社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－HCI－128 (9)
2008－MUS－ 75 (9)
2008／5／28

VocaListener: ユーザ歌唱を真似る歌声合成パラメータを自動推定するシステムの提案

中野 倫靖　　後藤 真孝

産業技術総合研究所

t.nakano@aist.go.jp m.goto@aist.go.jp

あらまし　本研究では、歌声合成を使用した音楽制作を支援するために、ユーザの歌唱音声から歌声合成パラメータを自動推定するシステム VocaListener を提案する。従来、ユーザの歌唱音声から、音高や音量等を推定して歌声合成パラメータとする研究はあったが、歌声合成の条件 (歌声合成システムやその音源データ) の違いに対してロバストでなく、入力歌唱を真似るだけでは、ユーザの歌唱力を超えることが出来ないという問題もあった。そこで VocaListener では、合成された歌唱が入力歌唱と近くなるように、合成パラメータを反復更新することで、上記の条件の変化へ対処する。さらに、入力歌唱に対して、音高のずれやビブラートなどの歌唱要素を修正できる支援機能も提供する。本稿では、市販の歌声合成システムを対象に、音高と音量に関する合成パラメータを推定した結果を報告する。

VocaListener: An Automatic Parameter Estimation System for Singing Synthesis by Mimicking User's Singing

Tomoyasu Nakano　　Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST)

Abstract　To support use of singing synthesis in producing music, we propose a system called *VocaListener* that automatically estimates parameters for singing synthesis from user's singing voice. Although there was a previous method that can estimate singing synthesis parameters of the F0 and power from user's singing voice, it was not robust with respect to different singing synthesis conditions (e.g., singing synthesis systems and their singer databases) and had difficulty overcoming limitations of user's singing skill. To deal with those different conditions, VocaListener repeatedly updates singing synthesis parameters so that synthesized singing can be closer to the user's singing. Moreover, we provide functions that help to modify the user's singing by correcting off-pitch phrases or changing vibrato. By using a singing synthesis system on the market, this paper reports results of estimating synthesis parameters of the F0 and power.

1　はじめに

本研究では、歌声合成システムを利用する多様なユーザが、魅力的な歌声を自由自在に作れたり、歌唱という音楽表現の可能性を広げることを支援できる技術の開発を目指す。歌声合成システムは、個人が歌唱付き楽曲を制作するのを容易にし、歌唱の表現を自在にコントロールできる重要なツールである。また、インターネットを介して、音楽の共同制作や新しいコミュニケーションを生み出している現状がある。さらに、高品質な歌声合成を目指すことは、人間の歌声知覚・生成機構の解明にも繋がる取り組みである。

本稿では、入力としてユーザが歌唱音声を与え、その歌唱の分析結果を編集しながら歌声合成を行える VocaListener を提案する。これによってユーザは歌うだけで、それを基にした表情豊かな歌声を合成できる。また、ユーザ歌唱の分析結果を編集することで、ユーザ自身が歌唱できない表現 (音高が声域より高い場合など) に対して歌声合成を行える機能も提案する。

これまで、「人間らしい歌声」を作るために、歌声合成に関する様々な研究がなされており、サンプリングした歌唱音声の素片 (波形) を連結する方式 [1–3] や、歌声をモデル化して合成を行う方式 (HMM 合成) [4] があった。また、朗読音声から歌唱音声を分析合成する研究 [5–7] では、ユーザの声質を保存したままの高品質な歌声合成が検討されてきた。これらの研究によって、現在では「人間らしい歌声」の合成が可能となりつつあり、商品化されているものもある [3, 8]。

これらの技術をユーザが利用するためには、歌詞と楽譜情報 (何を歌わせるか) と、歌唱の表情 (どう歌わせるか) を入力するインタフェースが必要となる。前者は従来、歌詞と楽譜 (音高・発音開始時刻・音長) を与える方法 [2–4]、歌詞のみを与える方法 [9]、朗読音声と歌詞・楽譜を与える方法 [5–7]、歌唱音声と歌詞を与える

方法 [10] があった。後者は、ユーザが表情に関するパラメータを調整する方法 [2,3]、歌い方や歌唱スタイルをモデル化しておく方法 [4,6]、演奏記号 (crescendo 等) を入力する方法 [7]、歌唱音声から表情パラメータを抽出する方法 [10] があった。しかし、歌唱音声を入力として与え、入力歌唱自体を修正できるものはなかった。

ヤマハ株式会社の Vocaloid [3] では、ユーザはピアノロール形式のスコアエディタで歌詞と楽譜情報を入力し、表情付けパラメータを操作して歌声を合成する。しかし、より自然、あるいはより個性的な歌声を得るためには、表情パラメータの細かな調整が必要であり、ユーザによっては、自分の望む歌声を作るのが困難であった。また、歌声合成の条件 (歌声合成システムやその音源データ) が異なると、パラメータを調整しなおす必要があった[1] 。

Janer *et al.* [10] は、歌唱音声と歌詞を入力として、音高、音量、ビブラート情報 (深さ・速さ) を抽出し、歌声合成パラメータとして与える手法を提案した。また、そのようにして得られた合成パラメータを、歌声合成システムのスコアエディタ上でユーザが編集することを想定していた。しかし、歌唱から抽出した音高等をそのまま合成パラメータとしたり、既存の歌声合成システムのエディタを通した編集では、歌声合成の条件の違いに対処できなかった。

さらに文献 [10] では、音声認識で用いられる Viterbi アラインメントによって、歌詞の音節毎の発音開始時刻と音長の決定 (以降、歌詞アラインメントと呼ぶ) も自動的に行っていた。ここで、高品質な合成音を得るためには、100%に近い精度の歌詞アラインメントが必要だが、Viterbi アラインメントではそのような精度を得ることが難しい。しかも、歌詞アラインメントの結果と、出力される合成音は完全には一致しない[2] が、そのような問題への対処は考えられていなかった。

上記の問題を解決するために、本研究では、合成された歌唱を入力歌唱と比較しながら、合成パラメータを反復更新していく。そして、入力歌唱の編集では、自動推定の結果自体を編集する。これらの枠組みによって、歌声合成の条件の違いを吸収して、ロバストに合成できる。また、歌詞アラインメントについては、推定結果が誤った箇所をユーザが指摘するだけで誤りを自動訂正する機能も提案する。

これ以降、**2** 章で VocaListener の全体像について説明した後、**3** 章 で実現方法を述べる。また **4** 章で、システムの運用と評価実験を行った結果を報告する。最後に、**5** 章で今後の展望とまとめを述べる。

[1] 例えば、歌声合成システムの音源データが異なると、同一のパラメータを与えても得られる合成結果が異なる。また、他の合成システムには、同じ合成パラメータを利用することはできない。

[2] 例えば、Vocaloid [3] では、子音と母音のペアの音節については、母音の開始が発音開始時刻となるよう、時間的に前へずらして合成される。また、音節の始端と終端は、前後の音節やそのあるなしによって、出力結果が変化する。

2 歌声合成パラメータ推定システム VocaListener

本研究では、合成歌唱を目標歌唱 (入力) へ近づけるコア技術を VocaListener-core、目標歌唱自体を編集する技術を VocaListener-plus と呼び、本章ではこれらの機能を概説する。また、それぞれで必要な要素技術を VocaListener-front-end と呼ぶ。これ以降、ユーザによって与えられた歌唱を目標歌唱、歌声合成システムによって合成された歌唱を合成歌唱と呼ぶ。

図 **1** にシステム全体の流れを示す。ユーザは、歌唱音声とその歌詞を入力として与える (Ⓐ)。システムは、それらの入力に対して分析を行うが、漢字かな混じり文をかな文字列に変換する際の誤りや、歌詞の割り当てでフレーズをまたがるような大きな誤りがあった場合は、ユーザが手作業で訂正する (Ⓑ, Ⓒ)。次に、VocaListener-plus によって、声域を変更したり、ビブラートの深さ等を調節したりできる (Ⓓ)。最後に、VocaListener-core によって、入力歌唱を真似る合成パラメータを推定する (Ⓔ)。この際、歌詞アラインメントの音節境界に誤りが生じていたら、ユーザはその箇所を指摘して訂正する (Ⓕ)。最後に、ユーザは推定されたパラメータによって合成された歌唱を得る (Ⓖ)。

2.1 VocaListener-plus: 目標歌唱の編集

VocaListener-plus は、歌唱入力の表現を広げるために目標歌唱自体を編集する機能であり、本稿では以下の二種類を提案する。これらは、状況に応じて利用すればよく、使わないという選択も可能である。

音高の変更機能

- 調子はずれ (off pitch) の補正
  音高がずれた音を修正する。
- 音高トランスポーズ
  自分では歌えない声域の歌唱を合成する。

歌唱スタイルの変更機能

- ビブラート深さ (vibrato extent) の調整
  ビブラートを強く・弱くという直感的操作で、自分好みの表現へ変更できる。
- 音高・音量のスムージング
  音高のオーバーシュート、微細変動等を抑制できる。

2.2 VocaListener-core: 歌声合成パラメータの推定

VocaListener-core は、次の 3 つの機構によって歌声合成パラメータを推定する。

- 歌声分析
- 歌声合成
- 合成パラメータ更新 (反復しながら更新)

まず目標歌唱から合成に必要な情報を分析・抽出する。ここで、分析は目標歌唱に対してだけでなく、合成歌唱に対しても行う。合成歌唱の分析が必要なのは、合成パラメータが同一であっても、歌声合成の条件の

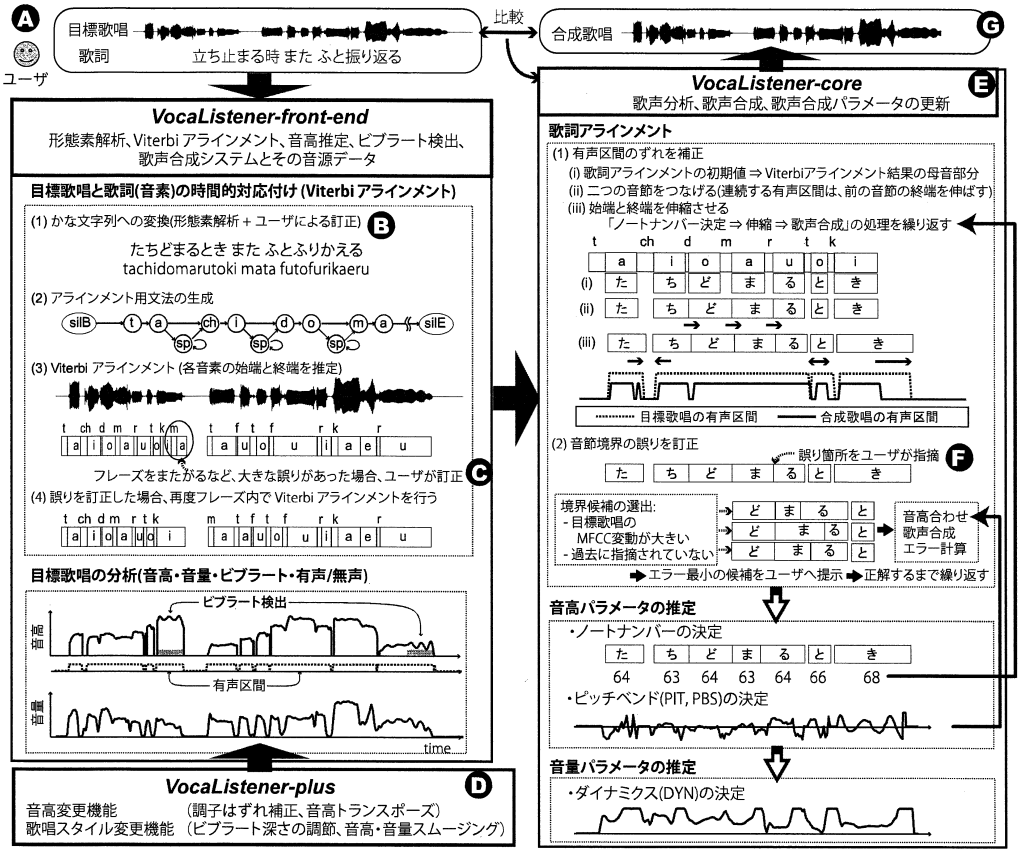

図 1: VocaListener の全体像（VocaListener-front-end, VocaListener-plus 及び VocaListener-core）

違いによって、合成される歌唱音声が異なるからである。これ以降、合成パラメータとの区別を明確にするため、分析によって得られた値は観測値と呼ぶ。

続いて、任意の歌声合成システムによって合成を行う。ただし本研究では、歌詞と楽譜情報を入力でき、また表情 (音高, 音量など) に関するパラメータを各時刻毎に指定できる必要がある。最後に、目標歌唱と合成歌唱を比較して、目標歌唱へ近づけるようにパラメータを反復更新する。

以上で述べた三つの機構は、人が歌を歌う過程を模していると考えることもできる。目標歌唱を分析して歌声を合成することは「課題曲に沿って歌ってみる」という発声練習に対応し、歌声分析によって「自分の歌声を客観的に聴いて目標と比較」して、「目標へ自己修正して再度歌う」ことで、目標歌唱へ近づけていく。

3 VocaListener の実現方法

本研究では音高と音量に焦点を絞って、VocaListener を実現する。声質を明示的に扱わない利点を二つ挙げる。

- 音高と音量は歌唱における最も重要なパラメータであり、対応している歌声合成システムが多いため。
- ユーザの歌唱の個人性を合成音に反映しにくくするため。その根拠としては、個人性知覚には声質 (スペクトル包絡) が重要であるという知見 [12] がある。しかし一方で、音高も場合によっては個人性知覚に影響を与えるという知見 [13] もあるので、音高のスムージングなどで個人性を弱くすることを試みる。

本章では、処理の流れに沿って、VocaListener-front-end、VocaListener-plus、VocaListener-core、の順にその実現方法を述べる。

3.1 VocaListener-front-end: 歌声分析及び歌声合成の要素技術

VocaListener-front-end として、「歌声分析」及び「歌声合成」に関する、要素技術を概説する。これ以降、歌唱音声信号はサンプリング周波数 44.1kHz のモノラル音声信号を扱い、処理の時間単位は 10 msec とする。

3.1.1 歌声分析の要素技術

歌声分析においては、歌唱の音響信号から、合成に必要な歌唱の要素を抽出する必要がある。以下、本研究では「音高」、「音量」、「発音開始時刻」、「音長」の抽出のための要素技術について概説する。これらの要素技術は、状況に応じて別の手法で代用しても構わない。

音高 歌唱音声の音高 (F_0: 基本周波数) を歌唱音声から抽出し、有声/無声の判定も同時に行う。F_0 推定には任意の手法が使えるが、本稿では、Gross Error が低いと報告されている SWIPE [14] を用いた。これ以降 F_0 (f_{Hz}) は、特に明記しない限り、次式で MIDI ノートナンバーに対応する単位の実数値 ($f_{\mathrm{Note\#}}$) へ変換して扱う。

$$f_{\mathrm{Note\#}} = 12 \times \log_2 \frac{f_{\mathrm{Hz}}}{440} + 69 \quad (1)$$

音量 音量は、N を窓幅、$x(t)$ を音声波形、$h(t)$ を窓関数として、以下のように計算した。

$$Pow(t) = \sum_{\tau=t-N/2}^{t+N/2} \left(\sqrt{(x(\tau) \times h(\tau - t))^2} \right) \quad (2)$$

現在の実装では、N は 2048 点 (約 46ms)、$h(t)$ はハニング窓とした。

発音開始時刻、及び音長 音声認識で使われる Viterbi アラインメントによって自動的に推定して利用する。ここで、漢字かな混じり文の歌詞は、形態素解析器 (MeCab [11] 等) によってかな文字列に変換した後、音素列に変換する。変換結果に誤りがあった場合は、ユーザが手作業で訂正する。Viterbi アラインメントでは、音節境界に短い無音 (short pause) が入ることを許容した文法を用いた。音響モデルには、朗読音声用の HMM [15] を、MLLR-MAP [16] によって歌唱音声に適応させて使用した。

3.1.2 歌声合成の要素技術

本研究では、歌声合成システムとしてヤマハ株式会社の開発した Vocaloid2 [3] の応用商品である、クリプトン・フューチャー・メディア株式会社の「初音ミク (以下、CV01)」及び「鏡音リン (以下、CV02)」 [17] を用いた。採用の理由としては、**2.2** 節で述べた条件を満たし、市販されていて入手しやすいこと、異なる音源データを利用できること、VSTi プラグイン (Vocaloid Playback VST Instrument) によって後述する反復推定 (イテレーション) の実装が容易であることがある。[3]

3.2 VocaListener-plus: 目標歌唱の編集

本章では、ユーザ補助機能群である VocaListener-plus についての、現在の実現方法を述べる。

3.2.1 音高の変更機能

目標歌唱の音高を変更する「調子はずれの補正」及び「音高トランスポーズ」機能を提案する。

[3] 観測値や歌声合成パラメータの推定における、処理の時間単位は前述のように 10 msec だが、VSTi によって合成する時のみ、合成パラメータを線形補間によって約 1 msec 毎 に与えた。

調子はずれの補正として、音高の遷移 (相対音高) が歌唱力の評価において重要であるため [19]、それを補正する。具体的には、音高遷移が半音単位となるように音高をずらす。このような補正方法を採ることで、ユーザ歌唱の歌唱スタイルを保持したまま調子はずれを補正できると考えられる。本稿では、有声音と判断された区間毎に、次式で定義する半音間隔に大きな重みを与える関数 (半音グリッド) をずらしながら、その区間の F_0 軌跡が最も適合するオフセット F_d を決定する。

$$F_d = \underset{F}{\mathrm{argmax}} \sum_t \sum_{i=0}^{127} \exp\left\{ -\frac{(F_0(t) - F - i)^2}{2\sigma_i^2} \right\} \quad (3)$$

現在の実装では、$\sigma = 0.17$ とし、$F_0(t)$ には事前にカットオフ周波数 5Hz のローパスフィルタをかけ平滑化[4] を行った。オフセット F_d は $0 \leq F_d < 1$ の範囲で計算し、音高を次式で変更する。

$$F_0^{(new)}(t) = \begin{cases} F_0(t) - F_d & (0 \leq F_d < 0.5) \\ F_0(t) + (1 - F_d) & (0.5 \leq F_d < 1) \end{cases} \quad (4)$$

続いて、音高トランスポーズは、ユーザ歌唱の音高を全体的、もしくは部分的にずらす機能である。本機能によって、ユーザ自身が表現できない声域の歌唱を合成することが出来る。変更したい区間を選択した後、次式によって F_t だけ変更する。

$$F_0^{(new)}(t) = F_0(t) + F_t \quad (5)$$

例えば、F_t を +12 とすれば、1 オクターブ高い音高の合成歌唱が得られる。

3.2.2 歌唱スタイルの変更機能

目標歌唱の歌唱スタイルを変更する機能として「ビブラート深さの調節」及び「音高・音量のスムージング」を提案する。

まず、$F_0(t)$ にカットオフ周波数 3 Hz のローパスフィルタをかけて、歌唱における F_0 の動的変動成分 [6] を除去した $F_{\mathrm{LPF}}(t)$ を得る。また、音量に関しても同様に $Pow(t)$ から $Pow_{\mathrm{LPF}}(t)$ を得る。ビブラート深さと音高・音量スムージングは、それぞれ調節パラメータ r_v と r_s によって、次式でその度合いを調節する。

$$F_0^{(new)}(t) = r_{\{v|s\}} \times F_0(t) + (1 - r_{\{v|s\}}) \times F_{\mathrm{LPF}}(t) \quad (6)$$

$$Pow^{(new)}(t) = r_{\{v|s\}} \times Pow(t) + (1 - r_{\{v|s\}}) \times Pow_{\mathrm{LPF}}(t) \quad (7)$$

基本的に r_v はビブラート自動検出法 [19] で検出された区間に適用し、r_s はそれ以外の区間に適用する。ここで、$r_v = r_s = 1$ の時に元の歌唱となる。これらは歌唱全体に対して適用しても、ユーザが指定した区間

[4] FIR フィルタを使用し、不自然な平滑化を避けるために、無音や閾値 (1.8 半音) 以上の周波数変化がない区間のみで平滑化した。

だけに適用してもよい。r_v を 1 より大きくすればビブラートをより強調し、r_s を 1 より小さくすれば F_0 の動的変動成分を抑制できる。例えば、オーバーシュートは、歌唱技量の差によらず生起するが、プロによる歌唱の方が、アマチュアによる歌唱よりも変動が小さいという知見 [20] があり、r_s を 1 より小さく設定することで変動を小さくできる。

3.3 VocaListener-core: 歌声合成パラメータの推定

VocaListener-core では、歌声分析によって得られた目標歌唱と合成歌唱の各観測値に基づいて、歌声合成パラメータを推定する (VocaListener-plus で目標歌唱を編集した場合は、その結果を観測値として用いる)。以降、図 **1** の詳細を説明する。

3.3.1 初期値の決定

まず、歌詞アラインメント、音高及び音量に関する初期値を与える。歌詞アラインメントには、Viterbi アラインメントによって得られた母音の開始時刻と終了時刻を初期値として与えた。

音高に関するパラメータは、Vocaloid2 では「音符の音高 (ノートナンバー)」「ピッチベンド (PIT)」「ピッチベンドセンシティビティ (PBS)」[5] である。ここで、PIT は −8192〜8191、PBS は 0〜24 の値を取り、デフォルト値はそれぞれ 0, 1 である。PBS が 1 なら、ノートナンバーから ±1 半音の範囲を、16384 の分解能で表現できる。また、ノートナンバーは 0〜127 の値を取り、1 が半音、12 が 1 オクターブに相当する。一方、音量に関するパラメータはダイナミクス (DYN) であり、0〜127 の値を取る (デフォルト値は 64)。

合成パラメータとしての PIT, PBS, DYN 初期値は、全時刻でデフォルト値とした。ノートナンバーは後述の手法 (**3.3.3** 項) を用いて、音節毎に適宜決定する。

3.3.2 歌詞アラインメントの推定、及び誤り訂正

音響モデルによって歌詞 (音素列) と目標歌唱を対応付けると、Viterbi アラインメントの誤りに加えて、歌声合成システムに対して指定した発音開始時刻や音長とずれて合成される問題がある。したがって、Viterbi アラインメント結果をそのまま用いた歌詞アラインメントでは、目標歌唱と合成歌唱の有声区間 (信号処理によって有声と判断された区間) にずれが生じてしまう。

そこでまず、有声区間のずれを以下の二つの処理によって補正する (図 **1** Ⓔ (1) 有声区間のずれを補正)。

- 二つの音節が繋がっておらず、かつ、目標歌唱ではその区間が有声と判定されていた場合、前の音節の終端を次の音節の始端まで伸ばす。
- 合成歌唱の有声区間が目標歌唱とずれている音節の始端と終端を、一致するように伸縮させる。

[5] PIT は音符の音高に対して、相対的に音高を時間軸上で動的に変化させることができるパラメータである。PBS によってその相対変化の幅を設定できる。

これらの処理と歌声合成 (ノートナンバーも推定する) を繰り返して行い、目標歌唱と合成歌唱の有声区間をあわせていく。

続いて、その合成歌唱をユーザが聴いて、ある音節境界が誤っていることに気付いて指摘すると、他の境界の候補が提示される。その候補は、目標歌唱の MFCC の変動 (時間変化) が大きい上位 3 箇所のそれぞれについて、まず音高を後述する反復計算で合わせて合成し、得られた合成歌唱と目標歌唱との振幅スペクトル距離が最小のものとした。それも誤りだと指摘されたら、次の候補を提示していく (最終的には手作業で修正してもよい)。MFCC の変動 $Mf(t)$ は、次数 I の $\Delta\mathrm{MFCC}(t,i)$ を用いて、次式で定義する。

$$Mf(t) = \sum_{i=1}^{I} \sqrt{\Delta\mathrm{MFCC}(t,i)^2} \tag{8}$$

現在の実装では、MFCC は 16kHz にリサンプリングした目標歌唱から算出し、次数 $I = 12$ である。また、振幅スペクトル距離は、目標歌唱と合成歌唱の振幅スペクトルをハニング窓 (2048 点) で算出し、それぞれを $S_{\mathrm{org}}(t,f)$, $S_{\mathrm{syn}}(t,f)$ として次式で定義する。

$$\mathrm{err}^2_{\mathrm{env}} = \sum_t \sum_{f=50\mathrm{Hz}}^{3000\mathrm{Hz}} \left(\overline{S_{\mathrm{org}}(t,f)} - \overline{S_{\mathrm{syn}}(t,f)} \right)^2 \tag{9}$$

$$\overline{S_{\{\mathrm{org|syn}\}}(t,f)} = \frac{S_{\{\mathrm{org|syn}\}}(t,f)}{\sum_f S_{\{\mathrm{org|syn}\}}(t,f)} \tag{10}$$

ここで、母音の特徴が現れる第 2 フォルマントまでを良く含むように、周波数 f には 50Hz〜3000Hz の帯域制限を設けた。また t は、対象の音節境界から前後 2 音節の区間を計算した (図 **1**Ⓕでは、「どまると」の区間)。

最後に、上記の処理で適切に訂正しきれない箇所のみ、ユーザが手作業で訂正を行う。

3.3.3 音高パラメータの推定 (1): ノートナンバー

観測された F_0 からノートナンバーを決定する。合成歌唱の音高観測値は、PIT と PBS の組み合わせによっては、ノートナンバー± 2 オクターブまで表現可能であるが、大きな PBS では量子化誤差が大きくなってしまう。そこで、その音符の区間に存在する音高の出現頻度から、PBS の値が小さくなるように、以下の式でノートナンバー (Note#) を選択する (図 **2**)。

$$\mathrm{Note\#} = \underset{n}{\mathrm{argmax}} \left(\sum_t \exp\left\{ -\frac{(n - F_0(t))^2}{2\sigma^2} \right\} \right) \tag{11}$$

ここで、$\sigma = 0.33$ として計算し、t は音符の始端から終端の時刻で計算した。これにより、F_0 が長い時間留まっているノートナンバーを選択することになる。

3.3.4 音高パラメータの推定 (2): ピッチベンド

ノートナンバーは固定したまま、合成歌唱の音高 $F0^{(n)}_{\mathrm{syn}}(t)$ が目標歌唱の音高 $F0_{\mathrm{org}}(t)$ に近づくように、

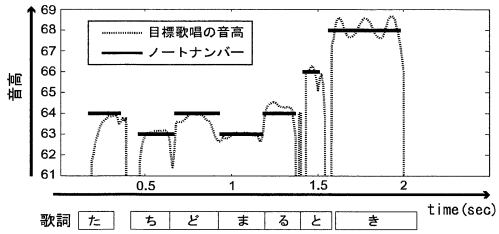

図 2: 目標歌唱の音高 (F_0) と選択されたノートナンバー

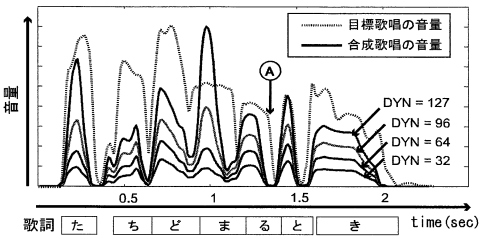

図 3: 目標歌唱と合成歌唱の音量観測値の違い

イテレーション (反復計算) によって音高パラメータ (PIT, PBS) を更新して推定する。

時刻 t、n 回目のイテレーションにおける PIT と PBS をノートナンバーに対応する値へ変換したものを $Pb^{(n)}(t)$ とすると、更新式は以下のようになる。

$$Pb^{(n+1)}(t) = Pb^{(n)}(t) + \left(F0_{\mathrm{org}}(t) - F0^{(n)}_{\mathrm{syn}}(t)\right) \quad (12)$$

このようにして得られた $Pb^{(n+1)}(t)$ から、PBS が小さくなるように、PIT と PBS を決定する。

3.3.5 音量パラメータの推定 (1): 目標音量の相対値化

目標歌唱の音量観測値は、収録条件の違い等が原因でその絶対的な値が変化するため、相対値化を行う。すなわち、音量の相対的な変化を表現するパラメータを推定するために、目標歌唱の音量を α 倍する。図 **3** に、DYN の値を 0〜127 まで変化させた合成歌唱と、目標歌唱の音量観測値をそれぞれ示す。

ここで、目標歌唱の相対変化を完全に表現するためには、全時刻で目標歌唱の音量を、DYN= 127 で合成した歌唱の音量以下に調整する必要がある。しかし、そのような条件を図 **3** のⒶの箇所などでも満たそうとすると、目標音量が小さくなりすぎて、量子化誤差が大きくなってしまう。

そこで、図 **3**Ⓐのような一部の再現を断念する代わりに、全体としての再限度が高くなるよう相対値化を行う。目標歌唱の音量観測値を $Pow_{\mathrm{org}}(t)$、DYN が 64 の時の合成歌唱の音量観測値を $Pow^{\mathrm{DYN=64}}_{\mathrm{syn}}(t)$ として、次式を最小化する相対値化係数 α を決定した。

$$\mathrm{err}^2 = \sum_t \left(\alpha Pow_{\mathrm{org}}(t) - Pow^{\mathrm{DYN=64}}_{\mathrm{syn}}(t)\right)^2 \quad (13)$$

3.3.6 音量パラメータの推定 (2): ダイナミクス

こうして得られた相対値化係数 α は固定したまま、音量パラメータ (DYN) を反復推定する。そのために、まずは全ての DYN における合成歌唱の音量観測値を取得する必要がある。そこで、DYN= $(0, 32, 64, 96, 127)$ のそれぞれで実際に各フレーズを合成して、音量観測値を取得しておき、その間は線形補間で求めた。

n 回目のイテレーションにおいて、DYN から上述のように求めた音量観測値へ変換したものを $Dyn^{(n)}(t)$ とし、その DYN で合成された歌唱の音量観測値を $Pow^{(n)}_{\mathrm{syn}}(t)$ とすると、更新式は以下のようになる。

$$Dyn^{(n+1)}(t) = Dyn^{(n)}(t) + \left(\alpha Pow_{\mathrm{org}}(t) - Pow^{(n)}_{\mathrm{syn}}(t)\right) \quad (14)$$

このようにして得られた $Dyn^{(n+1)}(t)$ から、上述の、DYN とその音量観測値の関係を利用して、音量パラメータ DYN に変換する。

4 運用及び評価実験

本章では、VocaListener-plus の運用結果を示し、VocaListener-core を「歌詞アラインメントの誤り訂正機能の有効性」、「イテレーションの必要性」及び「音源データの違いに対する頑健性」の観点から評価する。

4.1 VocaListener-plus の運用

本節では、VocaListener-plus の各機能について、運用結果を一部報告する。詳細な評価は今後の課題である。

図 4 に、音高変更機能として「調子はずれ補正」を、歌唱スタイル変更機能として「ビブラート深さの変更」及び「音高スムージング」を適用した結果を示す。音高が補正されること、ビブラートのみの深さを変更可能なこと、スムージングによってプレパレーションなどの変動を抑制可能なことが分かった。

ただし、調子はずれの補正については、有声区間毎に補正を行ったため、短い音符が適切に補正されない場合があった。今後は、このような問題へも対処する。

4.2 VocaListener-core の評価: 実験条件

VocaListener-front-end には前章で述べた技術を利用し、歌声合成システム (Vocaloid2) では、「ビブラートをつけない」、「ベンドの深さを 0 %」と設定した以外は全てデフォルト値を用いた。音源データとしては CV01 及び CV02 を用いた。今回の実験では便宜上、目標歌唱として、ユーザ歌唱の代わりに RWC 研究用音楽データベース (ポピュラー音楽) RWC-MDB-P-2001 [18] の伴奏なし歌唱データを用いた。

以下の A〜B の二種類の実験を行った。それぞれの実験で利用した楽曲を**表 1** に示す。

実験 A 長い歌唱 (曲中の 1 番) を利用し、歌詞アラインメントの誤り訂正機能の有効性を評価する。

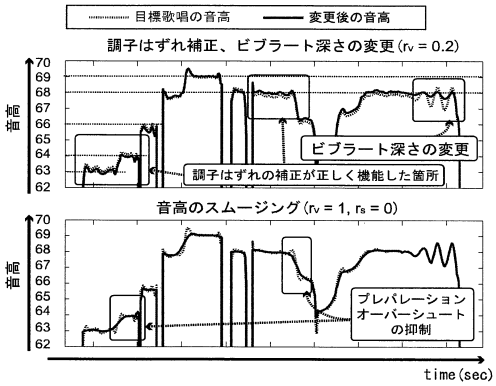

図 4: 音高変更機能、及び歌唱スタイル変更機能の運用

表 1: 実験で用いた目標歌唱及び歌声合成用音源データ

実験番号	曲番号	使用箇所	曲の長さ	目標歌唱 (歌手名)	合成用音源データ
A	No.07	1 番	103 秒	緒方智美	CV01
A	No.16	1 番	100 秒	吉井弘美	CV02
B	No.07	冒頭	2.4 秒	緒方智美	CV01,02
B	No.16	冒頭	3.5 秒	吉井弘美	CV01,02
B	No.54	冒頭	2.7 秒	凛	CV01,02
B	No.55	冒頭	2.9 秒	鏑木 朗子	CV01,02

※曲番号は RWC-MDB-P-2001

表 2: 音節境界の誤りを指摘した数、及び回数 (実験 A)

曲番号	合成用音源データ	音節総数	誤り指摘 n 回目の誤り数			
			0 (初期値)	1	2	3
No.07	CV01	166	8	3	1	0
No.16	CV02	128	1	0	—	—

実験 B 短い歌唱 (曲中の 1 フレーズ) を利用し、以下で定義するエラー ($\mathrm{err}^{(n)}_{\{\mathrm{f0|pow}\}}$) 及び相対エラー量 ($\Delta\mathrm{err}^{(n)}_{\{\mathrm{f0|pow}\}}$) を用いて、パラメータ推定におけるイテレーションの必要性と頑健性を評価する。

$$\mathrm{err}^{(n)}_{\mathrm{f0}} = \sum_t \left(F0_{\mathrm{org}}(t) - F0^{(n)}_{\mathrm{syn}}(t) \right)^2 \quad (15)$$

$$\mathrm{err}^{(n)}_{\mathrm{pow}} = \sum_t \left(Pow_{\mathrm{org}}(t) - Pow^{(n)}_{\mathrm{syn}}(t) \right)^2 \quad (16)$$

$$\Delta\mathrm{err}^{(n)}_{\{\mathrm{f0|pow}\}} = \frac{\mathrm{err}^{(n)}_{\{\mathrm{f0|pow}\}}}{\mathrm{err}^{(n=0)}_{\{\mathrm{f0|pow}\}}} \times 100 \quad (17)$$

ただし、実験 B では、パラメータ更新の評価が目的であるため、歌詞アラインメント (発音開始時刻と音長) については、人手で正解を与えた。

4.3 VocaListener-core の評価: 実験結果

本節では、前節で述べた二つの実験 (A, B) について、それぞれの実験結果を述べる。

4.3.1 実験 A: 歌詞アラインメントの誤り訂正

VocaListener-front-end での Viterbi アラインメント結果は、No.07 ではフレーズをまたぐ等の大きな誤りは起きず、No.16 では大きな誤りが 2 箇所起きた。それらを手作業で直した後、実験 A を行った結果を表 **2** に示す。No.07 では、計 166 個の音節について、8 箇所の境界誤りがあり、それらは 3 回の指摘で訂正できたことを表す。自動推定に誤りが発生する箇所としては、音節境界の直後の音節が/w/ や /r/ (半母音・流音)、/m/ や /n/ (鼻音) で始まる箇所が多かった。

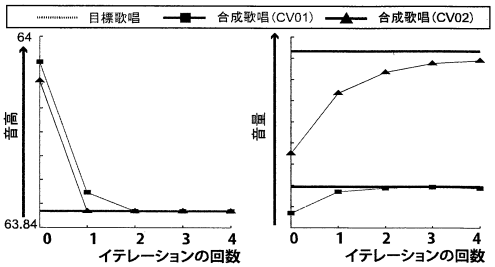

図 5: イテレーションによる音高・音量の推移 (実験 B)。音高と音量のそれぞれ 0.84sec の箇所を示している。音量の目標値は、CV01 と CV02 で相対値化係数 α が異なる。

表 3: n 回目のイテレーションにおける相対エラー量 [%] (実験 B: No.07)

推定したパラメータ	歌手データ	イテレーション n 回目の相対エラー量			
		1 回目	2 回目	3 回目	4 回目
音高	CV01	13.8	4.7	2.1	**2.4**
音高	CV02	8.1	3.7	2.3	1.7
音量	CV01	19.8	17.9	17.6	17.5
音量	CV02	16.0	14.2	13.9	13.8

太字は、エラーが増加したことを示す。

4.3.2 実験 B: ユーザ歌唱からの合成パラメータ推定

実験 B で対象としたどの曲に対しても、イテレーションによってエラーは減少した。4 回のイテレーションにおける初期値からの相対エラー量は、音高に関しては 1.7〜2.8 %、音量に関しては 13.8〜17.5%であった。これを No.07 について詳しく見ると表 **3** のようになり、そのうちの一箇所を取り出した結果を図 **5** に示す。

4.4 考察

表 **2** の結果からは、音節境界の誤り自体が少ないこと、2,3 回の指摘でその誤りが改善できることが分かった。No.07 での結果の例では、166 箇所という多数の音節に対し、計 12 箇所を指摘することで正しい音節境界が得られた。このことから、本手法はユーザの労力削減に寄与できると考えられる。

また、図 **5** 及び表 **3** から、イテレーションによってエラーが減少し、目標歌唱へ近づいていくといえる。音源データが変わることで初期値が異なっても、最終的に目標歌唱の音高・音量を得るためのパラメータを推定できた。ただし、音高パラメータ推定における、CV01 での 4 回目のイテレーションでは、エラーが増加していた (表 **3**)。これは、音高パラメータの量子化誤差が原因と考えられる。このような誤差は音量パラメータに

も存在し、場合によってはエラーが若干増加した。しかし、既に高い精度で合成パラメータが得られていることが多く、合成歌唱の品質への影響は少なかった。

入力歌唱の音高と音量を高い精度で近似することで、元の歌手の個人性を保持しているかのように聴こえる箇所があった。定量的な評価はしていないが、齋藤 他 [13] が述べているように、音高変化が歌唱の個人性知覚に影響を与えている可能性がある。

なお、本システムで合成した歌唱を目標歌唱として与え、パラメータの再推定を試みた結果、元のパラメータとほぼ同じとなることも確認した。本稿ではユーザの歌唱を目標歌唱として入力することを前提に説明したが、このように歌声合成システムの出力を入力してもよい。例えば、過去に CV01 用に手作業でパラメータ調整した合成歌唱を目標歌唱として、本システムで CV02 用にパラメータ推定すれば、手作業による再調整なしで音源データ (声色) を切り替えられて便利である。

5 おわりに

本研究は、人間の歌唱を入力としてそれを近似する歌声合成パラメータを推定するシステム VocaListener を提案した。また、ユーザが入力歌唱自体を修正できる VocaListener-plus も提案した。

今後は、以下のように研究を発展させていきたい。

より人間らしい合成歌唱の実現

ブレス音 (吸気音, 息継ぎ音) は、合成歌唱をより人間らしく聴こえさせるために重要であると考えられるため、ブレスの自動検出手法 [21] を利用して付与する。また、本稿では、声質に関するパラメータは敢えて推定していなかったが、声質 (スペクトル包絡) の動的な変動を組み込むことができれば、より人間らしい歌唱の実現につながる可能性がある。

VocaListener-plus の機能の充実

ユーザがより歌声合成システムを使いやすくするためには、VocaListener-plus の機能を拡張するとよい。調子はずれの補正法の改良や、他人の歌唱スタイルを利用する機能等が考えられる。後者は例えば、様々なビブラートを収録したテンプレートを用意し、好みのビブラートを状況に応じて付与する機能が考えられる。

『メタ歌声合成システム』の実現

従来、様々な歌声合成システムやその音源データが存在し、ユーザは手作業でその合成パラメータを決定していた。しかし、歌声合成の条件が変わると、パラメータの再調整が必要となる問題があった。本システムはそのような問題を解決し、ユーザはパラメータを一度だけ調整すれば、歌声合成システムや音源データに依存せず、同一の表現を様々な条件で合成できる。本研究では、これを**メタ歌声合成システム**と名付けて提案し、今後、機能の追加や拡張を行う予定である。

人間を知る追求

本研究の根底には、「人間らしい歌唱」とは何かを解明し、より人間を知ることがあり、本システムは、そうした歌声研究の基本ツールとしても貢献できる。例えば、音高や音量を独立に真似た合成歌唱を用いて心理実験を行うことで、歌唱の個人性知覚に関する新しい知見が得られる可能性がある。

謝辞

本研究の一部は、科学技術振興機構 CrestMuse プロジェクトによる支援を受けました。本研究では、ヤマハ株式会社及び、クリプトン・フューチャー・メディア株式会社の「CV01」「CV02」を使用させて頂きました。本研究に対し有益な議論をして頂き、VSTi ホストの実装へご助言を頂いた藤原 弘将氏 (産総研) に感謝致します。また、音響モデルの適応などでご助言を頂いた 緒方 淳 氏 (産総研)、歌声合成に関して有益なご意見を頂いた 齋藤 毅 氏 (産総研) に感謝致します。本研究では、RWC 研究用音楽データベース (ポピュラー音楽 RWC-MDB-P-2001) を使用しました。

参 考 文 献

[1] J. Bonada *et al.*: "Synthesis of the Singing Voice by Performance Sampling and Spectral Models," In *IEEE Signal Processing Magazine*, Vol.24, Iss.2, pp.67–79, 2007.

[2] 吉田 由紀 他: "歌声合成システム : CyberSingers," 情処研報 99-SLP-25-8, pp. 35–40, 1998.

[3] 剣持 秀紀 他: "歌声合成システム VOCALOID – 現状と課題," 情処研報 2008-MUS-74-9, pp.51–58, 2008.

[4] 酒向 慎司 他: "声質と歌唱スタイルを自動学習可能な歌声合成システム," 情処研報 2008-MUS-74-7, pp.39–44, 2008.

[5] 河原 英紀 他: "高品質音声分析変換合成システム STRAIGHT を用いたスキャット生成研究の提案," 情処学論, Vol.43, No.2, pp.208–218, 2002.

[6] 齋藤 毅 他: "SingBySpeaking: 歌声知覚に重要な音響特徴を制御して,話声を歌声に変換するシステム," 情処研報 2008-MUS-74-5, pp.25–32, 2008.

[7] 森山 剛 他: "好みの歌唱様式による歌詞朗読音声からの歌唱合成," 情処研報 2008-MUS-74-6, pp.33–38, 2008.

[8] NTT-AT ワンダーホルン, <http://www.ntt-at.co.jp/product/wonderhorn/>

[9] 米林 裕一郎 他: "Orpheus: 歌詞の韻律を利用した Web ベース自動作曲システム," インタラクション 2008, pp.27-28, 2008.

[10] J. Janer *et al.*: "Performance-Driven Control for Sample-Based Singing Voice Synthesis," In *DAFx-06*, pp.42–44, 2006.

[11] 工藤 拓, MeCab: Yet Another Part-of-Speech and Morphological Analyzer, < http://mecab.sourceforge.net/ >

[12] 河原 英紀 他: "モーフィングに基づく歌唱デザインインタフェースの提案と初期検討," 情処学論, Vol.48, No.12, pp.3637–3648, 2007.

[13] 齋藤 毅 他: "歌声の個人性知覚に寄与する音響特徴の検討, " 音講論集, 2-Q-26, pp.601-602, 2007.

[14] A. Camacho: "SWIPE: A Sawtooth Waveform Inspired Pitch Estimator for Speech And Music," Ph.D. Thesis, University of Florida, 116p., 2007.

[15] 河原 達也 他: 連続音声認識コンソーシアム 2002 年度版ソフトウェアの概要, 情処研報 2003-SLP-48-1, pp.1–6, 2003.

[16] V.V. Digalakis *et al.*: "Speaker adaptation using combined transformation and Bayesian methods," *IEEE Transactions on Speech and Audio Processing*, Vol.4, No.4, pp.294–300, 1996.

[17] クリプトン, VOCALOID2 特集, <http://www.crypton.co.jp/mp/pages/prod/vocaloid/>

[18] 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," 情処学論, Vol.45, No.3, pp.728-738, 2004.

[19] 中野 倫靖 他: 楽譜情報を用いない歌唱力自動評価手法," 情処学論, Vol.48, No.1, pp.227–236, 2007.

[20] 齋藤 毅 他: "歌声の基本周波数変化に含まれるオーバーシュートの知覚への影響に関する検討, 聴覚研資, Vol.36, No.7, pp.611–616, 2006.

[21] 中野 倫靖 他: "無伴奏歌唱におけるブレスの音響特性と自動検出," 音講論集, 1-11-12, 2008.