成し、その信号から LPMCC (線形予測メルケプストラム係数) と  $\Delta F_0$  を推定する。最後に、歌声・非歌声 GMM を用いた高信頼度フレーム選択によって、歌声らしさが高いフレームを選択し、特徴量として用いた [11, 31]。音色に関しては、音声認識で通常用いられる特徴量である MFCC (メル周波数ケプストラム係数)、 $\Delta$ MFCC、 $\Delta$ パワーを用い、リズムに関しては Fluctuation Patterns (FP) [12] に基づく音響特徴量を抽出した。それぞれに関して、特徴量をベクトル量子化して潜在的ディリクレ配分法 (LDA: Latent Dirichlet Allocation) [1] を用いたトピック (潜在意味) 分析を行う。

まず、日本の音楽チャートであるオリコン<sup>2</sup> で 2000 ~ 2008 年の上位 20 位以内に登場した楽曲、3278 曲を対象として、トピックモデルを構築した。楽曲 A と楽曲 B の類似度は、楽曲 A のトピック分布から楽曲 B が生成される確率を計算 [31] し、フレーム数で正規化して推定した。

### 3.3 ありがち度の推定

3.2 節の方法で推定したトピックモデルを用いて、各曲のトピック分布を統合することで楽曲集合 (共通) のモデルを推定する [31]。ある特定の音色が楽曲 A と楽曲 B のどちらにも出現した場合、その音色の出現回数はほかの音色に比べて相対的に増えるため、より「ありがち」となる。

ありがち度は、このようにして得られた楽曲集合のモデルを用いて推定できる。楽曲間類似度と同様、楽曲集合のモデルからの各楽曲の生成確率を計算することで推定できる [31]。しかし本稿では、プレイリスト毎に共通構造の推定に用いる楽曲集合が変わるため、計算コストを減らすため、楽曲集合のモデルと楽曲毎のモデルとの分布間距離 (対称カルバック・ライブラ距離: KL2) を計算して推定した。

### 3.4 男女度の推定

男女度は、音楽視聴支援サービス Songrrium [9] における手法と同様の条件で分析し、歌声特徴量は 3.2 節と同様とした。つまり、高信頼度フレーム選択によって LPMCC と  $\Delta F_0$  を特徴量として推定し、これを、Support Vector Machine (SVM) [2] に

カテゴリ: 日本のシンガーソングライター

日本のオルタナティブ・ロック・ミュージシャン | 日本の作曲家

NHK紅白歌合戦歌唱楽曲・作詞者・作曲者 | 和歌山県出身の人物

1969年生 | 存命人物 | L'Arc~en~Cielのメンバー

<http://ja.wikipedia.org/wiki/Hyde> より引用

図 3. アーティストページ (Wikipedia) のカテゴリ例

表 1. 上位カテゴリ (3278 曲のアーティスト)

カテゴリ名	アーティスト数
存命人物	256
日本の歌手	141
NHK 紅白歌合戦出演者	127
日本のシンガーソングライター	97
日本の俳優	81
日本の歌手グループ	75
日本のロック・バンド	68
エイベックス・グループのアーティスト	62
ユニバーサルミュージックのアーティスト	61
東京都出身の人物	48
...	...
1976 年生	15

よって事前に学習しておく。その後、新規楽曲に対して、各時刻で男女それぞれの事後確率を推定する。SVM の学習には、RWC 研究用音楽データベース (ポピュラー音楽) [24] から、男性ソロボーカルの 15 曲 (日本語 13 曲、英語 2 曲) と、女性ソロボーカルの 13 曲 (日本語 11 曲、英語 2 曲) を用いた。

### 3.5 アーティストの関係性の推定

Wikipedia<sup>3</sup> から、各アーティストページのカテゴリを抽出し、共通カテゴリの数を関係性の強さとした。近年、Echonest や MusicBrainz など音楽メタデータ基盤の整備が進んでいるが、これらが扱う情報は楽曲のジャンルや収録アルバムなどが中心となっている。一方で Wikipedia にはアーティストのデモグラフィックな情報や経歴が記載されているため、楽曲の音楽的特徴に由来するものとはまた違った関係性を見つけることができる。そこで本研究では Echonest は Wikipedia のアーティストページ URL を発見するためにのみ用いた。

具体的に得られたカテゴリ例を図 3 に、アーティストページに出現することが多い上位カテゴリを表 1 に示す。3.2 節でと同様の 3278 曲と、そのアーティストを対象とした。所属 (レーベル等) や出身地、活動形態 (シンガーソングライター等) が、複数人に共通して出現することが多いことが分かる。またその他、出生年代に関するものも多かった (2 人以上のアーティストに出現したのは 41 カテゴリ)。

<sup>2</sup> <http://www.oricon.co.jp/>

<sup>3</sup> <http://www.wikipedia.org/>

## 4 考察

従来、「音楽再生」に関しては、その聴き方やインタラクション、プレイリストの自動生成に関する様々な研究が提案されてきた。本章では、従来研究をまとめて本研究の位置付けについて考察する。

### 4.1 音楽再生の関連研究・事例

音楽再生の典型的なインタフェースは、「シークバー」(再生時刻の指定)と「前後の曲への頭出し」である。また、複数の曲をリストアップして、それを順番に聴く「プレイリスト」も典型的である。ここでは、再生を継続し続けるリピート再生と、再生順序を変えるランダム再生が可能である。

これを拡張するために、1曲の再生に関しては、主にシークバーの機能が拡張が研究されてきた。サビの自動推定に基づくサビ出し機能 [23] や、シークバーの自由な切り貼りや連結機能 [29] があった。

一方、プレイリストは、その自動生成に関する研究が数多くなされており [16]、1章で述べた以外で、楽曲間類似度やインタラクションに基づいたものには、楽曲の類似度が似ている順に再生する研究 ([5] 等)、プレイリスト中のユーザの行動(スキップ)やインタラクション(好みかどうかをフィードバック)によって楽曲再生の優先度を定める研究 [13, 22] や、個々の楽曲の再生時間をユーザの嗜好に応じて変化させる研究(気分に合わせて長く再生) [28] がある。

また、複数の楽曲の再生方法に関しては、楽曲間をなめらかにつないで再生する研究 [27] や人によるDJ、複数曲をマッシュアップ(重畳)させる研究(例えば [3, 30])、同一曲の複数人の歌唱を重畳する研究 [26]、大量の楽曲の視聴支援 [7, 20] がある。その他、楽曲間類似度や関係性、メタデータの可視化に関する研究も数多くある(例えば [4, 8])。

### 4.2 本研究の位置付け

関連研究と比較すると PlaylistPlayer では、

- 実世界の時間と関連づいている点
- 再生方針を楽曲の特性に基づいて決定できる点
- 楽曲間の連動した再生を実現する点

が特に異なる。音楽推薦やプレイリストの自動生成など、「何を聴くか」に関する研究・システムはあったが、既存のプレイリストを「どう聴くか」というインタラクションは研究されてこなかった。また、ありがち度 [31] をプレイリストと楽曲の類似度として捉えて活用した点も新しい。

## 5 おわりに

本論文では、音楽プレイリストの再生を対象とし、音楽世界と実世界での時間を関連付けて再生できる

PlaylistPlayer を提案した。また、音楽に同期して楽曲同士が連動するビジュアルライザを提案した。

近年では職業音楽家によって制作された商業楽曲だけでなく、一般の人々による楽曲(CGM)がWeb上で投稿されて共有される文化が広がり、個々人が楽しんで聴くことができる楽曲の数は増加し続けている。同じプレイリストを聴き続けるだけでは飽きてしまうが、逆に、常に変化し続けるプレイリストを聴き続ける場合にも聴取者にとって一種の負担があると考えられる。新しい曲と出会ってそれを好きになれることには喜びがあり、自分の感性を広げることもつながる一方で、好みの形成に時間がかかることがあるため、場合によっては何度か聴く必要がある<sup>4</sup>(その際、好きな曲を聴く場合に比べて快感が少ない)。したがって、例えば、よく知った好きな曲や好きそうな曲を多く含めて楽しみながら(親近性の増加)、初めて聴く曲やあまり聴かないタイプの曲をたまには多めに含める(新奇性の増加)というプレイリストの聴き方は、音楽を楽しむ上で効果的である可能性がある。

今後は、再生方針の決定方法やインタフェース設計等に拡張可能性がある。例えば、再生方針の設計について、現状の PlaylistPlayer では楽曲の再生開始時刻を決定したが、再生終了時刻と次の楽曲の再生開始が滑らかにつながるようにビート推定結果や和音進行推定を活用したり、音色の近さ等を考慮することが考えられる。また、インタフェースの設計に関しては、プレイリストの編集(楽曲の追加や削除等)などの扱いが研究課題である。

## 謝辞

本研究の一部は JST CREST の支援を受けた。

## 参考文献

- [1] D. M. Blei, et al. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003.
- [2] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM TIST*, 2:1–39, 2011.
- [3] M. E. P. Davie, et al. AutoMashUpper: An Automatic Multi-Song Mashup System. In *Proc. ISMIR 2013*, 2013.
- [4] J. L. M. et al. Learning to Embed Songs and Tags for Playlist Prediction. In *Proc. of ISMIR 2012*, 2012.
- [5] A. Flexer, et al. Playlist Generation Using Start and End Songs. In *Proc. ISMIR 2008*, pp. 173–178, 2008.
- [6] M. Goto. A Real-time Music Scene Description System: Predominant-F0 Estimation(略). *Speech Commun.*, 43(4):311–329, 2004.

<sup>4</sup> 曲によっては繰り返し聴くことが、快感が増す手段となることが知られている [21]。

- [7] M. Goto and T. Goto. Musicream: (略). In *Proc. ISMIR 2005*, 2005.
- [8] M. Hamanaka and M. Yoshiya. BandNavi: (略). *Journal of Communication and Computer*, 9(2):114–121, 2012.
- [9] M. Hamasaki, et al. Songrium: (略). In *Proc. WWW 2014*, 2014.
- [10] N.-H. Liu, et al. An Intelligent Music Playlist Generator Based on The Time Parameter with Artificial Neural Networks. *Expert Systems with Applications*, 37:2815–2825, 2010.
- [11] T. Nakano, et al. Vocal Timbre Analysis Using Latent Dirichlet Allocation and Cross-Gender Vocal Timbre Similarity. In *Proc. ICASSP 2014*, 2014.
- [12] E. Pampalk. *Computational Models of Music Similarity and Their Application to Music Information Retrieval*. Ph.D. Dissertation, Vienna Inst. of Tech., 2006.
- [13] E. Pampalk, et al. Dynamic Playlist Generation Based on Skipping Behavior. In *Proc. ISMIR 2005*, 2005.
- [14] G. Peeters, et al. Toward Automatic Music Audio Summary Generation from Signal Analysis. In *Proc. ISMIR 2002*, pp. 94–110, 2002.
- [15] Roderick Cater. MusicTime. <https://itunes.apple.com/jp/app/musictime/id560681752?mt=8>.
- [16] Y. Song, et al. A Survey of Music Recommendation Systems and Future Perspectives. In *Proc. CMMR 2012*, pp. 395–410, 2012.
- [17] 栗原 一貴. 動画の極限的な高速鑑賞のためのシステムの開発と評価. In *Proc. WISS 2011*, 2011.
- [18] 渡邊 恵太他. CastOven: (略). In *Proc. WISS 2008*, 2012.
- [19] 渡邊 恵太他. TimeFiller: (略). In *Proc. WISS 2012*, 2012.
- [20] 大坪 五郎. Goromi-Music (略). In *Proc. WISS 2007*, 2007.
- [21] 榊原 彩子. 音楽の繰り返し聴取が快感情に及ぼす影響 – リズムパターンの冗長性とハーモニーの典型性 –. *Japanese Journal of Educational Psychology*, 44:92–101, 1996.
- [22] 暦本 純一. UniversalPlaylist: (略). 情報処理学会インタラクシオン 2005 論文集, 2005.
- [23] 後藤 真孝. SmartMusicKIOSK: (略). 情報処理学会論文誌, 44(11):2737–2747, 2003.
- [24] 後藤 真孝他. RWC 研究用音楽データベース: 研究目的で利用可能な著作権処理済み楽曲・楽器音データベース. 情報処理学会論文誌, 45(3):728–738, 2004.
- [25] 後藤 真孝他. Songle: (略). 情報処理学会論文誌, 54:1363–1372, 2013.
- [26] 都築 圭太 他. Unisoner: (略). In *Proc. WISS 2013*, 2013.
- [27] 堀内 直明 他. Song Surfing: (略). *PIONEER R&D*, 17(20):2737–2747, 2007.
- [28] 日高 拓朗. ユーザの嗜好に合わせた楽曲再生時間調整システム. 数理解析研究所講究録, pp. 116–119, 2013.
- [29] 青木 惇季, 宮下 芳明. SeekRopes: (略). 情報処理学会インタラクシオン 2011 論文集, pp. 429–432, 2011.
- [30] 宮島 靖. Music Mosaic Generator: (略). In *Proc. WISS 2007*, 2007.
- [31] 中野 倫靖他. 確率的生成モデルに基づく音楽の類似度とありがち度の推定に関する検討. 情報処理学会 研究報告 音楽情報科学研究会, 2014-MUS-104, pp. 1–7, 2014.

## 未来ビジョン

PlaylistPlayer が目指す未来は、音楽聴取における人の嗜好の変化や成長を視野に入れたインタラクシオンである。従来、ユーザの「現在の嗜好」に基づいた研究が数多くなされてきた。例えば、自分の好きな曲に似ている楽曲を検索したり推薦したりする研究は多い。

それに対して、音楽情報検索・推薦の未来のためには、「好みの曲に似ている曲が得られた先」のインタラクシオンについて考える必要があると考える。自分の好きな曲は、それがその曲だから好きなのであって、それに似ていた曲が好きになるとは限らない。また、既に好みのものを聴くだけでなく、むしろ今あまり好みではないものを好きになれた方が、好きな曲の幅が広がる可能性もある。PlaylistPlayer で、既存プレイリストを「少しずつ全て」再生するという方針を採用したのは、以上のような考えに基づいている。PlaylistPlayer では、プレ

イリストの再生成において、各曲が含まれる割合を「ソフトに」決定する。つまり、「この曲を含めて、別の曲は含めない」というような「ハードな」割り当てではない。この再生方針が、ユーザが楽曲に少しでも触れる可能性を増やしている。それによって、好みや再生内容を画一化せず、聴取する楽曲の多様化、嗜好の成長への可能性につなげた。

自分の現在の嗜好の曲だけを聴くのではなく、毎週更新されるランキング、好みではないが話題の新曲、知り合いの好きな曲、多種多様な楽曲にたくさん触れることで、音楽聴取者の好みを拡張することへ貢献したい。例えば、音楽制作者が聴取者に好まれる楽曲を創作する努力だけでなく、楽曲聴取者が制作者の楽曲を好む努力を支援することは、音楽鑑賞の可能性を広げると考える。音楽に関わる人たちが、豊かで多種多様な感動体験を得られるような情報環境の実現を目指していきたい。