

能動的音楽聴取サービス《Songle》による音楽可視化の試み

南部 晃史¹ 中村 滋延²

概要: 本稿は産業技術総合研究所の開発した能動的音楽聴取サービス《Songle》のAPIを用いた音楽可視化の試みを報告するものである。本アプリケーションでは《Songle》がmp3ファイルから取得するコード進行を用いて音楽構造をWebで可視化している。

An Approach to Visualizing a Music with 《Songle》

AKIFUMI NAMBU¹ SHIGENOBU NAKAMURA²

Abstract: This paper provides a report about an approach to visualizing music with 《Songle》, a Web service developed by The National Institute of Advanced Industrial Science and Technology for active listening music. This application visualizes musical structure with chord progression analyzed by 《Songle》 on web.

1. はじめに

1.1 Songle

《Songle》^{*1}とは独立行政法人産業技術総合研究所^{*2}が開発した能動的音楽聴取サービスの名称である。《Songle》ではWeb上の音楽を音楽理解技術を用いて分析され、拍情報、コード進行、楽曲構造という3つの情報がAPIによって取得できる。ここでいう「能動的」とは、元来、音楽を受動的に聴取していた聴取者が聞きたい部分を能動的に選択して聞くなど、音楽聴取における様々な働きかけを意味している。[1]

1.2 背景

西洋芸術音楽の聴取者は楽譜を見て理解しながら分析的に楽しむ音楽愛好家も少なくなく、ある種の正統な西洋芸術音楽の楽しみ方とされている節がある。

一方で、近年、音楽のダウンロード販売や動画サイトでの視聴など大量の音楽を容易に聴取することができるよう

になり、大衆音楽の愛好家が聴取する音楽の数は増えたが、音楽を深く理解して分析的に楽しむ人はほとんどいない。

1.3 目的

本稿では《Songle》によって得られた楽曲情報を視覚化し、大衆音楽の簡易な分析ができる可能性を示すことを目的とする。

2. ez-songle-visualizer.js

アプリケーション制作に当たって、SongleAPIを簡単に扱うためにez-songle-visualizer.jsを制作した。ez-songle-visualizer.jsの機能は大きくイベント処理とユーティリティに分けられる。ここでは概要のみを説明するので、詳細については[2]を参照されたい。

2.1 使用方法

使用法はjQuery^{*3}とSongleAPIを読み込んだ後にez-songle-visualizer.jsを読み込み、任意の箇所

```
Songle.Visualizer(opt);
```

を実行するだけである。ここで、引数のoptはSongleAPIにおけるイベントのコールバック関数をラップしたオブジェクトであり、SongleAPIの全てのコールバック関数を

¹ 九州大学大学院芸術工学府
Graduated School of Design, Kyushu University

² 九州大学大学院芸術工学研究院
Faculty of Design, Kyushu University

^{*1} <http://songle.jp/>

^{*2} AIST
The National Institute of Advanced Industrial Science and Technology

^{*3} <http://jquery.com/>

指定できる。例えば、再生されたときにアラートを出した
いときは、

```
Songle.Visualizer({  
  onPlay: function(api, e){  
    alert("PLAY!");  
  }  
});
```

のように書くことができる。

2.2 ユーティリティ

Songle で得られる情報の中でも特によく使うものは
SongleUtil クラスの中にまとめている。例えば、コードの
種類 (m, M7 など) を取得したい場合は次のように書く。

```
Songle.Visualizer({  
  whilePlayingChord: function(api, e){  
    var type = SongleUtil.getChordTypeDetail(e);  
    console.log(type);  
  }  
});
```

3. アプリケーション概要

一つのアプリケーション例として《VIS.KHORD》を開
発した。

3.1 開発環境および、動作環境

2章の ez-songle-visualizer.js v0.0.1a と HTML5 の can-
vas 上での描画を支援するライブラリ Processing.js
v1.4.1^{*4}を用いて、CoffeeScript で作成した。

HTML5 対応のブラウザで動作する。

《VIS.KHORD》を制作するにあたっていくつかの目標
を設定した。

- Web サービスという《Songle》の特徴を活かし Web
ブラウザ上で動くアプリケーション。
- 複数の楽曲間で比較できるような大衆音楽理論に基
いた視覚化。
- 聴取者が見ていて飽きない視覚的に美しいもの。

《VIS.KHORD》は五度圏によるコード進行の分析を理論
的背景にしている。

3.2 仕様

《VIS.KHORD》の表示画面は図1のようなものである。

3.2.1 埋め込みプレイヤー

図1の1の部分には《Songle》から提供されている外部
埋め込みプレイヤーを表示させている。

なお、《Songle》での楽曲 URL を指定し、「楽曲変更」の
ボタンを押すと、分析対象となる音楽を変更することがで
きる。

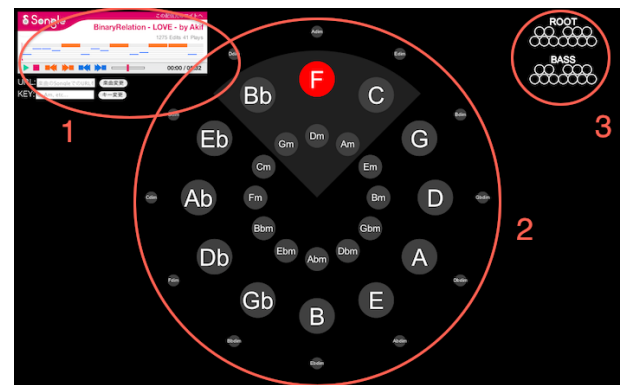


図1 《VIS.KHORD》の表示画面

また、楽曲変更時、最初はキーがCメジャーになるよう
に指定されている。キーが違うときは、任意のタイミング
でキーを指定し、キー変更ボタンを押すとキーを変更し、
後述の五度圏を回転させることができる。

3.2.2 五度圏

図1の2の部分には五度圏の円環が表示され、内側から
順にマイナーコード、メジャーコード、ディミニッシュコ
ードを表している。ディミニッシュコードは機能が似ている
長三度下のメジャーコードの外に表示した。(例: Bdim は
Gの外側。これはBdimがG9の根音省略と考えられるか
らである。)

キーを指定することで、そのトニックが中央上に表示さ
れるように回転される。このことによって複数の曲を比較
するときキーが違っていても、比較が容易になる。なお、
中央上に表示された扇形上にある和音は、三和音と考えた
とき、その構成音がキー上に存在することを示す。

コードの大きな分類を次のように与えた。

メジャーコード M, 6, 7, M7, …

マイナーコード m, m6, m7, mM7, …

ディミニッシュコード dim, dim7, m7⁻⁵, …

音楽が再生されると、今鳴っているコードを表す円が
躍動し、順々にコード進行が曲線で結ばれていく。ここ
で、直線ではなく曲線とした理由は描画することに用いた
Processing.jsにおいて、直線は二点によって決定されるが、
曲線は三点によって決定されるので、どの3つのコードが
連続するかによって線を分けて表示できるからである。

さらに、曲線の頂点をそれぞれのコードの中心からラン
ダムに微小な変化を与えている。そのことによって、楽曲
内に多く見られるコード進行は太く濃く表示されること
になる。

3.2.3 ルート音、ベース音の可視化部

大衆音楽ではコードの順次進行が散見される。しかし順
次進行、特に半音階の順次進行は五度圏上で遠いコードに
飛び回るので、目が追いつかず、その存在に気づきにくい。
そのために、図1の3の部分に五度圏の並びではなく、鍵
盤上の並びによってルート音の動きとベース音の動きを右

^{*4} <http://processingjs.org/>

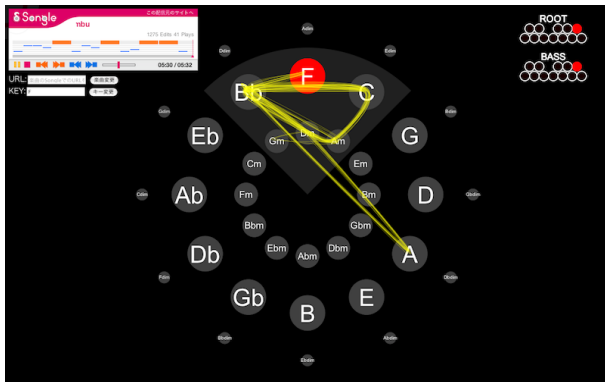


図 2 『Binary Relation -LOVE-』 (2010) の分析

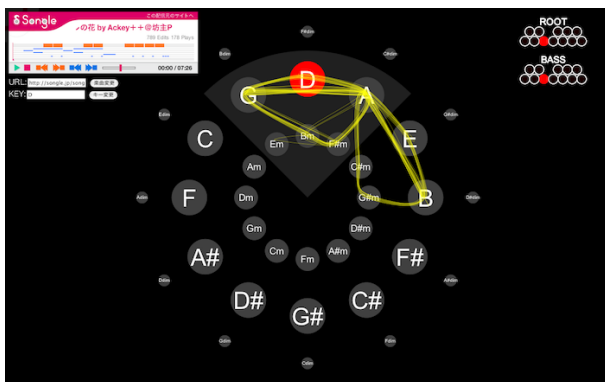


図 3 『スマイレの花』 (2008) の分析

上に可視化している。

4. 可視化例

ここで実際の楽曲を《VIS.KHORD》によって可視化した例を挙げる。可視化する楽曲は著者自身が2010年に作曲した『Binary Relation -LOVE-』*5と2006年に作曲した『スマイレの花』*6を用いた。それぞれの可視化結果は図2と図3に示した。

4.1 『Binary Relation -LOVE-』

図2を用いて『Binary Relation -LOVE-』について考える。この楽曲の可視化結果を見ると、キーがFメジャーであるにも関わらず、トニックであるFよりもサブドミナントであるB^bを中心に和声が行われていることがわかる。

また、ドミナントであるCからトニックのFに進行することはなく、Fの代理和音であるAmに進行することが多い。

4.2 『スマイレの花』

図3を用いて『スマイレの花』を考える。この楽曲の可視化結果をみると、似た形がそれぞれDとEを中心に広がっていることがわかる。これは楽曲の後半でポップスによく

見られる長二度上への転調が行われていることを意味している。

また、転調前のDを中心に広がる部分を見ると、大きく2つの輪 (G-A-D と G-A-F[#]m-Bm) が多く用いられており、西洋芸術音楽の和声理論で言うところのT-S-D-Tのカデンツが楽曲内のコード進行を特徴付けていることがわかる。

4.3 両者の比較

4.1節と4.2節で考えた両者を比較する。両者の図を見ると共に図2におけるB^b-C-Am-Dmの図形が特徴的に見られる。これはIV-V-IIIIm-VImの進行であり、JPOPによく見られる進行である。これらが、両者の楽曲では支配的であり、作曲家(著者)の癖であることがわかる。

5. まとめ

以上、《Songle》を用いることで音楽を視覚化する例を示してきた。

《VIS.KHORD》を使うことで楽曲を特徴付けているコード進行や作曲者が多用するコード進行が素人目にも把握することができ、作曲者自身が気付いていなかったことすらも可視化によって気付くことができる。また、この分析結果を知ることによって(コード進行的に)単純な音楽が好きか、複雑な音楽が好きかなどの一つの好みの目安を見つけられるかもしれない。

しかし、《VIS.KHORD》では増和音のことを考慮していない。また、減七和音に現れる異名同和音(例えば、CdimとAdim)が同じ物にもかかわらず、バラバラに表示されていて機能として分かりづらいなど問題点も多い。

今回はコード情報のみを用いたが、これはあくまで一例として作成したアプリケーションであり、《Songle》から取得できる拍情報、楽曲構造情報などを用いることによってさらに面白く楽曲を可視化できると思われる。

《Songle》から楽曲データをJSONデータとして取得することができるので、Processing*7やQuartzComposer*8、openFrameworks*9など描画に特化した言語やライブラリを用いることで、ブラウザ上では表現しきれない、美しく視覚的に訴えかけるアプリケーションを作ること可能である。

また、五度圏を3次元として可視化する理論も存在しているため、3次元で描画して、その理論に基づいて音楽を可視化するアプリケーションを作ることができるだろう。[3]

*7 Ben Fry らによって開発されたグラフィックに特化した Java ベースの言語
<http://processing.org>

*8 Mac OS 10.4 以降の Developer ツール内に標準搭載されている GPU を活用して映像を描くグラフィカル言語

*9 Zachary Lieberman らによって開発されたグラフィックに特化した C++ のライブラリ群
<http://openframeworks.cc>

*5 http://soundcloud.com/akifuminambu/binary_relation_love

*6 <http://piapro.jp/t/X6sv/20090323112232>

楽曲を音楽的な理論を背景に目で楽しむことができるので、大衆音楽の愛好家に音楽を分析的に聞こうとするきっかけを与えられるかもしれない。

2012年11月20日現在、Songle から公式の API 情報が公開されていないため、詳細な仕様が発表されれば、さらに高度な分析によってさらに面白い表現が可能になるかもしれないので、API の情報公開が待たれる。

参考文献

- [1] 独立行政法人 産業技術総合研究所, インターネット上の楽曲の中身を自動解析する音楽鑑賞システム, http://www.aist.go.jp/aist-j/press_release/pr2012/pr20120829/pr20120829.html
(2012年11月10日参照)
- [2] ez-songle-visualizer, <https://gist.github.com/3870098>
(2012年11月20日参照)
- [3] フィリップ ボール著, 夏目 大翻訳, 音楽の科学—音楽の何に魅せられるのか?, 河出書房新社 (2011)