情報処理学会論文誌 Vol.56 No.12 2370–2383 (Dec. 2015)

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

都築 圭太[1,a)] 中野 倫靖[2,b)] 後藤 真孝[2,c)] 山田 武志[3,d)] 牧野 昭二[3]

受付日 2015年3月4日, 採録日 2015年9月2日

概要：本論文では，Web上で公開されている「1つの楽曲を様々な歌手が歌った歌声」から，合唱と呼ばれる作品を制作するためのインタフェースUnisonerを提案する．従来，このような合唱制作では，伴奏を抑制した各歌声波形を楽曲のフレーズごとに切り貼りし，音量の大小や左右のバランスを調整したうえで重ね合わせる必要があり，時間と労力がかかっていた．それに対してUnisonerでは，歌詞に基づいた楽曲内位置の指定と，歌手アイコンのドラッグアンドドロップ操作に基づいた音量調整を可能とするインタフェースによって，直感的かつ効率的に合唱を制作することができる．さらに，歌声のF_0（基本周波数）とMFCC（Mel Frequency Cepstral Coefficient）に基づいた音響的な類似度や，MFCCに基づいた歌手性別の推定結果に加え，再生数などのWeb上のメタデータを活用した歌手検索機能も持つ．このような機能を実現するためには，伴奏をともなう歌声のF_0推定手法や，歌声と歌詞のアラインメント手法が必要となるが，それらの推定結果に誤りが含まれることが問題となる．そこで本論文では，誤りを含む単一の歌声からの推定結果に対し，複数の歌声の推定結果を統合して誤りを削減する手法を提案する．評価実験の結果，Unisonerによって合唱制作時間が短縮されること，提案手法によりF_0推定と歌詞アラインメントにおける誤りが減少することを確認した．

キーワード：歌声情報処理，ユーザインタフェース，基本周波数推定，歌詞アラインメント

Unisoner: An Interface for Derivative Chorus Creation from Various Voices Singing the Same Song on the Web

Keita Tsuzuki[1,a)] Tomoyasu Nakano[2,b)] Masataka Goto[2,c)]
Takeshi Yamada[3,d)] Shoji Makino[3]

Received: March 4, 2015, Accepted: September 2, 2015

Abstract: This paper proposes Unisoner, an interface for assisting the creation of derivative choruses, in which voices of different singers singing the same song are overlapped on top of one shared accompaniment. In the past, it was time-consuming to create such choruses because creators had to manually cut and paste vocal fragments from different singers, and then adjust the volume and panning of each voice. Unisoner enables users to perform such editing tasks efficiently by selecting phrases using lyrics and by dragging and dropping the corresponding icons onto a virtual stage. Moreover, Unisoner can search vocals with acoustic similarity based on F_0 and MFCC, estimated gender, and metadata such as the number of views. We use a vocal F_0 estimation technique from polyphonic audio signals, and a technique to synchronize audio signals with lyrics. However, estimation errors occur using conventional techniques for F_0 and lyric alignment, so we propose a novel method of reducing those errors by integrating the estimated results from many voices singing the same song. The experimental results confirmed that Unisoner can shorten the time for creating derivative choruses, and the proposed methods can reduce the estimation error of F_0 and lyric alignment.

Keywords: singing information processing, user interface, F_0 estimation, lyrics alignment

1. はじめに

近年，様々なエンドユーザが既存の楽曲を歌った2次創

[1] 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305–8568, Japan
[3] 筑波大学システム情報系
Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan

[a)] tsuzuki@mmlab.cs.tsukuba.ac.jp
[b)] t.nakano@aist.go.jp
[c)] m.goto@aist.go.jp
[d)] takeshi@cs.tsukuba.ac.jp

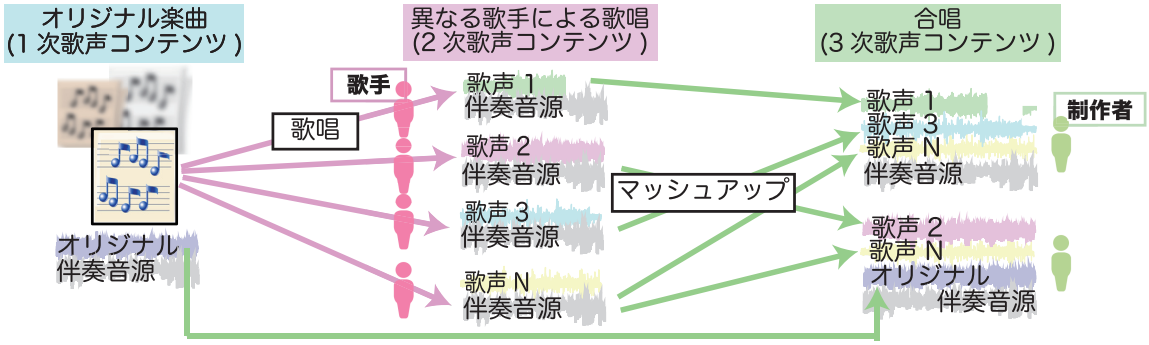

図 1 Web 上で公開されているオリジナル楽曲から，それを多数の歌手が歌った歌声コンテンツが派生し，さらにマッシュアップ（重ね合わせ）がなされて合唱が制作される過程の概要

Fig. 1 Relationship among original songs, vocal covers, and derivative choruses. Various singers sing the same song to create vocal covers. From these vocals, derivative choruses are created.

作コンテンツが，Web 上で多く公開されるようになった．そのようなコンテンツは，単に視聴されるだけでなく，同一楽曲を歌唱したものを複数切り貼りして重ねることで（マッシュアップすることで），あたかも複数人が 1 つの歌を歌っているような「合唱」と呼ばれる作品を創作する活動にもつながっている．2015 年 7 月の時点で，ニコニコ動画[*1]上では約 2 万件の合唱が投稿されており，再生回数が 200 万回を超える人気作品も存在する[*2]．1 つの合唱に含まれる歌手（歌声コンテンツ）の数は，数人の場合から 100 人以上となる場合まであり，合唱の再生数上位 20 個の動画には，平均 12 人の歌手の歌声が使用されている．

図 1 に示すように，ある 1 つの楽曲を 1 次歌声コンテンツとすると，別のユーザ（歌手）が同じ伴奏音源（カラオケ）に合わせて歌唱した 2 次歌声コンテンツが存在し，合唱は 3 次歌声コンテンツとして位置付けられる．本論文で取り扱う合唱では多くの場合，同一楽曲が同じメロディラインで歌われている．そうした歌唱形式は斉唱と呼ばれるが，本論文では楽曲の進行とともに歌手が切り替わる点に着目し，合唱と呼ぶ．

本論文では，多様なユーザが自分好みの合唱を手軽に制作できる新たなインタフェース Unisoner を提案する．従来，異なる楽曲を自動的にマッシュアップするインタフェース [1] や，異なる楽曲・動画のマッシュアップにおける制作支援インタフェース [2], [3], [4] が提案されてきたが，これらは楽曲の歌詞を考慮していなかった．合唱の制作では歌詞に基づいて使用する歌声を切り替えられることが求められるため，合唱制作の支援に関して機能が不十分であった．

また Unisoner を実現するために必要となる，歌声以外の伴奏音（複数の楽器音）の抑制，歌声の F_0 推定および歌声と歌詞の時間対応付け（歌詞アラインメント）についてもあわせて説明する．特に，F_0 推定と歌詞アラインメントについては，同一楽曲に対して複数歌声が存在することを活用して推定誤りを削減する新しい手法を提案する．

[*1] http://www.nicovideo.jp
[*2] たとえば，http://www.nicovideo.jp/watch/sm5132988

2. 合唱の制作効率化に向けた課題と解決法

本章では，合唱の制作を効率化するための課題と解決方法について説明する．本論文では以下のような状況を想定している．

- 同一伴奏にのせた複数の歌声がそれぞれ音響ファイルとして与えられる．
- 伴奏のみの音響ファイルは与えられるが，歌声のみの音響ファイルは与えられない．
- 楽譜情報は利用しない．
- 歌詞のテキストファイルは与えられるが，各単語の出現時刻は付与されていない．

2.1 現状の合唱制作の流れ

合唱の制作は，通常 DAW（Digital Audio Workstation）や波形編集ソフトウェアを用いて次のようなステップで行われる．

(1) **前処理** 重ね合わせる歌声コンテンツは，そのままでは演奏開始時間にずれがある場合が多いため，その時間を同期させる．また，重ねた際の違和感を軽減するために，歌声コンテンツに含まれる伴奏音を抑制する．

(2) **使用する歌手とフレーズの吟味** 各歌手の歌声を重ねたときの音を確認して，使用する歌手を決定する．また，同じ歌手でも区間（フレーズ）ごとに歌い方を変えている場合もあるので，どのフレーズを使用するかもあわせて吟味する．

(3) **歌声の切り貼り** (2) の結果に基づいて各歌声の波形を切り貼りし，それらを DAW などのソフトウェア上で重ね合わせるように配置する．

(4) **音量の調節** 各歌声に対して，その音量の大小や左右

チャネルのバランス調節を行う．

2.2 インタフェース上の課題

以上をふまえ，合唱制作を効率化するためには，インタフェースの観点から以下 2 つの課題の解決が必要である．

課題 1：楽曲中の位置や歌声の特徴を把握しやすいインタフェースの実現

合唱制作に使用される従来のツールは，通常波形表示に基づいたインタフェースであり，実際に音を再生して聞いて確認する必要がある．したがって，楽曲のどこを歌っているのか，どんな歌声なのかを把握するのに時間がかかる．

課題 2：多数の歌声を効率的に扱えるインタフェースの実現

構想した合唱を実現するためには，多くの歌声コンテンツの中から適切なものを見つけ出す必要がある．また，合唱制作に用いるツールは，使用する歌声すべてに対する使用タイミングや音量の調節が必要であり，手間がかかる．

Unisoner では，課題 1 を解決するために，まず歌詞に基づいた時間指定（クリッカブルな歌詞）や歌声の切り貼りを可能とすることで，楽曲中のどこを歌っているのかという時間情報を把握しやすくする．従来，歌詞を使用した楽曲内の位置決定は，再生位置 [5] や録音位置の指定 [6] に用いられることがあった．また，歌声の特性が可視化されたアイコン（歌手アイコン）により，各歌声の特徴を事前に把握しやすくする．

課題 2 については，歌手の声質や歌い回しに基づいた歌手の検索機能を実現することで歌声コンテンツを見つけやすくする．また，フレーズに配置した歌手とそれぞれの音量を複製できる機能により，複数のフレーズにおける使用タイミングと音量の調節を可能とする．

2.3 信号処理上の課題

以上で述べたインタフェースを実現するためには，伴奏音が含まれた歌声コンテンツに対し，信号処理における以下の課題も解決する必要がある．

課題 3：伴奏音に頑健な信号処理技術の実現

歌声の基本周波数（F_0）推定手法と歌詞アライメントが，歌手検索機能とクリッカブルな歌詞の実現のために必要となる．しかし，従来の推定手法を適用するのみでは大きな誤差が含まれる場合があり，ユーザが意図したインタラクションが適切に行えない．

課題 3 を解決するために，単一の歌声に対して既存の推定手法を用いるだけでなく，複数の歌声における個々の推定結果を統合することで，F_0 推定誤りと歌詞アラインメント結果の誤りを削減する手法をそれぞれ提案する．各歌声は同一楽曲を歌っているため，個々の推定結果に誤差が含まれていても，他の歌声に対する推定結果が正しい場合に，その結果を活用して推定結果を修正できる．

3. 合唱制作支援インタフェース Unisoner

本章では先述の課題を解決する，合唱制作支援インタフェース Unisoner について説明する（**図 2**）．ユーザは Unisoner を使用することで，様々な歌声コンテンツを聴き比べながら，手軽に合唱を制作できる．**図 3** に，従来ツールと Unisoner の違いをまとめて示す．Unisoner は，2 章で述べた課題に対して，図 3 の「歌詞に基づいた楽曲内位置指定機能」，「歌手アイコンに基づいた歌手配置機能」，「歌声の特徴に基づいた歌手検索機能」によって解決する．以下ではこれらの機能について説明する．

本論文ではニコニコ動画の歌声コンテンツを対象とし，各歌声は伴奏抑制（4 章で後述）が適用されている．

3.1 歌詞に基づいた楽曲内位置指定機能

インタフェースに表示された歌詞に対して，マウスのクリック操作を行うことで効率的に楽曲内位置を指定できる（図 2Ⓐ）．さらに楽曲をフレーズに分割し，フレーズごとに歌声を配置することで歌手の切替えを表現できる．楽曲の分割は，歌詞をクリックして楽曲内位置を指定し，分割

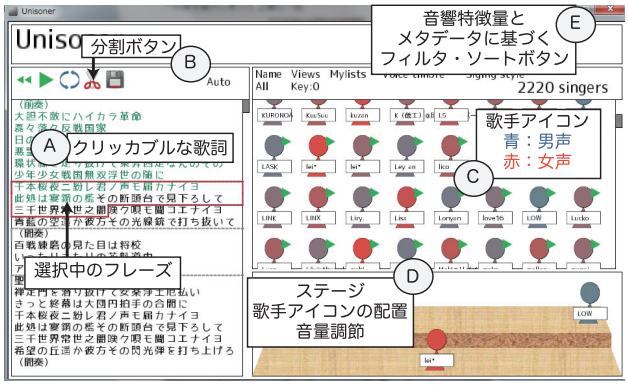

図 2 Unisoner の概要

Fig. 2 Overview of Unisoner.

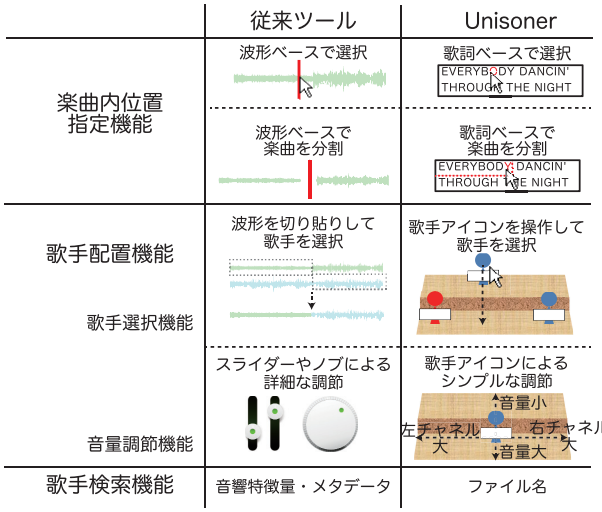

図 3 従来ツールと Unisoner の比較

Fig. 3 Comparison of the conventional tools and Unisoner.

ボタンをクリックすることで行える（図 2Ⓑ）．

3.2 歌手アイコンに基づいた歌手配置機能（歌手の選択と音量の調節）

各歌声コンテンツに対応するアイコン（図 2Ⓒ）を選択し，それをステージ上（図 2Ⓓ）へ並べることで，歌手の配置と音量の調節が可能である．各歌声の特徴を直感的に把握しやすくするために，歌手アイコンは，その歌声の男声らしさが高いほど青く，女声らしさが高いほど赤くなるよう色付けされている（4.1.5 項で説明）．各歌声の音量はステージ上の位置に応じて自動で決定される．ここで，前後段が全体的な音量の大小，左右の位置が左右チャネルのバランスに対応する．

また，あるフレーズにおける選択した歌手や歌手アイコンの配置を，別のフレーズに複製することができる．これはたとえば，1 番と 2 番のサビで同じ歌声を使いたい場合に便利である．具体的には，歌詞上（図 2Ⓐ）のあるフレーズを，別のフレーズへドラッグアンドドロップすることで複製できる．

3.3 歌声の特徴に基づいた歌手検索機能

合唱を構成する歌手を，大量の候補から選択することを支援するために，歌声の音響的特徴と，Web 上で公開されている歌声コンテンツのメタデータに基づいた歌手の検索機能を実装した（図 2Ⓔ）．具体的には，以下に示す歌手の並べ替え（ソート）を用いて，目的の歌声コンテンツを検索できる．

- 指定した歌声に対する声質，および歌い回しの類似度（4.1.4 項で説明）
- 歌手名[*3]，再生数，マイリスト数[*4]

さらにソートと併用して，歌声の男声らしさと女声らしさ，オリジナルの楽曲に対するキー（調）のずれの 2 つを用いた絞り込み（フィルタリング）が行える．たとえば，男声らしさが高い歌声コンテンツや，キーを 3 半音上げた歌声コンテンツを絞り込んで表示できる．

4. Unisoner における信号処理技術

本章では Unisoner の実現のために必要な信号処理技術について述べる．これらの手法は個々の歌声だけで処理が完結する手法と，多数の歌声を活用することで単一の歌声に対する推定誤りを削減する手法に分類できる．以下，それぞれについて説明する．

4.1 個々の歌声に適用する信号処理技術

本節では，Unisoner の実現のために必要な個々の歌声に適用する信号処理技術について説明する．なお，各歌手が歌唱した楽曲の伴奏音源は既知であるものとした．また，本論文で使用する音響信号はすべてサンプリング周波数が 16 kHz，量子化 bit 数が 16 のモノラル信号である．

また，本論文で使用する伴奏抑制手法は，伴奏音源が事前情報として必要であるが，ニコニコ動画に投稿されている VOCALOID 楽曲の多くでは，歌声コンテンツなどの二次利用を想定して楽曲の伴奏音源が公開されている [7]．

4.1.1 キーのずれと大まかな時間ずれの推定

歌声コンテンツが伴奏音源に対して時間・周波数（キー）ともにずれていることがあるため，まずはこれを補正する必要がある．そのために，伴奏音源と歌声コンテンツの対数周波数軸上の振幅スペクトログラム間の二次元相互相関を計算し，1 semitone（半音）単位でのキーのずれと 100 ms 単位の時間ずれを同時に推定する．伴奏からのキーのずれを推定するのは伴奏抑制（4.1.3 項で説明）を適切に行うためと，前述したフィルタリング（3.3 節）のためである．二次元相互相関関数を用いることで，ある程度の時間長を考慮しながらキーのずれと時間ずれが同時に推定できる．ここで，対数周波数スペクトログラムを用いることで，キーのずれを線形に扱い，二次元相互相関での推定を可能とした．

$x(t,m)$ を窓関数により切り出された m 番目のフレーム，t を時間方向のインデックス，N を離散フーリエ変換の点数，f_k を semitone k に対応する周波数 [Hz]，f_r をサンプリング周波数（16 kHz）とするとき，スペクトログラム $X(k,m)$ は次の式 (1) で求められる．

$$X(k,m) = \sum_{t=0}^{N-1} x(t,m) e^{-j\omega_k t} \tag{1}$$

$$\omega_k = 2\pi \frac{f_k}{f_r} \tag{2}$$

$$f_k = 440 \times 2^{\frac{k-69}{12}} \tag{3}$$

窓関数には 2,048 点（128 ms）のハニング窓を使用し，N は 4,096 点，k の範囲は 1, 2, $\cdots$, 119（8.7, 9.2, $\cdots$, 7,902.1 Hz）とした．シフト幅は 1,600 サンプル（100 ms）とした．

また，式 (1) から求められる伴奏音源，歌声コンテンツの対数周波数軸上の振幅スペクトログラム $|A(k,m)|$, $|X(k,m)|$ 間の，二次元相互相関関数 $C(l,n)$ は，K が周波数ビン数，M が時間フレーム数（楽曲全体），l が周波数方向のずれ，n が時間方向のずれを表すインデックスであるとき，次式によって求められる．

$$C(l,n) = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} |A(k,m)||X(k-l,m-n)| \tag{4}$$

なお，1 オクターブ低く（高く）歌唱する場合，伴奏音源は原曲のキーと通常同じにするため，l の範囲は $[-6,6]$

[*3] 現在は動画投稿者の名前を歌手の名前として代替している．
[*4] ニコニコ動画においてこの歌声コンテンツを「お気に入り」として登録しているユーザの数．

（伴奏音源に対して ±6 semitone 以内のずれに相当）とした．ランダムに選んだ歌声コンテンツ 100 曲（うち 50 曲は伴奏のキーと異なった）のうち 97 曲について本手法で伴奏からのキーのずれを正しく推定できた．

また，本手法で推定された時間ずれと人手で求めた時間ずれとの誤差の中央値は 32.7 ms であった．これは，後述する伴奏抑制において，およそ 2 フレーム分のずれに相当し（フレーム長を 16 ms としたため），伴奏抑制後の歌声の品質劣化につながると考えられる．そこで，より正確に時間ずれを推定するため次項の処理を行う．

以下の時間ずれの推定と伴奏抑制では，ここで推定されたキーのずれを補正するように音高シフト[*5]した伴奏音源を用いる．

4.1.2 時間ずれの推定

次項で説明する伴奏抑制のため，歌声コンテンツと伴奏音源の開始タイミングを揃える必要がある．そこで本論文では，伴奏音源と歌声コンテンツ間の（一次元）相互相関を計算することにより，1 サンプル（62.5 μs）単位で各歌声コンテンツの伴奏音源に対する時間ずれを推定した．ここで，相互相関の計算には楽曲全体を使用しており，その間に伴奏音源と歌声コンテンツに共通の「伴奏のみの区間」が含まれると仮定する．その後，相互相関関数を最大化するサンプル数だけ各歌声コンテンツの開始時間をずらすことで，すべての歌声コンテンツの開始時間を伴奏音源に揃えることができる．

伴奏音源の時間波形 $a(t)$ と歌声コンテンツの時間波形 $x(t)$ 間の相互相関関数 $c(\tau)$ は，t と τ がサンプル番号を表すとき，次式で表せる．

$$c(\tau) = \sum_{t} a(t)x(t-\tau) \tag{5}$$

また，$c(\tau)$ を最大化する $\tilde{\tau}$ は歌声コンテンツの伴奏音源に対する時間方向のずれを表す．

$$\tilde{\tau} = \operatorname*{argmax}_{\tau} c(\tau) \tag{6}$$

なお，$\tilde{\tau}$ はキーのずれ推定の際に同時に求まる時間方向のずれの ±800 サンプル（50 ms）以内に制限する．前項，キーのずれ推定の評価に用いた歌声コンテンツ 100 曲に対してこの手法も適用したところ，推定された時間ずれと人手で求めた時間ずれとの誤差の中央値は 11.6 ms であった．したがって，伴奏抑制においては 1 フレーム以下（0.725 フレーム）のずれとなり，前項の結果から改善された．また，本実験で使用した楽曲は BPM（Beats Per Minute）が 154 であったため，11.6 ms は全音符の 134 分の 1 の音価に相当する（128 分音符以下）．したがって，複数の歌声コンテンツを重ねた際の聴取への影響も少ないと考える．

[*5] Audacity（http://audacity.sourceforge.net）を使用して実現．

4.1.3 伴奏抑制

歌声コンテンツに含まれる伴奏音をスペクトルサブトラクション法 [8] によって抑制する．$X(\omega,t)$, $A(\omega,t)$, $V(\omega,t)$ がそれぞれ歌声コンテンツ，伴奏音源，伴奏抑制された歌声のスペクトル，α（≥ 0）が伴奏音源の音量を調節するパラメータ，ω と t が周波数と時間を表すインデックスであるとき，スペクトルサブトラクション法は次式のように表せる．

$$V(\omega,t) = \begin{cases} 0 & (H(\omega,t) \leq 0) \\ H(\omega,t)e^{j \arg X(\omega,t)} & (\text{otherwise}) \end{cases} \tag{7}$$

$$H(\omega,t) = |X(\omega,t)| - \alpha|A(\omega,t)| \tag{8}$$

ここで α は，歌声コンテンツによって異なる音量を正規化するために必要であり，伴奏抑制後の歌声の音質は α に強く影響される．したがって，各歌声コンテンツに対して α を適切に決める必要がある．本論文では，歌声コンテンツ中の伴奏区間の音量と伴奏音源中の同一区間の音量比は楽曲を通して一定と仮定し，伴奏抑制後の歌声における非歌唱フレームの音量を最小化する方針で α を決定する．

具体的には，まず歌声コンテンツから非歌唱フレームを推定するために，$\alpha = 1$ で伴奏抑制を行った後の波形において，各フレーム（10 ms，160 サンプル）の音量（振幅スペクトルの二乗平均）を計算し，伴奏抑制後の歌唱の全フレームにおける平均音量を閾値として，それよりも小さいフレームを非歌唱フレームと見なした．これは，伴奏抑制を行うと，α の値にかかわらず各フレームの音量は小さくなるが，非歌唱フレームは歌唱フレームより伴奏抑制後の音量が小さい傾向にあったためである．キーのずれが推定済みである 100 曲の歌声について，非歌唱フレームであると推定されたフレームのうち，それが連続する最長の区間が実際に非歌唱フレームであった歌声は 72 曲であった．それ以外の 28 曲の歌声のうち 18 曲は，非歌唱区間ではあったものの極端に短いフレームであり，想定していた前奏区間や間奏区間を推定することはできなかった．また，残りの 10 曲については，推定された区間の始めや終わりなどに一部歌声が含まれていた．

楽曲ごとに推定された非歌唱フレーム全体を使用して α を決定する．この区間において，$|X(\omega,t)| - \alpha|A(\omega,t)|$ の絶対値が最小となるように，α を 0.0 から 2.0 まで 0.1 刻みで変化させて決定し，伴奏抑制で使用した．α の決定と伴奏抑制では，256 サンプル（16 ms）のハニング窓を 128 サンプル（8 ms）でシフトさせる STFT（Short-Time Fourier Transform）によって振幅スペクトルを算出して用いた．周波数分解能を確保するため，STFT は 512 点にゼロ詰めして行った．

なお，コンプレッサなどのエフェクタによる音質の変化やエンコード時の劣化などの影響で，伴奏音源と歌声コンテンツに含まれる伴奏音は必ずしも一致しない．しか

し，合唱では，伴奏抑制された歌声に伴奏を重ね直すため，個々の伴奏抑制後の歌声に伴奏音が残留しても，完成した合唱においては複数の伴奏が重なっているように聞こえるなどの聴感上の違和感はなかった．また，Unisoner を実際に使用した被験者からも聴感上の違和感についての言及はなかった．

4.1.4 歌声間の距離計算

Unisoner の歌手検索機能（3.3 節）を実現するために，声質の近さと歌い回しの近さに基づいた，歌声間の距離を求める必要がある．本論文では伴奏抑制された歌声コンテンツの声質と歌い回しに関する音響特徴量を，それぞれ GMM（Gaussian Mixture Model）でモデル化し，EMD（Earth Movers Distance）[9] によって GMM 間の距離を算出し，歌声間の距離とした．

声質の音響特徴量には 13 次元の MFCC（Mel Frequency Cepstral Coefficient）を，歌い回しの音響特徴量には 4.2 節で説明する手法を用いて求めた F_0 と ΔF_0 を使用した．MFCC は音声認識や話者認識，楽曲の音色特徴として用いられるなど，音色を特徴付ける音響特徴量の 1 つとして知られている．また，F_0 と ΔF_0 は歌唱スタイル [10] や，話声と歌声の識別 [11] において有効性が報告されている．

4.1.5 歌手の性別（男声/女声らしさ）の推定

歌手アイコンの色分け（3.2 節）と歌手検索機能（3.3 節）を実現するために，歌声の男声/女声らしさ[*6]を推定する必要がある．本論文では，Songrium [7] の男女度推定技術を参考に，SVM（Support Vector Machine）[12] を用いて，伴奏抑制された歌声コンテンツの各フレームが男声クラスに属する確率を求め [13]，全フレームの中央値を男声/女声らしさとした．学習データには異なる 4 つのオリジナル楽曲を歌唱した，10 曲（男声・女声それぞれ 5 曲ずつ）の伴奏抑制された歌声コンテンツを使用した（計 40 曲）．Songrium の男女度推定技術と異なり，SVM の特徴量には MFCC を使用した．また，学習データには歌声コンテンツの歌唱区間（30 秒，人手でラベル付け）を用いた．MFCC を用いたのは，MFCC が性別推定において一般的に用いられている特徴量であったためである．たとえば，MFCC を含む特徴量から学習された SVM を用いた歌手の性別推定に関する研究 [14] や，話し声の性別推定における有効性が報告されている [15]．

4.2 多数の歌声を活用する信号処理（1）：F_0 推定の誤り削減

Unisoner の歌手検索機能（3.3 節）の実現には，4.1.4 項で述べた歌いまわしに基づく歌声間距離の計算が必要となり，その計算において歌声コンテンツの F_0 が使用される．しかし，伴奏音をともなった歌声の F_0 推定は一般的に難しい課題であり，オクターブエラーや他の楽器音に起因する推定誤りが生じてしまう．

そこで，同一楽曲を歌った他の歌声コンテンツの推定結果を活用することで，このような誤りを削減する．歌声コンテンツの F_0 推定結果に時間局所的なエラーが含まれていても，多様な歌声コンテンツの F_0 推定結果を集計すると，その推定結果は各歌声コンテンツにおける真の F_0 値に近い値に集中するため，各フレームの F_0 推定結果が集中している周波数の近傍に推定範囲を制限する．本手法は，任意の F_0 推定手法に対して適用可能だが，本論文では gross error [16] に頑健とされている SWIPE' [17] を選択した．以下，本手法の手順を説明する．なお，本節で使用する歌声コンテンツには事前に 4.1.3 項の伴奏抑制を適用している．

4.2.1 F_0 推定

本論文では，F_0 推定の際に SWIPE' を用いるが，その際に周期性を判定する指標である pitch strength を閾値として信頼度の高いフレームを推定する．pitch sterngth は，雑音環境下の pitched/unpitched 区間識別手法において，有効性が確認されている [18]．

同一楽曲に対する 4,524 曲の歌声コンテンツに対して SWIPE' を用いて F_0 推定を行い，4.1.1 項の手法によってキーを補正した後，各フレームにおいて全歌声コンテンツの F_0 推定結果からヒストグラムを作成すると図 4① が得られる．図 4① より，多くの推定結果が赤線で囲まれた

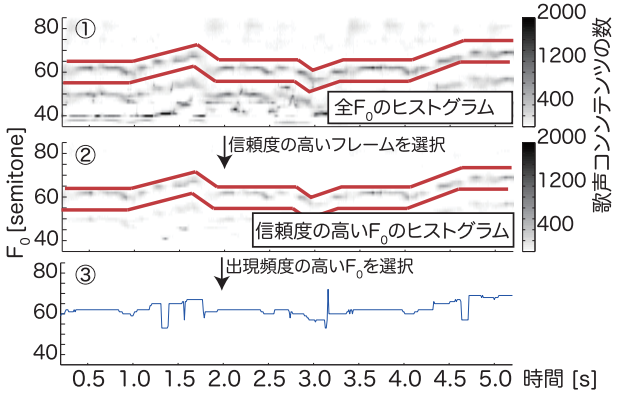

図 4 ①：歌声コンテンツ 4,524 曲における歌い出し 5 秒間の F_0 のヒストグラム．信頼度の制約なし．②：① から信頼度の高いフレームだけを選択したとき得られるヒストグラム．③：② のヒストグラムから図 5 の処理を用いてフレームごとに最も出現頻度の高い F_0（最頻 F_0）を求めて得られる軌跡．①，② 中の赤線は正解の F_0 に近い範囲

Fig. 4 ①: Histogram of F_0 values in 5 seconds after prelude for 4,524 vocal covers. ②: Histogram after selecting the frames with a high confidence value from ①. ③: Trajectory of the most frequent F_0 at each frame, which was obtained by applying the processing in Fig. 5 to ②. The red lines in ① and ② indicate a range surrounding the correct F_0.

[*6] 本手法では男声らしさが p_m なら女声らしさ p_f は $1-p_m$ となるため，男声らしさと女声らしさの推定は同等の意味を持つ．

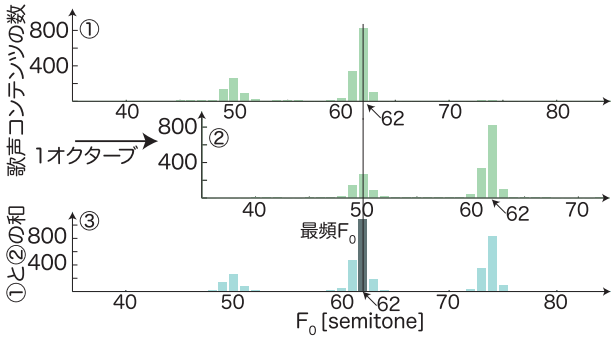

図 5 ①：図 4② の歌い出し 3 秒後のフレーム（10 ms）．②：① を 12 semitone 上にずらしたもの．③：① と ② の和

Fig. 5 ①: Histogram obtained from frames of 3 seconds after prelude in Fig. 4 ②. ②: ① was shifted by 12 semitone. ③: Sum of ① and ②.

正解に近い範囲に集中していることが分かる．しかし，40 semitone 付近に多くの推定誤差が現れてしまっている．一方，各フレームにおいて pitch strength が閾値よりも高い歌声コンテンツの F_0 推定結果のみから作成されたヒストグラムが図 4② である．図 4① に比べて 40 semitone 付近の誤差が減少していることが分かる．

4.2.2 最頻 F_0 の推定

図 4② の開始 3 秒後のフレームを取り出したヒストグラムを**図 5** の ① に示す．62 semitone に鋭いピークが見られ，1 オクターブ離れた 50 semitone にもなだらかなピークが見られる．これは，男女の音域のような 1 オクターブ異なる歌声が存在することが原因である（50 付近が男声，62 付近が女声）．

本論文では，このようなオクターブの違いも考慮したうえで，各フレームで最も多く現れている F_0 を最頻 F_0 と呼び，これを用いて F_0 の再推定範囲を決定する．ここで，semitone は連続値であるため出現回数を数えるためには事前に離散化しておく必要がある．本論文では小数点第 1 位で各フレームの推定 F_0 を四捨五入することで推定 F_0 を離散化した．最頻 F_0 はあるフレームにおける F_0 値の出現回数と，それより 1 オクターブ低い（−12 semitone に相当）F_0 の出現回数（図 5②）の和が最大になる F_0 として求める．図 5③ のフレームでは 62 semitone が最頻 F_0 となる．これを全フレームについて計算すると図 4③ のような軌跡が得られる．

4.2.3 最頻 F_0 と F_0 推定範囲の決定

前述のようにして得られた最頻 F_0 を使用して，F_0 の推定範囲を決定する．しかし，最頻 F_0 ではオクターブの違いまでは推定していない．つまり，分析対象の歌声が最頻 F_0 付近の音高で歌っているのか，それとも 1 オクターブ違う高さで歌っているのかは未知である．しかし，仮に推定誤りが含まれていても，曲全体で見れば最頻 F_0，その 1 オクターブ低い F_0，1 オクターブ高い F_0 のいずれかに近いと考えられる[*7]．

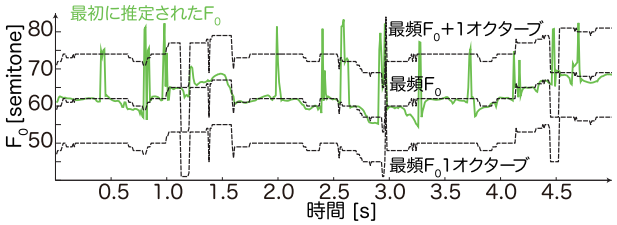

図 6 SWIPE' によって推定された F_0，最頻 F_0，および最頻 F_0 を ±1 オクターブした軌跡

Fig. 6 Trajectory of F_0 estimated by SWIPE', trajectory of the most frequent F_0, and trajectories obtained by shifting the most frequent F_0 by ±1 octave.

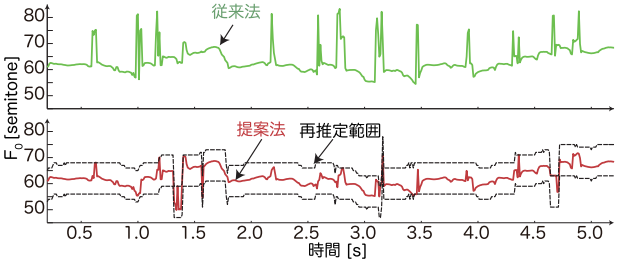

図 7 上図：SWIPE' を用いて推定した F_0．下図：提案手法を用いて推定した F_0，および再推定範囲（最頻 F_0 ± 6 semitone）

Fig. 7 Upper: F_0 estimated by SWIPE'. Lower: F_0 estmiated by the proposed method, and a re-estimation range (the most frequent F_0 ± 6 semitone).

そこで，**図 6** のように，最頻 F_0（$f_{mode}(t)$）を 1 オクターブ低くした $f_{mode-}(t)$ と高くした $f_{mode+}(t)$ を計算した．そして，1 回目の推定から得られる F_0（$f_0(t)$）に対する $f_{mode}(t)$，$f_{mode+}(t)$，$f_{mode-}(t)$ の 3 つの軌跡間の距離 d をそれぞれ次式のように計算し，d が最小となる軌跡を使用して再推定を行った．

$$d = \sum_t \sqrt{(f_0(t) - f_{\mathrm{mode}}(t))^2} \tag{9}$$

たとえば，図 6 では，最頻 F_0（$f_0(t)$）が最も推定 F_0 に近いので最頻 F_0 がそのまま再推定に使用される．

フレームごとに推定範囲を最頻 F_0 の ±6 semitone 以内に制限して再推定を行うと**図 7** 下図が得られる．赤線が再推定された F_0，黒線が推定範囲を表している．図 7 上図と比較して推定結果の乱れが減少することが分かる．

4.3 多数の歌声を活用する信号処理（2）：歌詞アラインメントの誤り削減

Unisoner の楽曲内位置選択機能（3.1 節）を実現するためには，各歌声コンテンツに共通で使用可能な，歌詞の時間情報が必要となる．しかし，伴奏をともなった歌声の歌

*7 図 5 の作成に用いた楽曲は女声による楽曲であるため，1 オクターブ高い音高で歌っている歌手は存在しない．しかし，一般的には男声による楽曲が女声で歌われることもあるため，本手法では 1 オクターブ高い音高も考慮する．

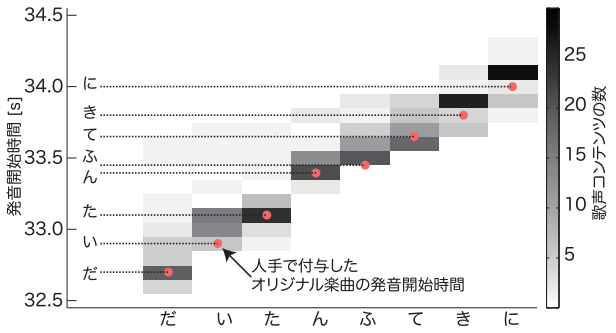

図 8 同一楽曲を歌った 50 曲の歌声コンテンツに対して Lyric-Synchronizer を適用して得られる推定発音開始時間の分布

Fig. 8 Distribution of start time of each syllable estimated by LyricSynchronizer for 50 vocal covers of the same song.

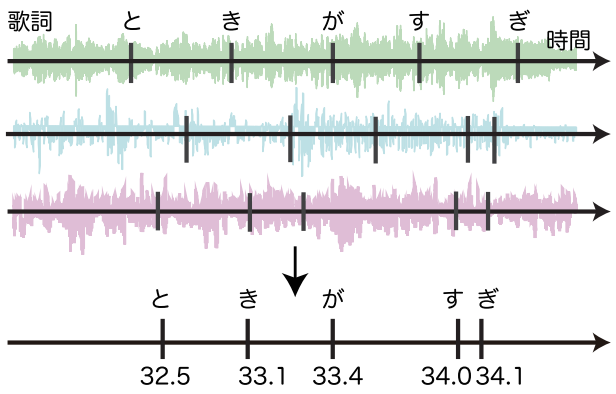

図 9 歌詞アラインメントの誤り削減手法の概要

Fig. 9 Overview of reduction of lyric alignment errors.

詞アラインメントも困難な課題であり，推定誤りの発生を避けるのは難しい．

同一楽曲を歌った歌声コンテンツ 50 曲の推定発音開始時間の分布を**図 8** に示す．この図から，各歌声コンテンツに対する推定結果には，ばらつきがあることが分かる．しかし，各歌声コンテンツの発音開始時間の推定値は狭い範囲に集中していることもあわせて見て取れる．そこで本手法（**図 9**）では，全歌声コンテンツから推定した発音開始時間について，歌詞の読みごとに中央値を計算しそれを発音開始時間として使用することで，歌詞アラインメントの誤りを削減する．

本手法も F_0 推定同様任意の歌詞アラインメント手法に適用することができるが，本論文では LyricSynchronizer [5] を使用した．LyricSynchronizer は，混合音中の歌声と歌詞を高精度にアラインメントする手法であり，歌詞にない発声を含む歌声に対しても頑健となるように実装されている．ただし，LyricSynchronizer が伴奏抑制などの処理がなされていない混合音の入力を想定しているため，入力として与える音響信号は，伴奏抑制適用前の音響信号（伴奏音を含む）とした．実際に予備実験を行った結果，伴奏を抑制した歌声に対する精度は，抑制前の歌声と比べて低かった．

LyricSynchronizer は音素単位で推定を行うため，本論文ではそれらを読み単位にまとめて使用した．たとえば，「大胆」という歌詞に対しては/d/，/a/，/i/，/t/，/a/，/N/という音素とその発音開始時間が得られるが，このうち/d/，/i/，/t/，/N/の発音開始時間を「だいたん」という読み（ひらがな）各文字に対する発音開始時間として使用した．

5. 評価実験

5.1 実験 A：Unisoner（インタフェース）の評価

本節では提案する合唱制作インタフェース Unisoner，多様な歌声を活用する F_0 推定手法，歌詞アラインメント手法についてその有効性を評価した結果について説明する．

Unisoner は合唱制作の効率化を目的としているため，制作時間に着目し，どの程度短縮可能かを被験者実験によって確認した．合唱制作には 2 章で述べたとおり，大きく分けて次のようなステップがある．

(1) 前処理（開始時間のずれ修正と伴奏抑制）
(2) 使用する歌手とそのフレーズの吟味
(3) 歌声の切り貼り
(4) 音量の調節

これらのステップのうち，手軽な合唱制作という観点から考えると，(1) 前処理，(3) 歌声の切り貼りの効率化が重要であると考えられる．(1) 前処理については，信号処理技術により自動化されているため，被験者実験では，(3) 歌声の切り貼りに着目して，どの程度効率化できたのか評価した．残りの (2) 使用する歌手とフレーズの吟味，および (4) 音量の調節の評価は今後の課題とする．

5.1.1 実験条件

全部で 7 つの歌声コンテンツ $S_1, S_2, \cdots, S_7$ を用いて 1 つの指定した合唱を制作するタスクを考える．本実験では被験者に以下のような合唱を制作するタスクを与えた．

- S_1，S_2，S_3 の 3 つを 1 番の A メロに配置
- S_4，S_5，S_6，S_7 の 4 つを 1 番の B メロに配置
- S_1，S_2，$\cdots$，S_7 の 7 つを 1 番のサビに配置

上記の合唱を実現するうえで，次の 3 つの方法による制作時間を測定した．

タスク 1 Unisoner を使用
タスク 2 従来ツールを使用（1 歌手につき 1 トラック）
タスク 3 従来ツールを使用（1 歌手 1 フレーズにつき 1 トラック）

タスク 1 は楽曲のフレーズへの分割が行われていない状態から，歌詞を選択してフレーズへ分割し（3.1 節），歌声アイコンをドラッグアンドドロップして適切なフレーズへ配置する（3.2 節）タスクである．使用する歌声の開始時間は 4.1.2 項の手法を用いて補正済みであり，各歌詞に対応する発音開始時間は 4.3 節で説明した手法を用いて推定した発音開始時間を使用した．本実験では，(3) 歌声の切り貼りの効率を測定するために（すなわち (2) 使用する

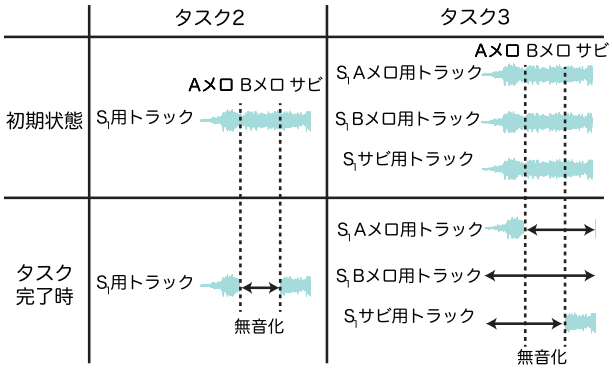

図 10 実験 A タスク 2・3 の初期状態とタスク完了時におけるトラックの状態の例（7 つの歌声コンテンツ S_1 から S_7 中，S_1 についてのみ説明）

Fig. 10 An example of the track status at the beginning and end of the experiment. Here one vocal cover S_1 is used from among the seven vocal covers S_1 to S_7.

歌手とフレーズの吟味に関する操作を不要とするために），Unisoner へ入力した歌声データは実験で使用する 7 つのみとした．また，歌声アイコンをドラッグアンドドロップする際にステージ上のどの位置に配置するかによって歌声アイコンに対応する歌声の定位角度と音量が変化する（3.2 節）が，(4) 音量の調節に関しては本実験の対象外であるため，それに関しては特に指示を与えなかった．

タスク 2 とタスク 3 は従来ツールを使用して合唱を制作するタスクである．その際，各歌声の全区間の波形が各トラックに入っている状態を初期状態とした（**図 10**）．ここでタスク 2 では，1 つの歌声につき 1 トラックを，タスクで 3 は 1 歌声 1 区間（A メロ，B メロ，サビ）に対して 1 トラック（すなわち 1 つの歌声につき 3 トラック）割り当てた．そのため，初期状態ではタスク 2 では各歌声ごとの波形が入力されており（7 トラック），タスク 3 では各歌声，各区間ごとに波形が入力されている（$7 \times 3 = 21$ トラック）．これは，楽曲を通して歌声コンテンツの定位角度が同じ合唱（タスク 2）か，歌声コンテンツの定位角度をフレーズごとに変更する合唱（タスク 3）かの違いを想定している．被験者には以下の指示を与えた．

不要な部分の無音化 S_1 の歌声が A メロとサビで必要ならタスク 2 では S_1 の B メロの区間を探して無音化するよう指示した（図 10，タスク 2）．タスク 3 では A メロ用トラックから B メロとサビを無音化，B メロ用トラックをすべて無音化[*8]，サビ用トラックの A メロと B メロ区間を無音化するという 3 つの操作を行うよう指示を与えた（図 10，タスク 3）．

区間の開始時間の統一 A メロ・B メロ・サビの各区間において，歌声ごとに開始時間が（わずかに）異なる可能性があるが，1 つの歌声の開始時間を調べれば他の歌声の開始時間も同じと見なしてよいものとした．Unisoner（タスク 1）でも各歌詞の発音開始時間を歌声単位で調節する操作するような操作は行えないため，タスク 2・3 がタスク 1 に対して不利になることを防ぐための指示である．

タスク終了の条件 無音化が完了した時点でタスク終了とした．つまり，タスク 2・3 では最後にトラックを 1 つにまとめる処理は行わなかった．

タスク 2・3 では被験者全員に使用経験があることから従来ツールとして Audacity[*9] を選択した．一般に音楽コンテンツ制作の現場では，DAW（Digital Audio Workstation）を用いることが通常であり，音量調節やエフェクトなどに関して，プラグインなどの作業をサポートするソフトウェアを利用することができる．しかしここでは，図 3 の従来ツールの要件（「波形ベースで」選択・分割），および複数トラックの同時再生が可能なインタフェースとの比較を行った．Audacity はシンプルな波形編集ソフトであり，自動化などはなされていない．しかし，本実験のタスク 2・3 を遂行するうえでは Audacity の機能で十分であり，遂行時間に DAW との大きな差が生じないように作業内容を設定した．

上記，本タスクで制作する合唱の構成は，web 上で公開されている人気の高い合唱[*10]の 1 番と同一とした．

5.1.2 実験手順

4 名の被験者（20 代男性）はタスク 1，2，3 またはタスク 1，3，2 の順にタスクを遂行した．なお，タスク 2 とタスク 3 については，1 度遂行すると各フレーズの開始時間を覚えてしまい，その後のタスク遂行の時間に影響を与える可能性がある．そのため，従来ツールを用いた 2 回目のタスクでは 2 番の区間を用いて実験を行った．被験者はまず，Unisoner の使い方の説明を受けた後，十分な時間をかけて操作の確認を行った，その後，実験タスクの説明，および A メロ，B メロ，サビが歌詞のどの部分に該当するかの説明を受け，タスク 1 を遂行した．次に，従来ツールを用いた合唱制作の方法を説明した後，タスク 2 とタスク 3 を遂行した．被験者全員に従来ツール（Audacity）の使用経験があったため，従来ツールの操作説明は行わなかったが，マルチトラックをまとめて無音化したり，ショートカットキーを用いたりして効率的に作業を行う被験者もいた．

被験者は全員，本論文で対象とする 2 次創作としての合唱の制作経験はなかった．また，実験に用いたオリジナル楽曲を事前に聴いたことがあって知っていたが，本タスクで参考とした合唱作品を聴いたことはなかった．被験者#2，被験者#3，被験者#4 はそれぞれ 5 年，14 年，2 年の演奏経験があり，被験者#1 と被験者#4 には楽曲制作の経験があった．

[*8] トラックをすべて無音化する操作はトラック全体の削除で代替してもよいものとした．

[*9] http://audacity.sourceforge.net

[*10] http://www.nicovideo.jp/watch/sm17125297

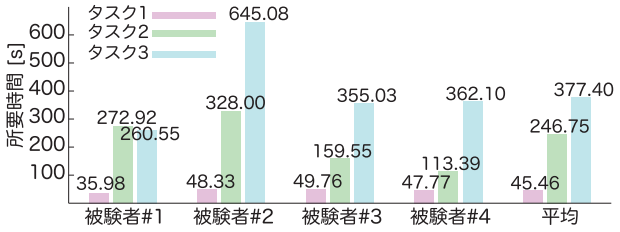

図 11 指定した合唱を制作するタスクに要した時間の被験者ごとの比較（実験 A）

Fig. 11 Comparison of task completing time for each subject (Experiment A).

5.1.3 実験結果

被験者ごとのタスク遂行に要した時間を**図 11** に示す．被験者がタスク 1，2，3 の完了に要した時間はそれぞれ平均で 46.22 秒，246.75 秒，377.40 秒であった．また，すべての被験者におけるタスク 1（Unisoner を使用）の所要時間はタスク 2，3（従来ツールを使用）に比べて短かった．

5.1.4 考察

実験結果より，Unisoner を使用した方が従来ツールを使用するより効率的にタスクを遂行できることが確認できた．従来ツールを用いた場合に Unisoner より時間がかかった大きな原因は，A メロなどの該当区間を探す際に前奏から聴いていく必要があるためである．実際，従来ツールを使用したタスク 2，3 ではすべての被験者が冒頭から楽曲を再生して波形を切るタイミングを探していた．実験終了後に被験者から得られたコメントでは，「従来ツールの方が馴染みがある」など，従来ツールの利点もあげられたが，Unisoner では「操作方法が直感的で簡単である」，「楽曲中の位置がイメージしやすい」，「前から聞いていく必要がない」とより多くの利点があげられた．

タスク 1 については，最も早くタスクを遂行した被験者と最も遅い被験者間では遂行時間に約 9.3 秒の差が生じた．これはタスク遂行時の操作ミスの有無によるものである．最も早くタスクを遂行した被験者#1 はミスをしておらず，それ以外の被験者は 1 回以上のミスがあった．また，被験者が必ず行わなければならない操作は楽曲の分割と歌声の配置の 2 つであるが，歌声と音量情報の複製機能を使用することによって，複数の歌声を素早く配置できたことがタスク遂行時間の短縮につながっていた．逆に，タスク遂行に時間がかかった原因は，誤った箇所での楽曲分割や誤ったフレーズへの歌手と音量情報の複製であった．しかし，このようなミスがあったにもかかわらず，どの被験者もタスク 1（Unisoner）の所要時間はタスク 2，3（従来ツール）より短かった．この結果は，Unisoner が操作に不慣れなユーザでも効率的に合唱制作を行えることを示している．

5.2 実験 B：F_0 推定性能の評価

提案手法が，従来手法を歌声コンテンツに対して適用した際の F_0 推定誤差をどの程度削減できるか評価した．

5.2.1 実験条件

ニコニコ動画において歌声コンテンツの投稿数が多いオリジナル楽曲[*11]を歌った歌声コンテンツ 5 曲（うち M1，M2，M3 が男声，F1，F2 が女声）の歌い出し 5 秒間を用いて，提案 F_0 推定手法の推定性能を評価した．

各歌声コンテンツは 4.1.3 項の手法を用いて伴奏抑制を行い，F_0 推定における時間分解能は 10 ms，周波数分解能は 0.1 semitone とした．

5.2.2 実験手順

F_0 の正解データは，音楽大学出身で歌声の音高を書き起こす作業の経験が十分にある音楽家 1 名が，音高推移を耳で聞いて書き起こしたものである．その際，ピッチベンド（連続的な音高変化）を用いて，可能な限り歌声の音高推移に近づけた．実際には書き起こしデータを，MIDI 音源を用いて再現し，それを出力した波形に SWIPE' を適用した結果得られる F_0 推定結果を正解データとして使用した．正解データのうち，pitch strength が 0 以下のフレームは非歌唱区間と見なして評価対象から除外した．

最頻 F_0 の推定には，評価対象の 5 曲を含む，同一楽曲を歌った 4,524 曲の伴奏抑制した歌声コンテンツを使用した．また，信頼度が高いフレームの選択に用いる pitch strength の閾値を $-\infty$（全フレームを使用），0, 0.1, ⋯, 0.5 と変化させた場合についてそれぞれ推定誤差を評価した．

推定誤差の大きさは誤差の絶対値の平均で評価した．正解 F_0 が $f(t)$，推定 F_0 が $\bar{f}(t)$ で，評価に用いるフレーム数が T であるとき，平均誤差 ϵ_f は次式のように計算した．ここで，F_0 の単位は式 (3) によって計算される semitone である．

$$\epsilon_f = \frac{1}{T}\sum_t |f(t) - \bar{f}(t)| \tag{10}$$

5.2.3 実験結果

図 12 に提案手法（pitch strength $= -\infty$, 0, 0.3, 0.5）と従来手法の平均推定誤差を示す．pitch strength を 0.3 に設定したとき，誤差が最小となった（5 曲の歌声コンテンツに対して平均 4.22 semitone の誤差）．また，その場合，提案手法は従来手法に比べてすべての歌声コンテンツで推定エラーが減少していた．M2，M3，F2 の歌声コンテンツについては Welch の t 検定 [19] において危険率 0.1%未満で提案手法と従来手法における推定誤差の平均値に統計的な有意差が見られた．pitch strength が 0.1，0.2 の場合，および 0.4 の場合については図 12 で結果を示していないが，どの歌声コンテンツにおいても pitch strength が 0.3 に近づくにつれて推定誤差が減少していた．

5.2.4 考察

提案手法は 5 つの歌声コンテンツすべてについて推定誤

[*11] http://www.nicovideo.jp/watch/sm15630734

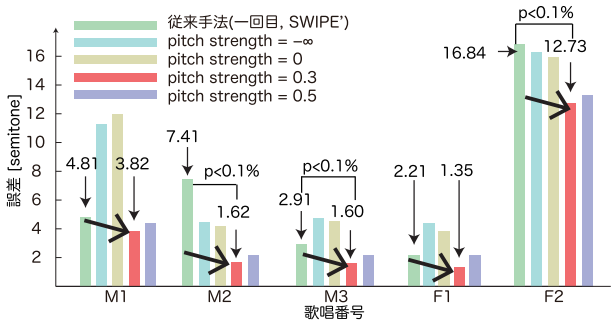

図 12 提案 F_0 推定手法（pitch strength = $-\infty$, 0, 0.3, 0.5）と従来手法（SWIPE'）の平均誤差 ϵ_f の比較（実験 B）

Fig. 12 Comparison of the average error ϵ_f by the proposed F_0 estimation method (pitch strength = $-\infty$, 0, 0.3, 0.5) and the conventional method (SWIPE').

差が減少した．しかし，pitch strength を 0 や $-\infty$（全フレームを使用）など 0.3 より低くすると誤りが増大することもあった．これは，提案手法のベースとして用いる F_0 推定手法が，ある程度雑音に頑健なことが求められることを示している．逆に pitch strength が 0.5 のときも推定精度が低下した．これは pitch strength を高くしすぎることによって，最頻 F_0 の推定に使用するフレーム数が減少し，最頻 F_0 の推定結果が不安定になったためと考えられる．

また，M1，F2 の歌声ではそれぞれ 3.82，12.73 semitone という大きな推定エラーが生じた．M1 については通常の伴奏と異なる楽器音が含まれていたことが原因であると考えられる．F2 については SWIPE' による最初の F_0 推定結果に定常的なオクターブエラーが生じたことが原因である．具体的には，提案手法では最頻 F_0 およびその ±1 オクターブの 3 つから，最初の推定結果に最も近いものを使用して推定範囲を制限するため，最初の F_0 推定結果のエラーに引きずられ提案手法が適切に機能しなかった．しかし，誤差自体が 4.11 semitone 減少しているのは，F2 の最初の推定結果がほぼ 40 semitone 付近に集中していて，1 オクターブ以上の誤差があったのに対し，提案手法では推定結果の誤差を 1 オクターブ程度に抑えられたためと考えられる．

5.3 実験 C：歌詞アラインメント手法の評価

オリジナル楽曲の発音開始時間を正解として，歌声コンテンツに対する発音開始時間の推定誤差をどの程度削減できたか評価した．このような評価とした理由は，提案手法が合唱制作インタフェースを実現するために，同一楽曲を歌った多数の歌声から「共通で利用可能な発音開始時間」を推定するためである．歌声コンテンツがすべて同一のオリジナル楽曲に基づいて派生したことを考慮すると，共通の発音開始時間はオリジナル楽曲の発音開始時間に近いと考えられる．

5.3.1 実験条件

F_0 推定の評価と同じオリジナル楽曲の読み（ひらがな）217 個（約 50 秒分に相当）に対して著者がスペクトログラムと実際の音を参照しながら手動で付与した発音開始時間を正解データとして使用した．提案手法で中央値の計算を行う際は，F_0 推定の評価と同じオリジナル楽曲における再生数上位 50 曲の歌声コンテンツ，およびオリジナル楽曲の計 51 曲の歌声を使用した．推定誤差の評価には，51 曲の歌声のうち，異なる歌詞を歌唱（替え歌）していない 37 曲を使用した．替え歌をしている歌唱 14 曲のうち 9 曲は半分以上の歌詞を変えており，3 曲はワンフレーズ程度の歌詞を変更していた．また，残りの 2 曲は歌詞は同じであるものの，本来は間奏である区間で歌唱していたため替え歌と見なした．本実験では替え歌も評価対象としたのは，実際の合唱制作において替え歌も重ねて使用することもある[*12]ためである．また，楽曲の一部の歌詞だけ変更していることもあり，制作者が必ずしも替え歌として認識していないことも考えられるため，替え歌に対して頑健に動作することは重要である．

推定誤差の大きさは読みごとの推定誤差と歌声ごとの平均絶対値推定誤差で評価した．s が歌手に対応するインデックス，i が読みに対応するインデックス，正解発音開始時間が $a_{org}(i)$，推定発音開始時間が $\bar{a}(s,i)$ で，歌声の数が S（$=37$），評価に用いる読みの数が I（$=217$）であるとき，読みごとの推定誤差 $\epsilon_a(s,i)$ と歌声ごとの平均絶対値推定誤差 $|\bar{\epsilon}_a(s)|$ は次式のように計算した．

$$\epsilon_a(s,i) = a_{org}(i) - \bar{a}(s,i) \tag{11}$$

$$|\bar{\epsilon}_a(s)| = \frac{1}{I}\sum_i |a_{org}(i) - \bar{a}(s,i)| \tag{12}$$

5.3.2 実験結果

従来手法と提案手法の読みごとの推定誤差 $\epsilon_a(s,i)$ のヒストグラムを**図 13** に示す．また，従来手法における歌声ごとの平均絶対値推定誤差 $|\bar{\epsilon}_a(s)|$ のヒストグラムを**図 14** に示す．提案手法の平均絶対値推定誤差は 89 ms であり，従来手法を用いて推定した 37 曲の歌声中 34 曲（91.9%）より正解データに対する平均絶対値推定誤差が小さかった．

5.3.3 考察

従来手法と提案手法の読みごとの推定誤差 $\epsilon_a(s,i)$ を示した図 13 より，提案手法は従来手法に比べて全体的に誤りが減少し，多くの読みで正解に対して ±0.1 秒以内の推定誤差に抑えられていることが分かる．また，外れ値という観点では，提案手法では 217 個の読みのうち 1 秒以上の絶対値推定誤差があったのは 3 個であったが，従来手法では 37 曲中 29 曲（78.4%）の歌声に 1 秒以上の絶対値推定誤差が 3 個以上存在した．このことは提案手法が大きな推

[*12] たとえば http://www.nicovideo.jp/watch/sm18301264 の動画における 2:55 から 3:04 の区間

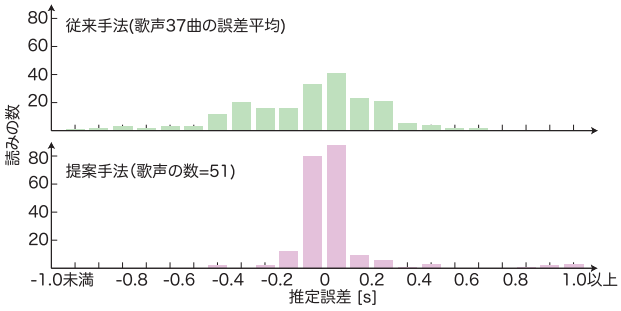

図 13 従来手法を 37 曲に適応した際の平均推定誤差のヒストグラムと，提案手法の読みごとの推定誤差 $\epsilon_a(s,i)$ のヒストグラム（実験 C）

Fig. 13 Histogram of the mean estimation error over 37 vocal covers by the conventional method and histogram of the estimation error $\epsilon_a(s,i)$ for each syllable by the proposed method (Experiment C).

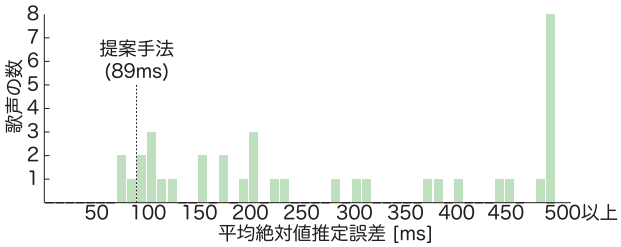

図 14 従来手法による歌声ごとの平均絶対値推定誤差 $|\bar{\epsilon}_a(s)|$ のヒストグラム（実験 C）

Fig. 14 Histogram of the mean absolute values of the estimation error $|\bar{\epsilon}_a(s)|$ by the conventional method (Experiment C).

定誤差を適切に補正できていることを示している．

また，歌声ごとの平均絶対値推定誤差 $|\bar{\epsilon}_a(s)|$ を示した図 14 より提案手法は歌詞アラインメントの誤りを減少させる手法として有効であることが確認できた．51 曲の中には，替え歌をしている歌声が 14 曲含まれていたが，多数の歌声の中央値を用いる提案手法により，替え歌を歌った歌声が含まれていても頑健に推定できていることが確認できた．ここで，替え歌を行っていない 37 曲のみを使用した場合も，提案手法の平均絶対値推定誤差は 89 ms であった．

本実験では 51 曲の歌声を使用したが，実用上はできるだけ少ない歌声で発音時間推定を行えることが望ましい．そこで，実験で使用した 51 曲の歌声の中からランダムに $N[1,51]$ 曲選択して，提案手法を適用するという操作をそれぞれの N に対して 100 回繰り返したときのオリジナル楽曲に対する平均絶対値推定誤差を調査した．その結果を**図 15** に示す．使用する歌声の数の増加にともなって，平均絶対値推定誤差が減少していることが分かる．

6. 今後の展望

本論文で対象とした 2 次創作コンテンツとしての合唱は，複数の歌声を同一楽曲の中で交互に聴くことで，それぞれの歌い方や声質の違いに気づきやすくなり，楽曲や歌声へ

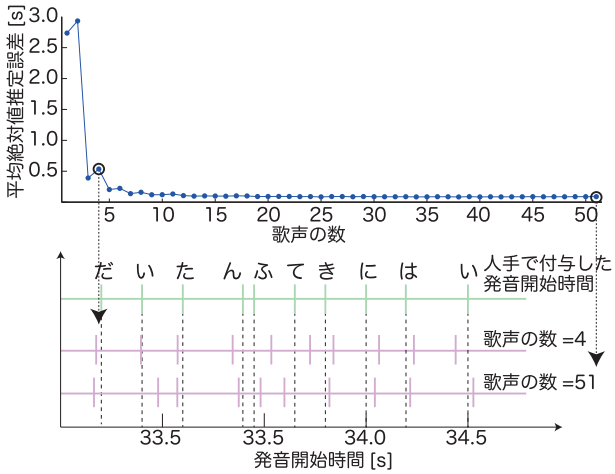

図 15 提案歌詞アラインメント手法において，歌声の数を変化させた際の平均絶対値推定誤差 $|\bar{\epsilon}_a(s)|$ の推移と，提案手法で歌声の数を 4 曲，51 曲にした場合の推定発音開始時間

Fig. 15 Transition of the absolute values of the average estimation error $|\bar{\epsilon}_a(s)|$ when changing the number of vocal covers, and the start time of each syllable estimated by the proposed method (the number of vocal covers is 4 and 51.

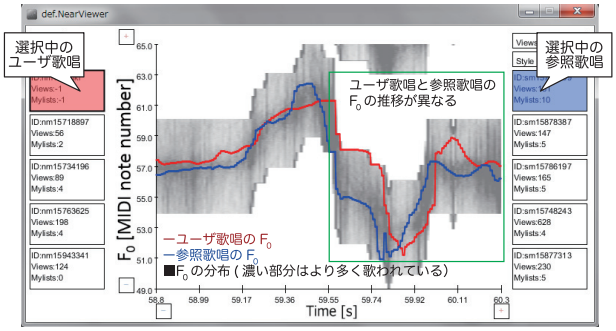

図 16 歌唱力向上支援インタフェースを用いた練習支援

Fig. 16 Support of training using the proposed vocal training interface.

の理解を深められる可能性がある．また歌手の視点からとらえると，自身の歌唱と他者の歌唱を合唱として聴くことで，それぞれの違いを理解しやすくなり，歌唱力の向上を支援できる可能性がある．本章では，このような合唱の持つ多様な可能性に着目した一例として，合唱を活用した歌唱力向上支援インタフェースについて説明する．従来，オリジナル楽曲の歌声と自分の歌声を比較できるインタフェース [20] は提案されてきたが，同一楽曲に対する複数の歌声を比較できるインタフェースは提案されてこなかった．

図 16 に歌唱力向上支援インタフェースを示す．本インタフェースではある歌唱（ユーザ歌唱）の F_0（赤）と比較したい別の歌唱の F_0（青），そして同一楽曲を歌唱した歌声における F_0 の出現頻度の分布（黒）をそれぞれ可視化する．分布の色が濃い部分はその F_0 で歌唱した歌手の人数が多いことを意味する．ここで，比較対象の歌唱には，ユーザ歌唱と声質や歌い回しが近い歌声（4.1.4 項で説明）

を推薦する機能（Unisoner における歌声の特徴に基づいた歌手検索機能（3.3 節）に近い）も持つ．さらに，再生数で推薦結果を絞り込むことができ，自分の声に類似していて人気が高い歌声を参考にして練習できる．

図 16 から，ユーザ歌唱と比較対象の F_0 には違いがあること，参照歌唱（青）の F_0 が分布の中央付近（色が濃い部分）を通っていることが確認できる．そのため，参照している歌唱の方がより一般的な歌い方に近いといえる．このようにユーザは，自身の歌唱と参照歌唱もしくは一般的な歌い方との違いに気づいたり，自身の歌声の F_0 や分布を確認しながら音として確認できる．

7. おわりに

本論文では，合唱制作支援インタフェース Unisoner，F_0 推定の誤り削減手法，歌詞アラインメントの誤り削減手法をそれぞれ提案した．Unisoner は歌詞のクリックや歌手アイコンのドラッグなどの簡単な操作だけで手軽に合唱を制作できるインタフェースである．また，F_0 推定および歌詞アラインメントの誤り削減手法は，個々の歌声に対する推定結果に誤差が含まれていても，他の歌声に対する推定結果を活用して推定結果を修正することができる．

本論文で提案したインタフェースおよび信号処理技術について有効性の検証を行った結果，Unisoner については，指定した合唱を制作するというタスクを遂行する時間を測定し，従来ツールと比べて効率的に合唱を制作できることを示した．また，F_0 推定および歌詞アラインメントについては，それぞれ従来手法と比較して推定誤差が減少したことを確認した．

謝辞 本論文の一部は，科学技術振興機構 OngaCREST プロジェクトによる支援を受けました．また，ニコニコ動画上の合唱動画を扱うために濱崎雅弘氏，石田啓介氏にご協力いただきました．感謝いたします．

参考文献

[1] Davies, M., Hamel, P., Yoshii, K. and Goto, M.: AutoMashUpper: An Automatic Multi-Song Mashup System, *Proc. ISMIR 2013*, pp.575–580 (2013).
[2] 宮島 靖：Music Mosaic Generator：高精度時系列メタデータを利用した音楽リミックスシステム，WISS 2007 論文集，pp.13–18 (2007).
[3] Tokui, N.: Massh! – A Web-based Collective Music Mashup System, *Proc. DIMEA 2008*, pp.526–527 (2008).
[4] Nakano, T., Murofushi, S., Goto, M. and Morishima, S.: DanceReProducer: An Automatic Mashup Music Video Generation System by Reusing Dance Video Clips on the Web, *Proc. SMC 2011*, pp.183–189 (2011).
[5] Fujihara, H., Goto, M., Ogata, J. and Okuno, H.G.: LyricSynchronizer: Automatic Synchronization System Between Musical Audio Signals and Lyrics, *IEEE J. Selected Topics in Signal Processing*, Vol.5, No.6, pp.1251–1261 (2011).
[6] Nakano, T. and Goto, M.: VocaRefiner: An Interactive Singing Recording System with Integration of Multiple Singing Recordings, *Proc. SMC 2013*, pp.115–122 (2013).
[7] Hamasaki, M., Goto, M. and Nakano, T.: Songrium: A Music Browsing Assistance Service with Interactive Visualization and Exploration of a Web of Music, *Proc. WWW 2014*, pp.523–528 (2014).
[8] Boll, S.F.: Suppression of Acoustic Noise in Speech Using Spectral Subtraction, *IEEE Trans. ASSP*, Vol.27, No.2, pp.113–120 (1979).
[9] Rubner, Y., Tomasi, C. and Guibas, L.J.: The earth mover's distance as a metric for image retrieval, *International J. Computer Vision*, Vol.40, No.2, pp.99–121 (2000).
[10] Kako, T., Ohishi, Y., Kameoka, H., Kashino, K. and Takeda, K.: Automatic Identification for Singing Style Based on Sung Melodic Contour Characterized in Phase Plane, *Proc. ISMIR 2009* (2009).
[11] 大石康智，後藤真孝，伊藤克亘，武田一哉：スペクトル包絡と基本周波数の時間変化を利用した歌声と朗読音声の識別，情報処理学会論文誌，Vol.47, No.6, pp.1822–1830 (2006).
[12] Chih-Chung, C. and Chih-Jen, L.: LIBSVM: A library for support vector machines, *ACM Trans. Intelligent Systems and Technology*, Vol.2, No.3, pp.1–27 (2011).
[13] Wu, T.-F., Lin, C.-J. and Weng, R.C.: Probability Estimates for Multi-class Classification by Pairwise Coupling, *J. Machine Learning Research*, Vol.5, pp.975–1005 (2004).
[14] Schuller, B., Kozielski, C., Weninger, F., Eyben, F. and Rigoll, G.: Vocalist Gender Recognition in Recorded Popular Music, *Proc. ISMIR 2010*, pp.613–618 (2010).
[15] Vogt, T. and André, E.: Improving automatic emotion recognition from speech via gender differentiation, *Proc. LREC 2006* (2006).
[16] De Cheveigné, A. and Kawahara, H.: YIN, a fundamental frequency estimator for speech and music, *J. Acoustical Society of America*, Vol.111, No.4, pp.1917–1930 (2002).
[17] Camacho, A.: SWIPE: A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music, Ph.D. Thesis, Univ. of Florida (2007).
[18] Camacho, A.: Detection of Pitched/Unpitched Sound using Pitch Strength Clustering, *Proc. ISMIR 2008*, pp.533–537 (2008).
[19] Welch, B.L.: The generalization of 'student's' problem when several different population variances are involved, *Biometrika*, Vol.34, No.1/2, pp.28–35 (1947).
[20] Nakano, T., Goto, M. and Hiraga, Y.: MiruSinger: A Singing Skill Visualization Interface Using Real-Time Feedback and Music CD Recordings as Referential Data, *Proc. ISM 2007 Workshops*, pp.75–76 (2007).

都築 圭太

2013 年筑波大学情報学群情報科学類卒業．2015 年筑波大学大学院システム情報工学研究科博士前期課程を修了．修士（工学）．

中野 倫靖 （正会員）

2008 年筑波大学大学院図書館情報メディア研究科博士後期課程修了．博士（情報学）．現在，産業技術総合研究所主任研究員．日本音響学会会員．2009 年情報処理学会山下記念研究賞（音楽情報科学研究会），2013 年 Sound and Music Computing Conference（SMC2013）The Best Paper Award 等各受賞．

後藤 真孝 （正会員）

1998 年早稲田大学大学院理工学研究科博士後期課程修了．博士（工学）．現在，産業技術総合研究所情報技術研究部門首席研究員兼メディアインタラクション研究グループ長．IPA 未踏 IT 人材発掘・育成事業プロジェクトマネージャー，情報処理学会理事等を兼任．日本学士院学術奨励賞，日本学術振興会賞，ドコモ・モバイル・サイエンス賞基礎科学部門優秀賞，科学技術分野の文部科学大臣表彰若手科学者賞，情報処理学会長尾真記念特別賞，星雲賞等，42 件受賞．

山田 武志 （正会員）

1999 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（工学）．同年筑波大学講師．現在，同准教授．音声認識，音環境理解，多チャネル信号処理，メディア品質評価，e ラーニングの研究に従事．IEEE，電子情報通信学会，日本音響学会，日本言語テスト学会各会員．

牧野 昭二

1981 年東北大学大学院修士課程修了．同年日本電信電話公社入社．博士（工学）．以来，NTT 研究所において，電気音響変換器，音響エコーキャンセラ，ブラインド音響分離等の音響信号処理の研究に従事．現在，筑波大学生命領域学際研究センター教授．IEEE Distinguished Lecturer．IEEE Fellow．電子情報通信学会 Fellow．