

# Song Surfing: 類似フレーズで音楽ライブラリを散策する音楽再生インターフェース

堀内直明<sup>†</sup> 藺田俊行<sup>††</sup> 田中浩司<sup>††</sup> 田中淳一<sup>††</sup>  
長沢秀哉<sup>††</sup> 萩山真一<sup>†</sup>

## Song Surfing: Music playback interface to explore a music library through similar-sounding phrases

Naoaki Horiuchi<sup>†</sup> Toshiyuki Sonoda<sup>††</sup> Koji Tanaka<sup>††</sup> Junichi Tanaka<sup>††</sup>  
Hideya Nagasawa<sup>††</sup> Shinichi Gayama<sup>†</sup>

### 1. はじめに

近年、ハードディスクや半導体メモリの大容量化・小型化に加え、定額制の音楽配信サービスが開始されるなど、個人でも大規模な音楽ライブラリを容易に構築し、楽しめる環境が整ってきた。しかし、楽曲名やアーティスト名等の書誌情報で楽曲を検索する従来のインターフェースでは、個人のライブラリであっても数千曲もの楽曲を把握するのは困難であるため、ユーザは大規模な音楽ライブラリを保有しているにもかかわらず、そのライブラリが持つ多様性や意外性を十分に活用することができなかった。

この問題を解決するため、後藤らは Musicream [1] を開発し、楽曲を効果的に視覚化した上で、自由かつ直感的な操作によって類似楽曲の集合化や過去の聴取楽曲の再現を容易にする、大規模な音楽ライブラリのための新たなインターフェースを提案した。一般に、楽曲は特徴の異なる多様なフレーズの時間的な結合と遷移によって音楽の曲調を表現している。それに対し、Musicream の類似楽曲の集合化は楽曲単位で扱うものであり、楽曲の内部まで踏み込んだ類似性や多様性を扱うことはできなかった。そこで我々は、楽曲の単位より細かいフレーズの特徴に注目し、似ているフレーズを手がかりに多くの異なる楽曲を連続的に楽しむことのできる音楽インターフェース Song Surfing を開発した。本稿では、Song Surfing の特長について解説する。

### 2. Song Surfing の機能

Song Surfing では、類似フレーズを含む複数の楽曲を自動で見つけ出し、各々の楽曲において対応する類似フレーズの位置でそれらを違和感なくつなげて連続的

に再生する。したがって Song Surfing を用いれば、大規模な音楽ライブラリを活用して楽曲のもつより多くの多様性を引き出すとともに、あたかも DJ が演出しているかのような新鮮さと心地よさをユーザに提供できる。これらの機能について、以下で紹介する。ここで、図 1 は Song Surfing の実際の動作画面であり、図 2 はその画面構成図である。



図 1 Song Surfing 動作画面

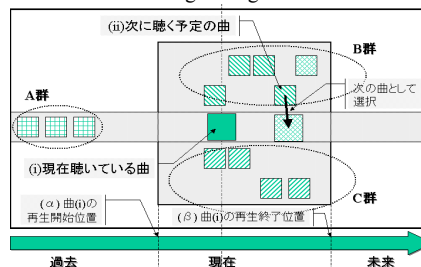


図 2 Song Surfing 画面構成図

#### 2.1 Auto Surfing 機能

Song Surfing は、現在再生中の楽曲における各フレーズの特徴量から次曲を自動的に選択する。この特徴量抽出には、萩山らの技術 [2] を用いている。Song Surfing では、画面左端が過去、右端が未来、中央が現在を表している。中央の白い帯は音楽視聴の流れを表しており、白い帯に中央にある楽曲が現在聴いている曲 (i) である。したがって、曲 (i) のジャケット画像は再生終了するまで中央から動かない。曲 A 群は今まで聴いた曲の履歴である。曲 (ii) は未来にある楽曲であり、曲 (i) とつながる予定の楽曲である。Song Surfing は曲 (ii) として、曲

<sup>†</sup> バイオニア株式会社 総合研究所  
Corporate Research and Development Laboratories, Pioneer Corporation  
<sup>††</sup> バイオニアデザイン株式会社  
Pioneer Design Corporation

A 群の中で最も曲 (i) に近いフレーズをもつ楽曲を選ぶ。曲 B, C 群のジャケット画像を囲む、大きなジャケット画像は曲 (i) のジャケット映像を拡大表示したものであり、このジャケット画像の左右両端が曲 (i) の開始点と終了点を表している。つまり、左端の位置 ( $\alpha$ ) が開始点を、右端の位置 ( $\beta$ ) が終了点を表しており、曲 (i) の再生が進むに従って位置 ( $\alpha$ ), ( $\beta$ ) と曲 (ii) と B, C 群が未来 (右) から過去 (左) へと移動する。また、曲 (ii) の位置は現在聴いている曲 (i) とスムーズにつながる地点を表しており、曲 (ii) が曲 (i) と重なる位置まで来れば、自動的につなげて再生される。

したがって Song Surfing を用いれば、ユーザはシステムに対して明示的な指示を与えなくても音楽を楽しみ続けることができる。さらに Song Surfing は類似フレーズでスムーズに曲をつなぐので、ランダム再生のようにそれまでの視聴雰囲気を変えていくことが無い。

## 2.2 類似曲提示・選択機能

ユーザはシステムの自動再生に任せるだけではなく、他の候補曲を次に聴く曲として選べる。現在再生中の曲 (i) の上部にある曲 B 群は、曲 (i) とスムーズにつながる箇所を持つ曲である。また、曲 B 群にあるそれぞれのジャケットの表示位置は、曲 (i) のどの部分でスムーズにつながるかを表している。したがってユーザが現在視聴中の曲を知っていれば、つながる箇所から対象曲をある程度類推できる。曲 B 群から曲を選べば、その曲が音楽再生ラインを表す中央の帯上に配置される。図 1, 2 では、曲 (ii) が次の曲として選ばれているが、別の曲を選べば曲 (ii) の代わりにその曲が表示される。

また、曲 B 群にある候補曲と中心の白い帯との距離にも意味がある。中心の帯と各ジャケット画像の距離が現在聴いている曲との特徴量の違いの大きさを表現している。ジャケット映像が中心の帯に近ければ、現在聴いている曲により近い特徴をもっている。したがって、現在聴いている曲が既知の曲であれば、それぞれの候補曲がどのような特徴をもつかおおよそ想像できる。また、それぞれの候補曲が既知であれば、現在聴いている曲がどのように展開していくのかおおよそ想像できる。一般に曲をそのまま特徴度で表現した場合、ユーザにとってその曲のイメージをつかむのは難しい。しかし音楽に代表される、言葉や数値による表現が難しいコンテンツであっても、具体的なコンテンツとの類似/比較を用いればユーザにとって理解しやすい。

## 2.3 雰囲気チェンジ機能

現在聴いている曲の雰囲気を大きく変更したい場合、聴いている曲から特徴度が異なる曲を選択・再生する機能である。現在再生中の曲 (i) の下部にある曲 C 群は、現在聴いている曲 (i) と特徴度が大きく違う箇所を持つ曲である。この機能がランダム再生と異なるのは、ランダム再生では必ずしも曲調の変化が保証されないのに対して、本機能はできる限り曲調の異なる曲を再生する点である。

本システムは従来のインタフェースと比較して、Auto Surfing 機能と類似曲選択/雰囲気チェンジ機能とを用いてユーザの音楽視聴に必要な労力を軽くしている。つまり、ユーザはシステムに干渉したい時にだけ干渉すればよく、現在聴いている曲に不満が無ければシステムに再生を任せればよい。

## 3. 被験者実験

数百曲以上の音楽ライブラリをもつ人達に、本システムを使用してもらった。実験用として約 300 曲と約 5000 曲からなるライブラリ環境を用意した。この場合、5000 曲の方がより違和感無く曲がつながることを実感できたようである。また、ライブラリの大きさと Song Surfing が提示する候補曲数は比例しないので、ライブラリが大きくなった場合でも同じ操作感が保たれていた。つまり従来インタフェースでは引き出せなかった、大規模音楽ライブラリが本来持っている魅力を、本システムが活かしていることを確認できた。

## 4. まとめ

フレーズ単位で曲をつなぐ音楽視聴システム Song Surfing を構築し、大規模音楽ライブラリをユーザが容易に楽しめるインタフェースを提案し、その有効性を確かめた。今後は候補曲の選択に注目して研究を進め、「より明るい曲につなぐ」「より遅い曲につなぐ」といった操作を組み込んだインタフェースを実現する予定である。また本研究は、RWC 研究用音楽データベース (クラシック音楽、ジャズ音楽、ポピュラー音楽、音楽ジャンル) を利用した。

## 参考文献

- [1] 後藤 孝行, 後藤 真孝, “Musicream: 楽曲を流し続けてくっつけて並べることのできる音楽再生インタフェース,” WISS2004, pp.53-58, 2004.
- [2] Shinichi Gayama, “Continuous Song Playback Based on Its Chord-Progression Continuity,” ICCE2006, pp.1-2, 2006.