DEIM Forum 2022 F23-3

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

近藤　芽衣† 伊藤　貴之† 中野　倫靖†† 深山　覚†† 濱崎　雅弘††
後藤　真孝††

† お茶の水女子大学　〒 112–0012 東京都文京区大塚 2-1-1
†† 産業技術総合研究所　〒 305–8568 茨城県つくば市梅園 1-1-1
E-mail: †{g1620517,itot}@is.ocha.ac.jp, ††{t.nakano,s.fukayama,masahiro.hamasaki,m.goto}@aist.go.jp

あらまし　2 次創作やソーシャルメディア環境の普及にともない，同一楽曲に対する多数の歌唱（同曲異唱）を人々が楽しめるようになった．そうした同一楽曲に対する多数の歌唱群の音響データを分析して比較することで，同一楽曲に対する各歌唱者の癖や個性の違いを理解することが可能になる．そのような歌唱理解を支援する一手段として我々は，同一楽曲に対する多数の歌唱群の音響データから歌唱されている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研究している．本手法では，音高および時刻を 2 軸とする 2 次元ヒストグラム画像による可視化と，その局所部分をズームアップして折れ線表示する可視化の組み合わせにより，音高の特徴的な分布を強調表示する．本報告ではその応用として，ソーシャルメディア上の再生数で音高推移を色分け表示するとともに，原曲の音高を目標歌唱として比較表示することで，原曲と比べた音高の違いと再生数の傾向を見比べた． 本報告では 67 人の歌唱者による同一楽曲の音高推移分布と，それぞれの再生数を可視化した例を示す．

キーワード　情報可視化，音高，歌声情報処理，基本周波数，再生数，インタフェース

1 はじめに

多数の歌唱者が同一の楽曲を歌った同曲異唱コンテンツを公開する機会が近年増えている．このような多数の歌唱データを分析することは，歌唱の癖や楽曲の傾向の分析など学術的な観点からも有用であるだけでなく，歌唱力向上や流行分析などの実用の可能性も高い．例えば，どのような癖を有する歌唱者が人気を博する傾向にあるか，原曲の音高に対しどのようにバリエーションのある表現をする歌唱者がいるか，等を分析できる．また，人気のある歌声とその他の歌唱者の歌唱を比較することで，人気のある歌唱者のどのような歌唱の模倣が容易か困難か，といった議論も可能になる．このような歌唱データ分析において，本報告では一つの楽曲に対しどのようなバリエーションを持った歌唱をする歌唱者がいるのかを分析した結果を報告する．

歌唱の音高推移分布は，音高の時系列データとみなすことができる．時系列データの分類や特徴検出は従来から多くの研究がなされており [1], [2]，それらを適用することで歌唱の癖の違いや音高の逸脱を検出することが可能である．一方で，歌唱の分析においては音高の逸脱を発見するだけでなく，その意味付けが重要である場合もある．例えば同一楽曲の特定の瞬間の音高に個人差が見られた際に，意図的な歌唱技法として音高をずらしているのか，技量不足や練習不足により意図せずに音高がずれているのか，といった点を解釈する必要がある．このような解釈をするには分析者が歌唱への理解と経験を持ち合わせている必要があり，従来手法の適用のみでは実現できない．

そこで我々は，主観的・定性的に歌唱データを観察するための可視化手法の開発に取り組んでいる．可視化は大規模データを理解するために多用されており，歌唱の大規模データを理解することで，多数の歌唱者の歌い方のバリエーションを分析するという本研究の目的にも合致する．その一環で伊藤らは，多数の歌唱者の音高推移を可視化する SingDistVis [3] を提案した．SingDistVis では 2 種類の可視化手法を組み合わせることで，曲全体にわたる音高推移の分布を俯瞰しつつ，各歌唱や楽曲の一部分に注目することができる．1 つ目の可視化手法では，音高を一定の時刻ごとに推定することで得られる音高の列を時系列情報とみなし，横軸を時刻・縦軸を音高とする格子の上にプロットする．その格子上の各長方形領域におけるプロット回数を集計することで 2 次元ヒストグラムを構成し，これをグレースケールの画像として表示する． 2 つ目の可視化手法は局所的な音高遷移の様子を折れ線の集合で可視化する．

本報告では SingDistVis の実装を応用して，ソーシャルメディア上の再生数で各歌唱の音高推移を色分け表示するとともに，原曲の音高を目標歌唱として比較表示することで，原曲と比べた音高の違いと再生数の傾向について考察した結果を報告する．次章以降，関連研究を述べた後で，パラメータによって見え方を調整できる我々の可視化手法を説明する．また，実応用においては歌唱を含む音楽音響信号を扱う必要があるため，実例を用いて歌声データを分離する手法とその可視化結果を示す．

2 関連研究

2.1 多数の歌唱データの活用

同一の歌唱技法を異なる歌手が歌ったデータがあれば，歌唱技法の傾向を分析することが可能になる．この点に着目した研究の例として Wilkins ら [5] は，20 人のプロ歌唱者による 10 時間以上の歌唱録音データベースを構築し，ビブラートやトリルといった歌唱技法を分析した結果を示している．

また歌唱作品の制作環境の一例として都築ら [6], [7] は，同一楽曲に対する複数歌唱を組み合わせて合唱作品を制作する過程を支援するツールを提案している．

2.2 音高データの分析

歌唱の癖や違いを測る基本的な指標の一つに音高（F0）があげられる．歌唱データの各時刻における音高を推定し，楽譜から得られる音高と比較することで，音高をあえて外す意図的な歌唱，あるいは音高が大きく逸脱している歌唱などを発見できる [8]．あるいはビブラート等の歌唱テクニックを音高の推移から検出したり [9], [10]，オーバーシュートなどの動的変動 [11] に着目するなどにより，歌唱の個性を分析できる．山本らは，同一歌唱を 2 名の歌手が模倣して歌ったデータを用い，ビブラートを含む 13 種類の歌唱テクニックを対象に，その頻度・特徴・出現箇所を分析した [14].

2.3 音高推移の可視化

歌唱データおよびそれに限定しない演奏データにおける音高推移の分析や観察に可視化を用いた研究事例は，既にいくつか報告されている [12]．また，Nakano ら [13] の MiruSinger，Shiraishi ら [15] の HAMOKARA, Moschos ら [16] の FONASKEIN, Mayor ら [17] の歌唱採点手法には，歌唱の練習成果を正解楽譜（ピアノロールなど）と比較可視化する機能がある．複数の歌唱を対象とした例として，都築らは 4,524 人の F0 分布を局所的にヒートマップで可視化する機能を提案した [7]．また著者らは，2,024 人の F0 に対し，ヒートマップを用いた大域的な可視化を提案している [19].

演奏情報の中から F0 値およびその時間推移の適切な同定を支援するための可視化 [20], [21] や，周波数情報から推測される調性の可視化 [22] などの事例がある．また，歌唱の音高推移から歌唱スタイルを理解することを目的として，音高とダイナミクスを 2 軸とした可視化 [23] や，音高と音高差分を 2 軸とした可視化 [24] が報告されている．Wilkins [5] らによる歌唱技法の分析結果はスペクトログラムとして可視化されている．

2.4 時系列データの可視化

歌唱の音高の推移は時系列データであり，汎用的な時系列データ可視化手法を適用することが可能である．

ここで n 個の標本がそれぞれが m 個の時刻における実数値を有する時系列データがあるとする．このようなデータに関する多くの可視化手法は以下のいずれかのアプローチを有する．なお，以下での「実数値」は音高に，「密度」は近い音高を有する歌唱者の人数に対応する．

図 1　DAMP-balanced dataset [2] に収録された”Let It Go”の 2024 人の歌唱データを折れ線グラフで表示した例．Visual Cluttering と呼ばれる画面上の折れ線の過密状態を回避することが難しい．

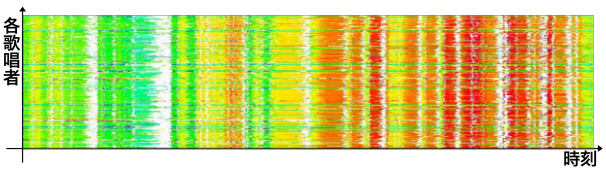

図 2　図 1 と同一のデータを実数値のヒートマップで表示した例．音高推移の個人差を読み取ることが困難である．

（1） 一方の座標軸に m 個の時刻，他方の座標軸に実数値を割り当てた折れ線グラフ [25], [26] や散布図.

（2） (1) の折れ線や点群を密度関数に置き換えて，密度を各画素の明度や色相に変換したヒートマップで表現したもの [27].

（3） 一方の座標軸に m 個の時刻，他方の座標軸に n 個の標本を割り当てたマトリクスに対して，実数値を各画素の明度や色相に変換したヒートマップで表現したもの [28], [29].

これらのアプローチの各々にはいくつかの問題点がある．(1) に示した折れ線グラフや散布図では，画面上の描画物の過密状態が引き起こす Visual Cluttering と呼ばれる視認性の低下が避けられない．また可視化結果からのデータ読み取りにおいて色の識別能力は高くない [30] ことが知られており，(3) に示したヒートマップでは実数値を正確に読み取れないという問題がある．実際に多数の歌唱データを有する DAMP-balanced dataset を (1) および (3) の手法を用いて表示したものが図 1 と図 2 である．図 1 では横軸に時刻，縦軸に音高を割り当てている．画面上の折れ線の過密状態によって Visual Cluttering が生じていることがわかる．また描画処理時間が折れ線の数に依存し，得られる画面解像度と不釣り合いに計算時間を要するという問題もある．図 2 では，横軸に時刻，縦軸に歌唱者をならべ，画素値の色相で音高を表現している．この表現では音高推移の個人差を読み取ることが困難であることがわかる．

本研究の目的の一つとして「歌唱における音高推移のパターンを発見する」という点がある．時系列データの可視化では，クラスタリングや部分頻出パターン検出などの汎用的な手法を用いている事例がいくつかある [31]．ここで音高推移のパターンは部分的に「同じ箇所に該当するパターンの時間長が異なりうる」という特徴があり，この点に着目した時系列データ可視化手法はまだ多くない．一方で，任意のタイミングで統合や分離を繰り返し，異なる長さのパターンの集合が同時に生じるような時系列データにおいては，Sankey Graph や Storyline [32] の

適用が有効な場合が多い．しかしこれらの表現では，パターンの集合間の標本の入れ替えが頻繁に発生するようなデータにおいて絡まったような複雑な表示になるため，視認性の低下が避けられない．本研究が採用する可視化手法は同一の音高推移に対して (1) と (2) の二つの異なる可視化手法を適用して表現した点に特徴がある．

3　再生数を考慮した同曲異唱の可視化

本章では, 同曲異唱の音高（F0 値）を可視化する SingDistVis [3] とその拡張について，処理手順に沿って説明する．

3.1　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．

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

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 番目の歌唱者の時刻 j における F0 値の対数，J は F0 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，最も再生数の低い歌唱群に $e_i = 1$ を，最も再生数の高い歌唱群には $e_i = 4$ というように再生数の低い順に評価係数を付与している．また，原曲にあたる歌唱には，他の歌唱との区別のために $e_i = 5$ を付与する．

楽曲により評価係数を定める再生数の閾値を変えることができ，本論文の実行例では 5000 回再生以上の歌唱に e_4 を，1000 回再生以上の歌唱に e_3 を，100 回再生以上の歌唱に e_2 を，100 回再生未満の歌唱に e_1 を付与している．

なお本研究では，全ての歌唱が同じ長さ・同じタイミングになるようデータを揃えた上で，同一の時刻における音高を推定することを前提としている．

3.2　ヒストグラム画像の生成

本手法では，時刻を横軸，周波数の対数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．歌唱の開始時刻および終了時刻をそれぞれ $t_{\text{start}}, t_{\text{end}}$ として長方形領域 R の左右端にわりあて，この区間を N 個に分割する．また可視化の対象となる周波数領域の上限と下限を設定し，各々の対数をそれぞれ $p_{\max}, p_{\min}$ として R の上下端にわりあて，これを M 個に分割する．なお，以下の記述では $t_{\text{start}} = t_1, t_{\text{end}} = t_N, p_{\min} = f_1, p_{\max} = f_M$ とする．

続いて本手法では，式 1 に示す 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する．具体的には，左から u 番目，下から v 番目の長方形領域について，

$$t_u < j < t_{u+1}$$
$$f_v < p_{ij} < f_{v+1} \quad (2)$$

が成立するようであれば，p_{ij} は当該長方形領域に該当するとして，変数 r_{uv} に 1 を加算する．

以上の処理による集計結果は 2 次元ヒストグラムを構成するが，本手法ではこれを横 N 画素，縦 M 画素の画像として扱う．長方形領域に包括される p_{ij} の個数を集計した変数 r_{uv} から，以下の式

$$I_{uv} = 1.0 - (\alpha r_{uv})^{\gamma} \quad (3)$$

によって，左から u 画素目，下から v 画素目の明度 I_{uv} を求める．ここでの α および γ はユーザが調整可能であり，全体的な明るさと強調度合いを変更できる．

3.3　SingDistVis の拡張

3.3.1　折れ線集合による可視化

SingDistVis ではグラフィックユーザインタフェース中に表示されるヒストグラム画像において，ユーザが指定したピンク色の枠で示される矩形領域内部に対応する音高推移を折れ線の集合で表現する．本研究ではこの折れ線の集合による可視化を目標音高を表示する目的で拡張した．

再生数の異なる歌唱と原曲の音高の違いを分析できるように色の割り当てを行なった．各折れ線の色は評価係数 e_i により色分けされている．原曲の音高を意味する $e_i = 5$ には緑色，再生数の高い歌唱を意味する $e_i = 4$ には赤色，再生数の低い歌唱を意味する $e_i = 1$ には青色を割り当てている．

SingDistVis には以下の 3 つの描画モードが搭載されており [4]，本実装ではより再生数の高い歌唱群を見やすくするため透明度の値を変更している．各モードの比較を図 3 に示す．

（1） both：再生数が最も高い歌唱群と最も低い歌唱群の色を濃く表示し，間に属する歌唱群の色を薄く表示するモード

（2） higher：再生数が高い歌唱群の色を濃く表示し，低い音源を薄く表示するモード

（3） lower：再生数が低い歌唱群の色を濃く表示し，高い音源を薄く表示するモード

例えば本論文で可視化に用いたデータは再生数が低い歌唱の方が多く，再生数の高い歌唱の音高推移は埋もれやすいため，higher モードを用いることで再生数の高い歌唱群を強調表示することができる．

表 1 は現在の実装における評価係数 e_i と折れ線の色・各モードの不透明度の対応例である．どのモードにおいても $e_i = 5$ の折れ線の不透明度は変わらない．

表 1　評価係数 e_i と折れ線の色・不透明度の対応の例

評価係数	R	G	B	α_{higher}	α_{lower}	α_{both}
1	0.0	0.0	1.0	0.2	1.0	1.0
2	0.1	0.1	0.8	0.3	0.7	0.5
3	0.8	0.1	0.1	0.7	0.3	0.5
4	1.0	0.0	0.0	1.0	0.2	1.0
5	0.0	0.8	0.0	1.0	1.0	1.0

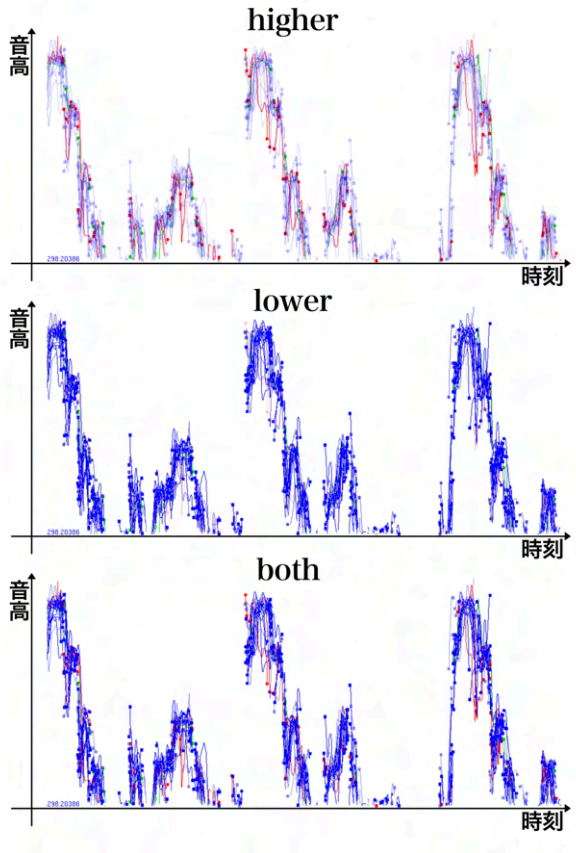

図 3　折れ線集合の描画モード比較

3.3.2　サンプリングによる折れ線の本数制御

折れ線集合による可視化において，SingDistVis では 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していた．

本研究では，各歌唱の再生数を考慮して折れ線本数を制御する．具体的には，歌唱 s_i の折れ線各々に対して，ユーザ指定のタイミングで再計算できる一様乱数 $z_i(0.0 \leqq z \leqq 1.0)$ を割り当て，以下を満たす場合のみ描画する．

$$\beta_{e_i} z_i > Z_{\text{thres}} \tag{4}$$

ここで，β_{e_i} は歌唱評価 e_i に応じた係数，Z_{thres} はグラフィカルユーザインタフェース中のスライダーで調整する閾値であり，いずれもユーザが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．

図 4 は both モードによる描画，$\beta_{e_i=1} = 0.8$，$\beta_{e_i=2} = 0.5$，$\beta_{e_i=3} = 0.2$，$\beta_{e_i=4} = 2.0$ としており，表示本数を減らしても再生数の高い音源が画面から消えにくくなるようになっている．なお，$e_i = 5$ の音源についてはいかなる Z_{thres} においても表示されるように設定している．

3.3.3　原曲の音高表示

評価係数によって原曲の音高を異なった色で表示することで，折れ線集合の画面には常に原曲の音高が表示されるようになっている．またヒストグラム画像にも音高推移が重畳表示されるようになっており，その初期値は原曲の音高に設定されている．ヒストグラム中に重畳される音高推移は折れ線集合の中からユーザが選択することができる．

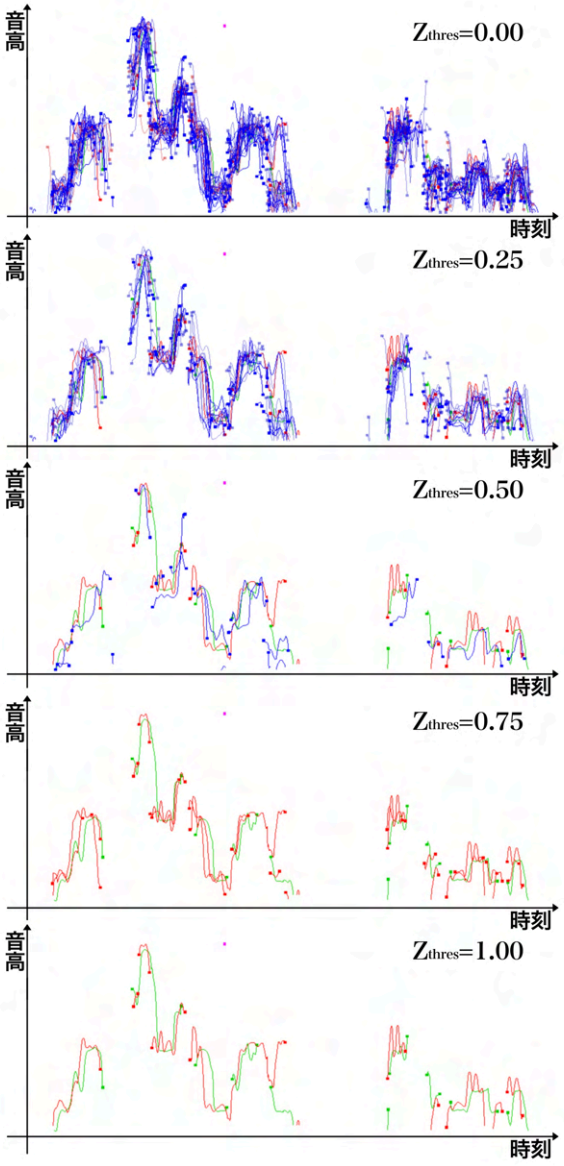

図 4　Z_{thres} の調節による折れ線の本数の変化．

4　音楽音響信号からの歌声の音高推定

伊藤らの先行研究 [3] で示した実行例では，多数の歌唱者によるデータの例として，最初から無伴奏歌唱音源として公開されているオープンデータを用いた．しかし，多様な楽曲について分析を行いたい場合このようなオープンデータを幅広く入手できるとは限らない．伴奏とあわせて録音された一般的な歌唱音源から無伴奏データを分離（歌声分離）して用いることで，幅広い楽曲に対して可視化が可能になる．そこで，本研究では伴奏付き歌唱音源から歌声を分離してその F0 を推定する．

以下，時刻及び歌唱キーのオフセット推定，Spleeter [36] による歌声分離，歌唱音源からの F0 推定，の順に説明する．

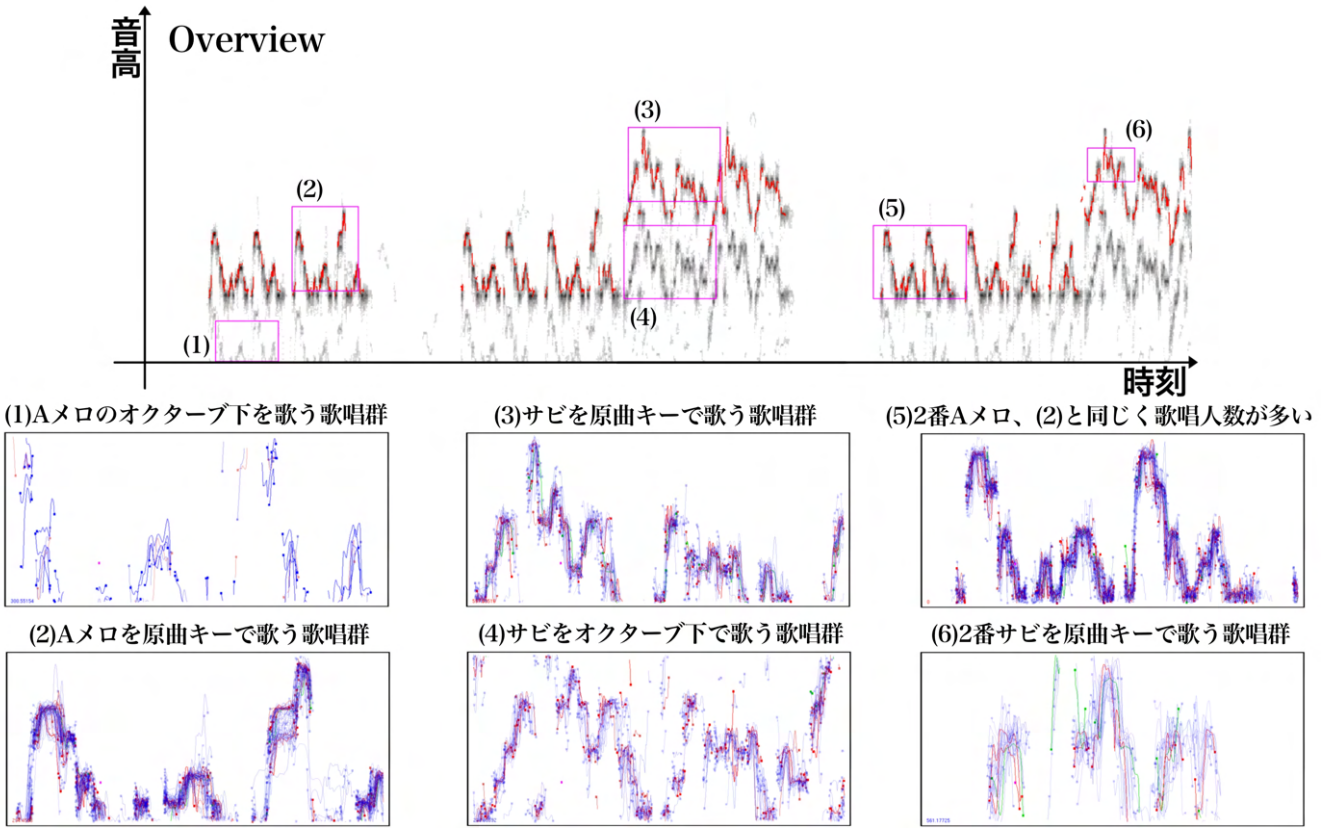

図 5　音高推移分布をヒストグラム画像として表示したものと，6 箇所をズームアップし折れ線集合で表示した例.

4.1　時刻及びキーのオフセット推定

各々の歌唱者が独自に公開している伴奏つき歌唱音源は，オリジナルの伴奏音源から時刻がずれていたり，歌唱のキーが異なる場合があるため，それらの違いをオフセットとして自動推定して調整する．まず，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 音源のスペクトログラムを構成する時間・周波数軸からなる二次元配列の相互相関関数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．ここで，この方法は 2 音源間において音源の始まる時刻とキーのズレ以外はほぼ共通している場合に有効であるため，相互相関係数を求める範囲には歌唱が含まれていないことが望ましく，多くの楽曲において前奏となる「音源開始から 10 秒程度」を用いる．ただし，歌唱から始まる楽曲においては，音源開始以外の箇所から 10 秒程度を用いる．ここで，キーが異なる音源については別のファイルに書き出す．現在の実装では原曲とキーが同じ歌唱のみ可視化対象としている．

4.2　Spleeter による抽出

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 [36] を用いて，歌唱と伴奏の混合音から歌声を分離する．入力音響データはサンプリングレート 44.1kHz のステレオの音源が MP3 形式で保存されたものとした.

4.3　F0　推　定

歌唱ありの音源の音響信号から伴奏音源の音響信号を減算する処理による歌唱音源，または Spleeter を用いて抽出した歌唱音源に対して，歌唱されている音高を推定する．これらの手法で得られる無伴奏歌唱音源には抑制しきれない雑音が残ることがあるため，耐雑音性に優れた F0 推定手法を適用することが望ましい．本報告では PYIN [37] を用いた.

5　実　行　例

本手法による可視化の例を紹介する．可視化ソフトウェアの開発には 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】（https://www.nicovideo.jp/watch/sm24892241）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．またこの楽曲には B メロがない．本報告では音響データから Spleeter [36] の 2stem モデルを用いて音源を分離した後，PYIN [37] を用いて推定した F0 を用いて可視化を行なった．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 から 1760Hz（オクターブ表記付き音名で A1 から A5）の 4 オクターブとした.

図 5 は音高推移分布をヒストグラム画像として表示した例とその中で 6 箇所を both モードで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．ズームアップし表示した例は全

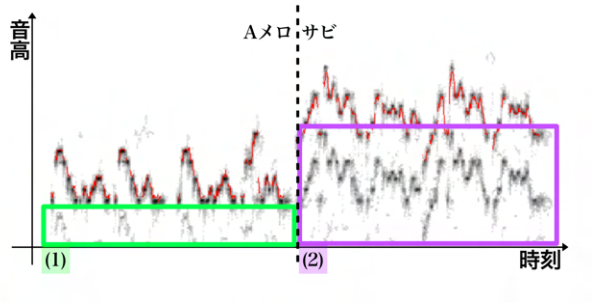

図 6　音高推移分布をヒストグラム画像として表示し原曲の音高を赤線で表示した例.

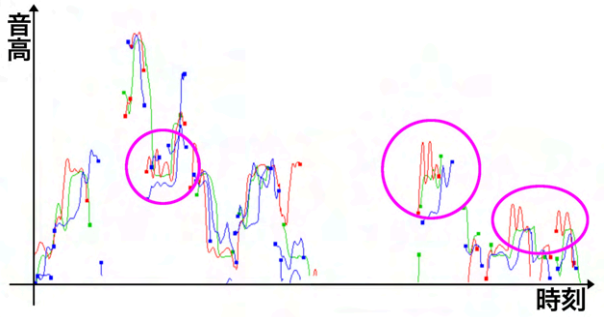

図 7　本数制御した折れ線集合（サビ冒頭）
円内でビブラートとみられる音高の揺れが確認できる.

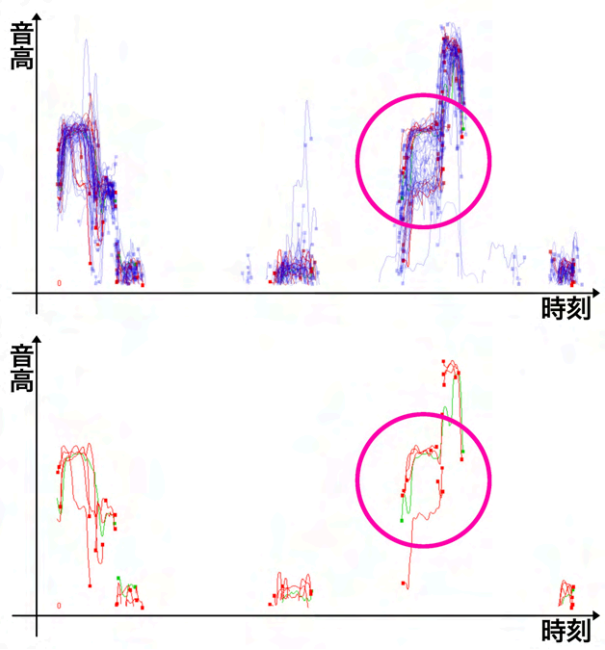

図 8　本数制御した折れ線集合（A メロ）
円内で縦に幅のある音高推移の歌唱群が確認できる.

て $Z_{thres} = 0.0$ で表示している.

図 5 の各ズームアップ例に注目すると，(1) のような歌唱人数の少ない領域と，(2)・(5) などの歌唱人数の多い領域があることがわかる．このように選択箇所に応じて折れ線の本数が大きく異なるためユーザによる Z_{thres} の調節が必要である．また矩形領域の大きさや形に関わらずズームアップ画像は同じ比率で表示されるため，(1)・(6) のように小さな矩形領域を選択した場合は折れ線の密度が低くなる.

図 6 は図 5 のヒストグラム画像の中からワンコーラスにあたる部分を抜粋したものであり，赤線は原曲の音高推移を表している．(1),(2) の枠内に注目すると原曲の音高に沿った折れ線の一群（クラスタ）の下に同様の推移をしているクラスタが見られる．赤線に注目すると楽曲のサビにあたる箇所で，原曲の音高周辺の他に原曲より低い音高の箇所にも歌唱が集中していることがわかる．濃さにも注目すると (2) では上下のクラスタは同じ濃さであることから，半分ほどの歌唱者がサビで 1 オクターブ低く歌唱していたことがわかる．それに比べて (1) では上下のクラスタに濃さの違いがあり，(1) 内部のクラスタは上に比べて薄く表示されているため，この領域で 1 オクターブ低く歌っている歌唱者は少ないことがわかる．この楽曲の A メロを 1 オクターブ下げた音域は非常に低いことからこの音域を選択している歌唱者が少なかったと考えられる．またサビ部分にあたる 1 オクターブ低くした音高推移は，A メロにあたる箇所の音高推移とほぼ同じ音域に属していることから，この楽曲はサビを 1 オクターブ下げても違和感は少なく，この音域で歌う歌唱者がいることは十分に考えられる．このように本手法を用いることによって，原曲とは異なっていながら違和感のない歌い方を発見することができる.

図 7，8 では緑色の折れ線が原曲を，暗い赤色の折れ線が再生数の高い歌唱，青い折れ線が再生数の低い歌唱を示している．また図 7 は both モードによる出力である．円内を見てみると，特に赤色の再生数の高い歌唱についてビブラート（周期的に音高を変動させる歌唱テクニック）とみられる音高の揺れを確認できる.

この箇所はこの楽曲のサビの最高音の後，音域を下げて発音される部分であり，歌い手にとって難所と考えられる．実際に歌ってみると音をあわせるだけでも難しく，特に再生数の低い歌唱群ではこの箇所で音を下げ過ぎてしまったり，下がりきれていないものが多く見られる．このように歌唱テクニックを要する箇所で多くの人がどのように歌う傾向にあるかを分析することができる.

図 8 は上の図が $Z_{thres} = 0.0$，higher モードでの可視化結果，下の図が同じ箇所の $Z_{thres} = 0.75$，higher モードでの可視化結果である．円内に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.

6　今後の展望

本研究の今後の研究課題として，以下を検討している.

6.1 付随情報の同時表示

歌唱者の年齢や歌唱経験年数，ウェブ上での歌唱の評価や再生数といった属性を付随情報として各歌唱に付与し，これと音高推移の関係を観察したい．

また歌唱技法と音高推移の関係を可視化することも考えられる．音高推移から評価の数値化がある程度可能な歌唱技法の例として，ビブラートやオーバーシュート等が挙げられる [14]．周波数の時間推移を保持する変数とは別に，歌唱技法の使用の有無を示す変数を設け，これを時系列データとして歌唱群を可視化することで音高推移が歌唱技法の模範的な歌い方通りであるかを分析できると考えられる．

6.2 制作者支援ツール

歌唱者が自分の歌唱技術や表現力を向上させるための支援ツールや歌声合成技術を用いる制作者が他者の制作技術を参考にするための支援ツールとして，本手法がどのように貢献できるかを実証したい．

7 ま と め

我々は，同一楽曲に対する多数の歌唱データを可視化する研究の一環として，67 人の歌唱者による同一楽曲の音高推移分布とそれぞれの再生数を可視化した．この可視化に際して，機械学習を用いた歌声分離ツール Spleeter を用いて歌声のみを抽出し，PYIN によって基本周波数を推定することで音高推移分布を得た．この可視化結果から，多くの歌唱者が原曲とは異なるが別の音高推移で歌唱している部分や，サビでオクターブ下げて歌う歌唱者が複数見られる部分が見られたほか，再生数の高い歌声と低い歌声で異なった音高推移が見られた．

文　献

[1] M. Last, A. Kandel, H. Bunke, Data Mining in Time Series Databases, World Science Publishing, ISBN-981-238-290-9, 2004.

[2] T. W. Liao, Clustering of Time Series Data — A Survey, Pattern Recognition, 38, pp.1857-–1874, 2005.

[3] 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, pp.1–8, 2021.

[4] 伊藤 貴之, 中野 倫靖, 深山 覚, 濱崎 雅弘, 後藤 真孝, 同一楽曲に対する多数の歌唱の基本周波数推定値分布の可視化, 情報処理学会研究報告音楽情報科学 (MUS), 2019-MUS-123, 47, pp.1–6, 2019.

[5] J. Wilkins, P. Seetharaman, A. Wahl, B. Pardo, Vocalset: A Singing Voice Dataset, Proc. ISMIR 2018, 2018.

[6] K. Tsuzuki, T. Nakano, M. Goto, T. Yamada, S. Makino, Unisoner: An Interactive Interface for Derivative Chorus Creation from Various Singing Voices on the Web, Proc. Joint ICMC SMC 2014 Conference, 2014.

[7] 都築 圭太, 中野 倫靖, 後藤 真孝, 山田 武志, 牧野 昭二, Unisoner：様々な歌手が同一楽曲を歌った Web 上の多様な歌声を活用する合唱制作支援インタフェース, 情報処理学会論文誌, 56(12), pp.2370–2383, 2015.

[8] S. Wager, G. Tzanetakis, S. Sullivan, C. Wang, J. Shimmin, M. Kim1, P. Cook, INTONATION: A Dataset of Quality Vocal Performances Refined by Spectral Clustering on Pitch Congruence, Proc. ICASSP2019, pp.476–480, 2019.

[9] 中野 倫靖, 後藤真孝, 平賀譲, 楽譜情報を用いない歌唱力自動評価手法, 情報処理学会論文誌, 48 (1), pp.227–236, 2007.

[10] J. Driedger, S. Balke, S. Ewert, M. Muller, Template-Based Vibrato Analysis in Complex Music Signals, Proc. ISMIR 2016, 2016.

[11] 後藤 真孝, 齋藤 毅, 中野 倫靖, 藤原 弘将, 歌声情報処理の最近の研究, 日本音響学会誌, 64 (10), pp.616–623, 2008.

[12] D. Hoppe, M. Sadakata, P. Desain, Development of Real-time Visual Feedback Assistance in Singing Training: A Review, Journal of computer assisted learning, 22(12), pp.308—316, 2006.

[13] T. Nakano, M. Goto, Y. Hiraga, MiruSinger: A Singing Skill Visualization Interface Using Real-Time Feedback and Music CD Recordings as Referential Data, Proc. the IEEE International Symposium on Multimedia (ISM 2007) Workshops, 2007.

[14] 山本 雄也, 中野 倫靖, 後藤 真孝, 寺澤 洋子, 平賀 譲, ポピュラー音楽における模倣歌唱を用いた歌唱テクニックの頻度・特徴・生起箇所の分析, 情報処理学会 研究報告音楽情報科学（MUS）, 2021-MUS-132, 20 pp. 1–8, 2022.

[15] M. Shiraishi, K. Ogasawara, T. Kitahara, HAMOKARA: A System for Practice of Backing Vocals for Karaoke, Proc. SMC 2018, pp.511–518, 2018.

[16] F. Moschos, A. Georgaki, G. Kouroupetroglou, FONASKEIN: An Interactive Software Application for the Practice of the Singing Voice, Proc. SMC 2016, pp.326–331, 2016.

[17] O. Mayor, J. Bonada, A. Loscos, Performance Analysis and Scoring of the Singing Voice, Proc. AES 35th International Conference, 2009.

[18] T. Nakano, M. Goto, VocaRefiner: An Interactive Singing Recording System with Integration of Multiple Singing Recordings, Proc. SMC 2013, pp.115–122, 2013.

[19] 近藤芽衣, 伊藤貴之, 中野倫靖, 深山覚, 濱崎 雅弘, 後藤 真孝. 同一楽曲に対する多数の歌唱と正解歌唱の音高推移分布の可視化. 情報処理学会 インタラクション 2021 (インタラクティブ発表), pp. 745–750, 2021.

[20] A. Klapuri, A Method for Visualizing the Pitch Content of Polyphonic Music Signals, Proc. ISMIR 2009, 2009.

[21] L. Jure, E. Lopez, M. Rocamora, P. Cancela, H. Sponton, I. Irigaray, Pitch Content Visualization Tools for Music Performance Analysis, Proc. ISMIR 2012, 2012.

[22] E. Gomez, J. Bonada, Tonality Visualization of Polyphonic Audio, Proc. ICMC 2005, 2005.

[23] K. W. E. Lin, H. Anderson, N. Agus, C. So, S. Lui, Visualising Singing Style Under Common Musical Events Using Pitch-Dynamics Trajectories and Modified TRACLUS Clustering, Proc. ICMLA'14, 2014.

[24] T. Kako, Y. Ohishi, H. Kameoka, K. Kashino, K. Takeda, Automatic Identification for Singing Style Based on Sung Melodic Contour Characterized in Phase Plane, Proc. ISMIR 2009, 2009.

[25] Y. Uchida, T. Itoh, A Visualization and Level-of-Detail Control Techniquefor Large Scale Time Series Data, Proc. IV09, pp.80–85, 2009.

[26] C. Perin, F. Vernier, J.-D. Fekete, Interactive Horizon Graphs: Improving the Compact Visualization of Multiple Time Series, Proc. ACM CHI 2013, pp.3217–3226, 2013.

[27] Y. Wang, F. Han, L. Zhu, O. Deussen, B. Chen, Line Graph or Scatter Plot? Automatic Selection of Methods for Visualizing Trends in Time Series, IEEE Trans. on Visualization and Computer Graphics, 24(2), pp.1141–1154, 2018.

[28] M. Imoto, T. Itoh, A 3D Visualization Technique for Large Scale Time-Varying Data, Proc. IV10, pp.17–22, 2010.

[29] G. Oliveira, J. Comba, R. Torchelsen, M. Padilha, C. Silva, Visualizing Running Races through the Multivariate Time-Series of Multiple Runners, Conference on Graphics, Patterns and Images, 2013.

[30] R. Mazza, Introduction to Information Visualization, Springer, ISBN:978-1-84800-218-0, 2009.

[31] J. J. van Wijk, E. W. van Selow, Cluster and Calendar based Visualization of Time Series Data, Proc. InfoVis'99, 1999.

[32] Y. Tanahashi, C.-H. Hsueh, K.-L. Ma, An Efficient Framework for Generating Storyline Visualizations from Streaming Data, IEEE Trans. on Visualization and Computer Graphics, 21(6), pp.730–742, 2015.

[33] H. Kawahara, I. Masuda-Katsuse, A. de Cheveigne, Restructur-

ing Speech Representations Using a Pitch Adaptive Time-frequency Smoothing and an Instantaneous Frequency Based on F0 Extraction: Possible Role of a Repetitive Structure in Sounds, Speech Communication, 27, pp.187-–207, 1999.

[34] D. Holten, Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data, IEEE Trans. on Visualization and Computer Graphics, 12(5), pp.741–748, 2006.

[35] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, CoRR, Vol. abs/1505.04597 , 2015.

[36] 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

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

[38] L. Shi, J. K. Nielsen, J. R. Jensen, M. A. Little, M. G. Christensen, Robust Bayesian Pitch Tracking Based on theHarmonic Model, IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27, 11, pp.1737–1751, 2019.