同一楽曲に対する多数の歌唱と目標歌唱の音高推移分布および再生数の可視化

理学専攻　情報科学コース　2040643　近藤 芽衣　(指導教員：伊藤 貴之)

1　はじめに

2 次創作やソーシャルメディア環境の普及にともない，自らの歌唱を録音・録画して動画共有サービスへ投稿する機会が増えた．その結果，同一楽曲を様々な歌唱者が歌った音源を，人々が鑑賞して楽しめるようになった現状がある．このような歌唱群に対し，個々の歌唱者の癖や個性を理解するための一手段として我々は，それらの歌唱音響データからそれぞれの音高の推移を抽出し，その分布を可視化する手法を開発している．そのような手法の一つとして，伊藤らが提案した SingDistVis [1] は，音高および時刻を 2 軸とするヒストグラムによる可視化と，その局所部分をズームアップした折れ線表示での可視化により，音高の特徴的な分布の発見を支援する．折れ線表示には，サンプリングで同時に表示する本数を制御することで Visual Cluttering を防いでいる．本論文では SingDistVis の応用として，ソーシャルメディア上の再生数で各歌唱を色分けし，音高分布との関係を可視化した事例を報告する．

2　提案手法

本章では提案手法の各処理を手順に沿って示す．

2.1　音楽音響信号からの歌声の音高推定

多様な音源を可視化対象とするために，伴奏がミックスされた歌唱音源（混合音）を扱う．まず混合音から歌声のみを分離し，その分離された歌唱音源から音高を推定する 2 段階の処理を行うが，それぞれ以下の手法を用いた．

2.1.1　時刻及びキーオフセット推定

伴奏付き歌唱音源と伴奏音源を Constant–Q 変換によりスペクトログラムに変換する．この 2 音源の時間と周波数の二次元配列の相互相関の最大値を求めることにより，音源の始まる時刻やキーのずれを検出する．この時，相互相関を求める範囲には歌唱が含まれていないことが望ましい．これによりサビ始まりの曲以外は音源の始まりから 5 秒程度を用いる．

2.1.2　Spleeter による歌声分離

U-Net 構造を持つ深層学習ベースの音源分離手法である Spleeter [2] を用いて，混合音から歌声を分離する．入力音響データはステレオ MP3 形式，サンプリングレートは 44100Hz とする．

2.1.3　基本周波数 (F0) 推定

Spleeter により分離した歌声から，音高として基本周波数（F0）を推定する．混合音から伴奏音を完全に除いて歌声分離するのは難しいため，F0 推定手法は耐雑音性に優れたものが望ましい．そこで本手法では PYIN[3] を用いた．

2.2　SingDistVis による音高可視化

音高の可視化について，SingDistVis の処理手順を概説する．詳細は文献 [1] を参考にされたい．

2.2.1　音高データの表記

本章では歌唱者集合 S を構成する各歌唱者の音高の推移を以下のように表記する．

$$S = \{s_1, s_2, ..., s_I\}$$
$$s_i = \{e_i, p_{i1}, p_{i2}, ...p_{iJ}\} \quad (1)$$

ここで s_i は i 番目の歌唱者による歌唱の音高系列，I は歌唱者の総数,，e_i は i 番目の歌唱者の歌唱動画の再生数に応じた評価係数である．p_{ij} は i 番目の歌唱者の j 番目の時刻における F0 値の対数，J は基本周波数推定の対象区間における標本化された時刻の総数（各音高系列の F0 値の個数）である．なお休符に相当する無音部分には，便宜上，F0 値の対数にゼロを代入した．現状の実装では評価係数 e_i は 4 段階となっており，最も再生数が低い歌唱群 $e_i = 1$ を，最も再生数が高い歌唱群には $e_i = 4$ を付与する．なお原曲にあたる歌唱には，他の歌唱と区別するために $e_i = 5$ を付与する．

2.2.2　ヒストグラム画像の生成

本手法では，時刻を横軸，音高を縦軸とした長方形領域を設定し，これを格子状に分割する．p_{ij} の各々が上述の格子構造のいずれの長方形領域に該当するかを算出し，各格子領域を通過した歌唱数から各格子領域の濃淡色を算出することで，グレースケールのヒストグラム画像を生成する．

2.3　SingDistVis の拡張

SingDistVis の GUI におけるヒストグラム画像内において，ユーザが指定した矩形領域に対応する音高推移を，折れ線の集合で表現する．この際 Visual Cluttering を防ぐために，同時に描画する折れ線の本数をサンプリングにより制御する．これは，各々の折れ線 p_i に対して，ユーザ指定のタイミングで再計算できる一様乱数 $z_i(0.0 \leq z \leq 1.0)$ を割り当て，以下を満たす場合のみ描画する．

$$\beta_{e_i} z_i > Z_{thres} \quad (2)$$

ここで，β_{e_i} は歌唱評価 e_i に応じた係数，Z_{thres} は GUI スライダーで調整する閾値であり，いずれもユーザが

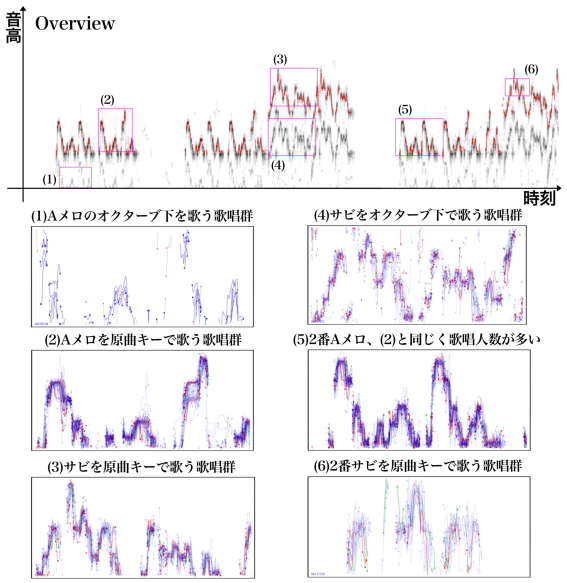

図 1: グレースケール画像と 6 箇所のズームアップ実行例

調節可能なパラメータである．例えば高（低）評価な p_i のみを表示したい場合は，その β_{e_i} が大きな値を取るように設定する．

3　実行例

本手法による可視化の例を紹介する．プログラミング環境は Java 1.12.0 および JOGL (Java binding for OpenGL) 2.3.2 を用いた．実行例には【初音ミク】夜明けと蛍【オリジナル】[1]の 67 人の歌唱を用い，再生数はニコニコ動画における再生数を採用した．本報告では音響データから Spleeter [2] の 2stem 版モデルを用いて音源を分離し，PYIN [3] を用いて推定した F0 を入力とした．可視化結果の画素数は N=1000,M=480 とし，対象となる周波数を 110Hz〜1760Hz（オクターブ表記付き音名音階で A1 から A5）の 4 オクターブとした．

図 1 は音高推移分布をグレースケール画像として表示した例とその中で 6 箇所をズームアップし表示した例である．グレースケールで黒に近い箇所では，同じような音高推移の歌唱が多いことを意味する．例えば図中の (2)(4) では音高推移が 2 つに分かれていることがわかるが，各遷移が同じ濃さであることから，サビに入ってから音高を 1 オクターブ低く歌唱した人も多かったことがわかる．

図 2 は緑色の音高が原曲の音高，赤色の折れ線が再生数の高い歌唱，青色の折れ線が再生数の低い歌唱を示している．円内には縦に幅のある音高推移の集合を確認できる．さらに本数を制御してみるとこの集合を囲うように高音を歌っている再生数の高い歌唱と，低音を歌っている再生数の高い歌唱があることがわかる．また緑線に注目することで，原曲の音高は高音側の音高推移であるとわかる．この箇所では低音側の音高推移が歌唱者によるアレンジであり，一定数の歌唱者がこのアレンジを採用していることがわかる．

[1]https://www.nicovideo.jp/watch/sm24892241

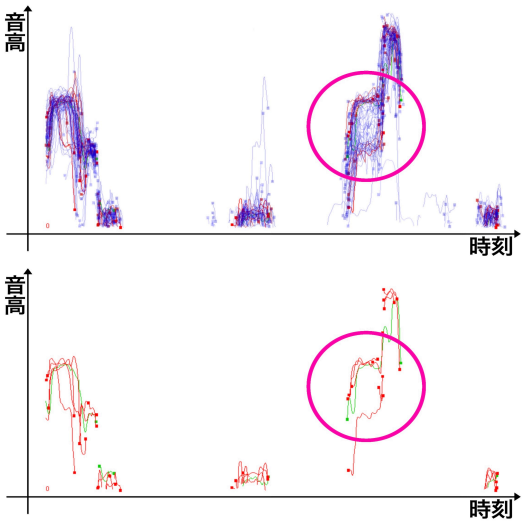

図 2: 本数制御した折れ線集合（A メロ) 円内で縦に幅のある音高推移の歌唱群が確認できる．

4　まとめ

本論文では，多数の歌唱者による同一楽曲に対する伴奏付きの歌唱音源の音高推移分布と，それぞれの再生数を可視化した例を示した．多くの歌唱が原曲とは異なる音高推移を描いているケースや，サビでオクターブ下げて歌う歌唱者が複数見られるケースのほか，再生数の高い歌声と低い歌声では，ビブラートの有無など音高の変化に違いが見られた．今後の展望として，ビブラートなどのボーカルテクニックを特徴量として抽出し，可視化画面への表示を加えることで，より詳細な分析を可能としたい．また異なるキーで歌唱された同一楽曲について，キーを合わせて可視化する機能を追加予定である．また GUI 機能の拡充として，可視化画面からの音源再生機能や，ユーザ自身の歌唱を区別して表示する機能を追加することで，好きな歌唱を選んでそれを目標として歌唱を練習するための支援ツールを開発したい．

謝辞：本研究にあたり，産業技術総合研究所中野 倫靖氏，深山 覚氏，濱崎 雅弘氏，後藤 真孝氏にはデータ提供および多大な助言を賜りました．ここに感謝申し上げます．

参考文献

[1] 伊藤 貴之, 中野 倫靖, 深山 覚, 濱崎 雅弘, 後藤 真孝, SingDistVis: 多数の歌声から歌い方の傾向を可視化できるインタフェース，ソフトウェア科学会 WISS 2021 論文集, 94, 1–8, 2021.

[2] R. Hennequin, A. Khlif, F. Voituret, M. Moussallam, Spleeter: A Fast and Efficient Music Source Separation Tool With Pre-trained Models, Journal of Open Source Software, 5(50):2154, 2020. doi: 10.21105/joss.02154.

[3] M. Mauch, S. Dixon, PYIN: A fundamental fre-quency estimator using probabilistic threshold distribu-tions, Proc. ICASSP2014, 659–663 , 2014.