歌声・調波楽器音・打楽器音分離とユーザ演奏のリアルタイム可視化に基づく音楽演奏練習システム

土橋 彩香[1,a)]　池宮 由楽[1,b)]　糸山 克寿[1,c)]　吉井 和佳[1,d)]

概要：本稿では，任意の音楽音響信号に対してカラオケ音源やマイナスワン音源を自動生成し，それら伴奏に合わせてユーザの歌唱・楽器演奏をリアルタイムで解析・可視化する音楽演奏練習システムについて述べる．本システムは音量調節・音楽内容表示・演奏認識の三つの機能からなり，どこでも手軽に利用できるようタブレット端末上に実装されている．ユーザが音楽に合わせて歌唱や楽器を練習するとき，各楽器パート（ボーカル・ギターやキーボードなどの調波楽器・ドラム）の音量を個別に抑制でき，原曲の音楽内容（ビート時刻・コード進行・歌声の音高）は手本として同期スクロール表示される．さらに，練習を効果的に行うため，ユーザ自身の歌声の音高やギター演奏のコードはリアルタイムで認識され，手本と比較表示される．これらの機能を利用するには，あらかじめ歌声・調波楽器音・打楽器音の分離と音楽内容解析を行うことが必要となる．この問題を解決するため，ロバスト主成分分析に基づく歌声・伴奏音分離とスペクトログラムの異方性に基づく調波楽器音・打楽器音分離を組み合わせる手法を提案する．また，原曲の音楽内容は，クラウドソーシング型音楽鑑賞 Web サービス Songle による解析結果と連動させる．これにより，誤りを修正したら他ユーザと共有したり，タブレット上に最新結果を反映することができる．被験者実験の結果，提案システムは三つの機能ともに，音楽練習に有益であることが示された．

A Music Performance Practice System based on Vocal, Harmonic, and Percussive Source Separation and Real-time Visualization of User Performances

Ayaka Dobashi[1,a)]　Yukara Ikemiya[1,b)]　Katsutoshi Itoyama[1,c)]　Kazuyoshi Yoshii[1,d)]

Abstract: This paper presents a music performance practice system that can generate karaoke or minus-one versions of arbitrary music recordings and analyze and visualize user performances in real time. This system is implemented on a tablet computer for mobility and consists of three functions: instrument volume control, music content visualization, and user performance analysis. When a user sings a song or plays a musical instrument, the user is allowed to control the volume of each instrument group (vocal, harmonic instruments, and drums) and musical contents (beat times, chord progressions, and vocal pitches) of original recordings are visualized in synchronization with the music playback. To help a user effectively practice the musical performance, the pitches of user's singing voices and the chords of user's guitar performances are analyzed in real time and visually compared with original ones. This system needs to perform vocal, harmonic, and percussive source separation and automatic content analysis in advance. To solve this problem, we propose a source separation method that combines vocal-and-accompaniment source separation based on robust principal component analysis with harmonic-and-percussive source separation based on spectral anisotropy. In addition, we link the musical contents of original recordings with those analyzed by a crowd-sourcing Web service called Songle. This enables us to reflect the latest content analysis results of Songle to the system. The subjective experiment showed the effectiveness of the main three functions.

[1] 京都大学 大学院情報学研究科
Graduate School of Informatics, Kyoto University
[a)] dobashi@kuis.kyoto-u.ac.jp
[b)] ikemiya@kuis.kyoto-u.ac.jp（2015 年 3 月修士課程修了）
[c)] itoyama@kuis.kyoto-u.ac.jp
[d)] yoshii@kuis.kyoto-u.ac.jp

1. はじめに

単に聴くだけだけではなく，歌ったり演奏したりすることは，音楽の楽しみ方の一形態である．好きな楽曲に合わせて歌うためには，カラオケ施設を利用するのが一般的である．しかし，そこでの伴奏音は MIDI 音源を用いて機械的に合成されたものであることが多く，市販 CD に収録されているプロ歌唱に付随する伴奏音とは音色や品質が異なることから，臨場感に欠ける場合があった．一方，もっと手軽に場所を問わずにカラオケを楽しんだり，皆とカラオケに行く前に練習するためには，市販 CD に収録されているカラオケ音源がしばしば利用される．しかし，カラオケ音源が収録されている楽曲は，有名なシングル CD をはじめとして全体のごく一部であった．もしカラオケ音源が利用可能だとしても，プロ歌唱を含む音源には収録されているコーラス歌唱などもすべて除去されており，エフェクトや楽器構成などが異なる場合も多かった．

好きな楽曲に合わせてギターやキーボード，ドラムなどの練習を行うには，マイナスワン音源（原曲から特定の楽器パートのみを除去したもの）があると有益である．例えば，ボーカル・ギター・ドラムからなるバンドが練習を行う際，欠席したメンバーのパートのみを伴奏音として再生できれば便利である．最近では，伴奏として原曲をそのまま再生し，卓越したソロ演奏・歌唱を録画して動画共有サービスに投稿するアマチュア演奏家が多数存在する（「演奏してみた」「歌ってみた」動画と呼ばれる）．その際，担当パートとの衝突を避けるために，原曲の音量を小さくする必要がなくなれば，より完成度の高い作品を投稿することが可能になる．一方，路上パフォーマンスとして伴奏つきでソロ演奏を披露する状況も考えられるため，手軽に持ち運びできるマイナスワン音源再生装置が望まれる．

本論文では，任意の音楽音響信号（市販 CD に収録されているような歌唱を含む完全な楽曲）に対して，カラオケ音源やマイナスワン音源を自動生成し，ユーザの歌唱や楽器の練習を支援するシステムについて述べる（図 1）．本システムは Android タブレット上に実装されており，どこでも手軽に音楽を楽しむことができる．ユーザが歌う際には，原曲中に含まれるコーラス歌唱を残しつつ，メインボーカルだけを抑制したカラオケ音源を再生することができる．このとき，原曲中のプロ歌手の歌声の音高が画面上にガイドとして表示されており，ユーザはそれを参考にしながら歌うことができる．一方，ユーザの歌声の音高はリアルタイムで解析・可視化され，自分の歌声とプロ歌唱の音高を比較しながら反省できる．また，ユーザがギターやキーボードなどの調波楽器，あるいはドラムなどの打楽器を弾く際には，カラオケモードと同様に，調波楽器音あるいは打楽器音の音量を個別に抑制することができる．このとき，原曲のビート時刻やコード進行は音楽再生と同期してスクロール表示され，調波楽器を演奏する際にはそのコードがリアルタイムで解析・可視化される．

図 1 システムの使用例：ユーザは，画面上にスクロール表示されるビート時刻とコード進行を見ながら，他パート（ボーカル・ドラム）の再生に合わせてキーボードを演奏している．

Fig. 1 Usage example: A user is playing a keyboard with the playback of the other instrument parts (vocal and drums) while seeing the beat times and chord progressions scrolled on the screen.

本システムを実現するには，歌声・調波楽器音・打楽器音分離，音楽内容の可視化，リアルタイム演奏認識に取り組む必要がある．まず，各パートの音源分離を行うため，ロバスト主成分分析 (Robust Principal Component Analysis: RPCA) に基づく歌声・伴奏音分離 [1] とスペクトログラムの異方性に基づく調波楽器音・打楽器音分離 [2] を組み合わせる手法を提案する．原曲の音楽内容を取得するには，クラウドソーシング型音楽鑑賞 Web サービス Songle [3] を用いる．本サービスでは，Web 上にある任意の音楽音響信号のビート時刻・コード進行・歌声の音高・楽曲構造などの要素を自動解析しブラウザ上に表示できるだけではなく，SongleWidget と呼ばれる API を用いて解析結果を取得可能である．自動解析には誤りが含まれるため，Wikipedia と同様に，ユーザは解析誤りを Web 上で訂正でき，その結果は他ユーザと共有される．この Web サービスを用いることで，提案システムは音楽内容の逐次アップデートが可能になる．また，ユーザ演奏のリアルタイム解析のため，テンプレートに基づくコード認識と，Subharmonic Summation [4] に基づく歌声の音高推定を用いる．これらの手法は簡便ではあるが，計算資源の限られた携帯端末上でも軽快な動作が可能である．

本論文の構成は以下の通りである．2 章で提案システムのユーザインタフェースの設計を，3 章で実現方法をそれぞれ説明する．4 章で被験者実験について報告する．5 章で関連研究を紹介し，6 章でまとめとする．

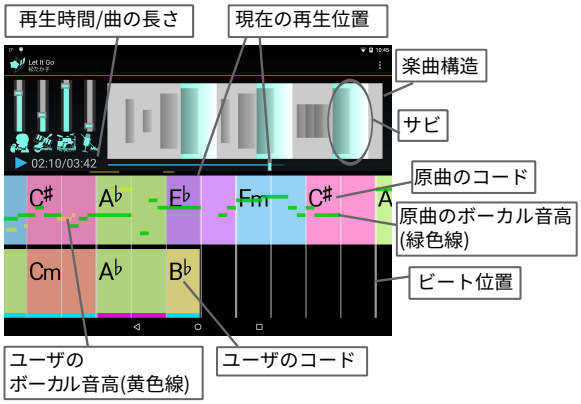

図 2　提案システムのスクリーンショット．
Fig. 2　A screenshot of the proposed system.

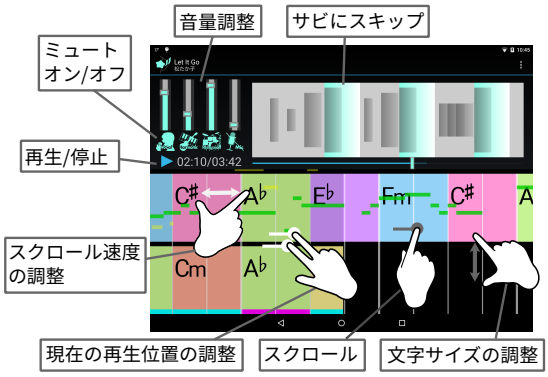

図 3　提案システムの操作方法．
Fig. 3　How to use the proposed system.

2.　ユーザインターフェースの設計と機能

本章では，Android タブレット (HTC Nexus 9) 上に実装した提案システムのユーザインタフェースについて述べる．図 2 に画面内の要素，図 3 に操作方法を示す．本システムは主に，分離音源の音量調節，音楽内容表示，ユーザの演奏認識の三つの機能を有している．これらの機能は，歌唱や楽器の練習を効果的に行う上で有用である．

ユーザインタフェース設計の際には，迷いなく操作できる標準的なインタフェースを用いながらも，新鮮なインタラクティブ体験が得られるように工夫した．例えば，音量調節機能に関しては，縦型スライダを並行に配置し，マルチトラックレコーディング用のミキサと同様の操作を実現した．ユーザは，不可逆にミキシングされているはずの混合音を，あたかもミキシングエンジニアのように操作し，その結果をリアルタイムに視聴できる．ここで，インタフェースの操作方法や効果は十分予測できるにもかかわらず，不思議なインタラクティブ体験がもたらされるのが重要な点である．音楽内容表示やユーザの演奏認識についても同様である．原曲とユーザ演奏の解析結果を音楽再生に合わせてスクロールしながら比較表示するのは，音楽ゲームなどで一般的なインタフェースである．しかし，任意の楽曲に対して，実楽器（電子デバイスではない）を用いて普段通り演奏しても音楽ゲームが可能になることから，音楽ゲームを楽しむと同時に，あたかも音楽ゲームを製作しているかのような不思議な感覚がもたらされる．

上記の議論から，本システムは，能動的音楽鑑賞 (5 章)，すなわち，音楽の演奏・制作・鑑賞の総合的な楽しさを提供するものである．Songle から取得した音楽内容をそのままタブレット上に表示できるため，解析結果訂正機能の省かれたいわば軽量版 Songle として楽しめる．このとき，Songle にはない音量調節機能を用いることで，同期表示される音楽内容を見ながら特定のパートを集中して聞いたり，楽器パートの音量バランスを自分好みにイコライズすることにより，より深く音楽を楽しめるようになる．

2.1　音量調節機能

ユーザは，原曲中の歌声（メインボーカルのみ），調波楽器音（コーラスの歌声を含む），打楽器音の各音量を独立に調節できる．画面左上に各パートに対応した三本の音量スライダがある．その右のスライダでは，マイク入力の音量の閾値を変更できる（図 2）．ユーザが歌唱あるいは演奏したいパートの音量を下げることにより，システムは原曲のカラオケ・マイナスワンバージョンを再生する．この機能は練習のときに有用である．例えば，歌の練習をするとき，歌声の音量を下げることができる．歌いながらギターを弾きたいときは歌声と調波楽器音の音量を下げればよい．バンドメンバーの都合が合わない場合は，欠けたメンバーのパートを代わりに再生することができる．

2.2　音楽内容表示機能

画面右上部には，階層的な楽曲構造が表示されている．左から右へ時系列順に並んでいる同じ大きさの四角形は同じ構造の繰り返しを表す．中でも，水色の四角形はサビを表していて，これをタップするとそれぞれのサビの開始時刻に再生位置をジャンプできる．

画面下部に，音楽の再生に同期してビート時刻やコード進行がスクロール表示される．上段に原曲のコード進行が表示されるので，ギターやキーボードを演奏する場合は手本を見ながら演奏できる，また，原曲の歌声の音高は緑色の軌跡で表示されており，伴奏に合わせて歌う場合も手本を参照しながら歌うことができる．

コードの表示幅は水平方向のピンチイン・アウトで変更する．表示幅を縮めると音楽内容を俯瞰しやすくなり，広げると音高やビート時刻が見やすくなる．コードの文字サイズは垂直方向のピンチイン・アウトで変更する．現在の再生位置の左側の再生済みの領域は暗めで表示され，その部分を二本指で水平方向にスワイプすると，領域の大きさを変更できる（画面内再生位置の変更）．この領域の大小を調節すれば，ユーザ演奏の認識結果を確認しやすくなったり，先の歌声の音高やコードを確認できるようになる．

2.3 演奏認識機能

画面下部の上段には原曲のコード進行（Songle から取得）が表示されているのに対して，下段にはマイク入力からリアルタイムで認識したコードが表示・記録されている．この機能により，ユーザはギターやキーボードを演奏する際に，正しく弾けているかあとで振り返って確認することができる．現在のところ，認識できるコードの種類は maj と min の二種類のみであり，ルート音が 12 種類あるので，合計 24 種類のコードを認識できる．

また，緑色の軌跡は原曲の歌声の音高（Songle から取得）を表しているのに対して，黄色の軌跡はマイク入力からリアルタイムで認識した音高を示している．この機能により，ユーザはカラオケを楽しむ際に，原曲のプロ歌唱と自分の歌唱の差を視覚的に確認することができる．同様の比較機能は，歌唱力向上インタフェース Mirusinger [5] でも提案されており，詳細な音高軌跡を表示することで，ビブラートやグリッサンドなどの歌唱表現の訓練を行うことができる．一方，本システムでは，歌声の音高は半音単位にあらかじめクオンタイズされており，各音符を正しい音高で歌うことに主眼を置いている．

3. システム実装

本章では，高度な信号処理に基づく提案システムの内部実装について説明する．各音源の音量調節機能は歌声・調波楽器音・打楽器音分離によって実現し，音楽内容はクラウドソーシング型音楽解析サービス Songle から取得する．さらに，リアルタイムでの歌声の音高推定やギター・キーボード演奏に対するコード認識を行う．

3.1 音源分離

音楽音響信号を歌声・調波楽器音・打楽器音に分離するには，まず音楽音響信号を歌声とその他の伴奏音に分離し，後者を更に調波楽器音と打楽器音に分離する（図 4）．歌声・調波楽器音・打楽器音の音源分離は，あらかじめデスクトップ型の計算機上で行うものとする．ただし，上記の計算量は比較的小さいため，携帯端末の CPU の性能向上に伴い，将来的にはタブレット上でのスタンドアロン動作が可能になると考えられる．

3.1.1 歌声・伴奏音分離

歌声・伴奏音分離には，RPCA と歌声の音高推定手法を組み合わせる手法 [1] を用いる（図 5）．本手法は，2014 年の国際的な音楽解析コンテストである MIREX において，歌声分離トラックで最も優れた分離性能を達成している．まず，RPCA により音楽音響信号のスペクトログラムを歌声にあたるスパース行列と伴奏にあたる低ランク行列の和に分離する．次に，得られた二つの行列を要素ごとに比較してバイナリマスクを作成し，入力スペクトログラムに適用することで歌声スペクトログラムを取得する．

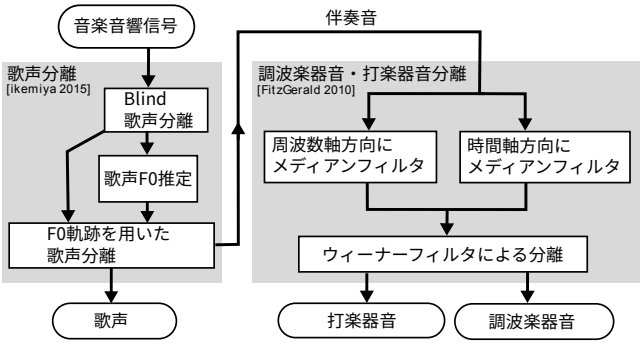

図 4 歌声・調波楽器音・打楽器音分離．
Fig. 4 Vocal, harmonic, and percussive source separation.

次に，Subharmonic Summation (SHS) [4] により歌声の音高軌跡を推定する．具体的には，対数周波数軸において，倍音強度を足しこんだ Salience 関数 $H(t,s)$ を計算する．

$$H(t,s) = \sum_{n=1}^{N} h_n P(t, s + 1200 \log_2 n) \tag{1}$$

ここで，t は時間フレームのインデックス，s は対数周波数 [cents]，$P(t,s)$ は時刻 t・周波数 s での振幅，N は想定する倍音成分の数，h_n は重みを表している．上記の Salience 関数の和を最大化する滑らかな歌声音高軌跡 $\hat{S}$ は，以下の最適化問題をビタビアルゴリズムで解くことで得られる．

$$\hat{S} = \arg\max_{S_1,\ldots,S_T} \sum_{t=1}^{T-1} \{\log a_t H(t,s_t) + \log T(s_t, s_{t+1})\} \tag{2}$$

ここで，$T(s_t, s_{t+1})$ は，現在の音高 s_t から次の音高 s_{t+1} への遷移確率（適切に設定）であり，a_t は正規化係数である．

最後に，与えられた音高の倍音成分のみを通すバイナリマスクを作成し，RPCA マスクと調波マスクを統合する．この統合マスクを入力スペクトログラムに適用することで最終的な歌声スペクトログラムを得る．

3.1.2 調波楽器音・打楽器音分離

調波楽器音・打楽器音分離には，スペクトログラムの異方性に着目し，メディアンフィルタに基づく手法を用いる（図 6）．具体的には，調波楽器音はスペクトログラム中で時間軸方向に滑らかであり，打楽器音はスペクトログラム中で周波数軸方向に滑らかであることに着目する．したがって，調波楽器音を抽出したい場合は，時間軸方向のメディアンフィルタを，打楽器音を抽出したい場合は周波数軸方向のメディアンフィルタを適用すればよい．

入力の振幅スペクトログラムを $\boldsymbol{W}$，調波楽器音の振幅スペクトログラムを $\boldsymbol{H}$, 打楽器音の振幅スペクトログラムを $\boldsymbol{P}$ とすると，$\boldsymbol{W} = \boldsymbol{H} + \boldsymbol{P}$ が成立する必要がある．まず，$\boldsymbol{W}$ に時間軸方向・周波数軸方向のメディアンフィルタをそれぞれ適用することで，暫定的なスペクトログラム $\boldsymbol{H}$ と $\boldsymbol{P}$ が得られる．しかし，このままでは $\boldsymbol{W} = \boldsymbol{H} + \boldsymbol{P}$ を満たさないので，ウィーナーフィルタのためのソフトマスクを次式によって計算する．

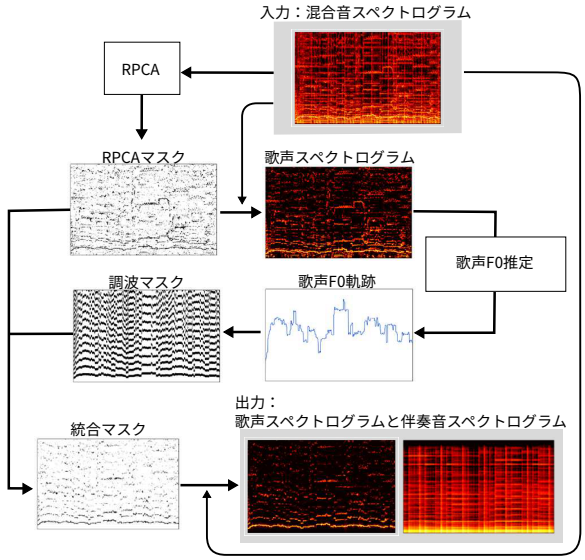

図 5　歌声・伴奏音分離.

Fig. 5　Vocal and accompaniment source separation.

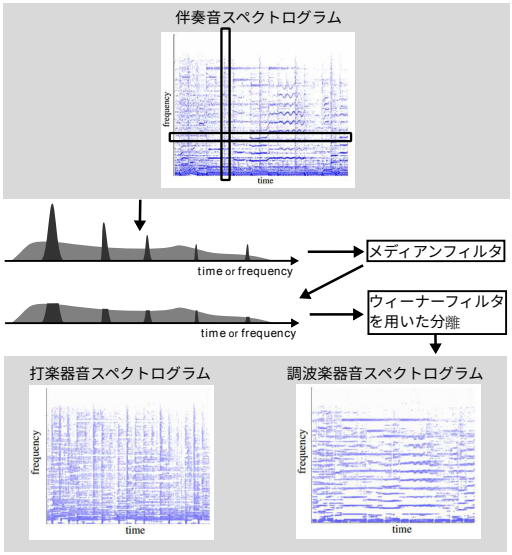

図 6　調波楽器音・打楽器音分離.

Fig. 6　Harmonic and percussive source separation.

$$\boldsymbol{M_H} = \frac{\boldsymbol{H}^p}{\boldsymbol{H}^p + \boldsymbol{P}^p} \qquad \boldsymbol{M_P} = \frac{\boldsymbol{P}^p}{\boldsymbol{H}^p + \boldsymbol{P}^p} \tag{3}$$

ここで，全ての計算は要素ごとに行うものとし，p は個々の要素を p 乗することを意味する．最終的なスペクトログラム $\boldsymbol{H}$ と $\boldsymbol{P}$ は以下の通り計算できる．

$$\boldsymbol{H} = \boldsymbol{W} \otimes \boldsymbol{M_H} \qquad \boldsymbol{P} = \boldsymbol{W} \otimes \boldsymbol{M_P} \tag{4}$$

ここで $\otimes$ は要素ごとの乗算を表す．

3.2　音楽内容表示

表示する音楽内容は，能動的音楽鑑賞サービス Songle から SongleWidget と呼ばれる Web API を用いて取得する．Songle では Web 上の任意の楽曲を自動解析し，繰り返し構造・ビート時刻・コード進行・歌声の音高を表示する．ユーザの誤り訂正により，解析の精度は徐々に向上するに伴い，タブレット上のデータも逐次更新される．

3.3　演奏認識

本システムでは，ユーザの歌声の音高や演奏したコードをリアルタイムで認識する．マイクまたはライン入力を想定しているので，入力音響信号中では，認識したい歌声または楽器音の音量が，他の音より優勢になると考えられる．そのため，複雑な混合音を想定していない単純な手法でもある程度精度良く推定できることが期待される．

ユーザの歌声の音高推定には，歌声・伴奏音分離と同様に SHS を用いる．リアルタイム処理を行うため，フレームごとに独立に Salience 関数を計算し，その最大値を探索することで音高を推定する．ただし，音高軌跡を平滑化するため，前フレームの音高を平均値としたガウス関数を Salience 関数に掛け合わせる．推定された音高は，直ちに画面上に黄色の軌跡として表示・記録されていく．

ユーザの演奏のコード推定には，単純なテンプレートマッチングを用いた．フレーム毎に音響信号のクロマベクトルを計算し，あらかじめ準備した 24 種類のテンプレートとコサイン距離を計算することにより，最も距離の近いコードに同定を行う．Songle から取得した正解コードの各区間において，再生位置が区間に入ると同時にクロマベクトルを足し合わせていくとともに，逐次コード推定を行い，推定結果を画面上に反映させる．したがって，区間の終わりに近づくほど解析結果が安定することになる．

4.　評価実験

本章では，提案システムの有効性を評価するために行った被験者実験について述べる．

4.1　実験条件

ギター練習や歌唱の練習支援に関する実験を行った．実際の使用シーンを想定し，携帯性を重視して最低限の機器セットであるタブレット端末・スピーカ・マイクを使用した．ギター演奏時の配置を図 7 に示す．音楽はスピーカ 1 から．ギターの演奏音はスピーカ 2 から出力され，後者はマイクで録音される．歌唱時の配置を図 8 に示す．音楽はスピーカから出力され，ユーザの歌声はマイクで録音される．歌唱時には，被験者にコード付き歌詞を配布した．

被験者は事前にシステムの操作方法の教示を受け，一曲あたり 8 分間，システムの機能を自由に使いながらギター演奏・歌唱を行い，全曲終了後にアンケートに回答した．使用曲は中島みゆき「地上の星」，スピッツ「チェリー」，松たか子「レット・イット・ゴー ～ありのままで～」の三曲であり，曲順は被験者ごとにランダムに設定した．アンケートでは，インタフェース全体の操作性がどうであったか，{ 音量調整機能・演奏認識機能 } は { 必要だったか・

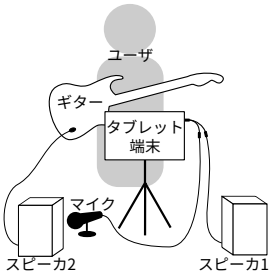

図 7 ギター演奏時の配置.
Fig. 7 Guitar performance.

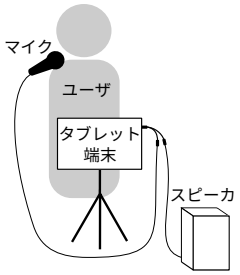

図 8 歌唱時の配置.
Fig. 8 Singing.

スキル向上に役立ったか・機能を楽しめたか } について 9 段階評価（1 から 9）で，適切だと思う音量設定を元の音量に対する割合 [%] で回答させた．また，感想・要望・気づいたことなどを自由記述させた．

被験者は研究室外の 20 代男性 4 人であり，うち 3 人はギタリスト，ボーカリスト両方，1 人はギタリストとしてのみ実験に参加した．4 人は 2 年間から 9 年間のギター演奏歴があり，ボーカリストとして参加した 3 人は月二，三度から週二，三度カラオケを利用していた．

4.2 実験結果

ギター演奏のためにシステムを使用した際のアンケート結果を表 1，歌唱のためにシステムを使用した際のアンケート結果を表 2 に示す．

音量調節機能については，ギター演奏・歌唱のいずれにおいても，機能の必要性・スキル向上における有用性・楽しさの観点で有効性が示された．全体的に，歌唱時の方がスコアが高い傾向が見られた．自由記述から，自分のパートを置き換えるような使い方のほか，歌声だけを残してギターを合わせ，弾き語りのようにするなど目的に応じて各音源音量を調節できることは有益であるという意見が複数得られた．伴奏音を消すことで，同じキーの違うコードを弾くなどのアレンジができるという意見が得られた一方，ギター演奏時の伴奏音および歌唱時の歌声の音量の平均がそれぞれ 56%，33%あることから，ユーザの理解度や用途によっては歌声や特定のパートを完全にカットしてしまうよりも多少残っている方が利用しやすいことが示された．伴奏音はもう少し細分化してほしいという意見も得られた．例えば，ギター練習用のマイナスワン音源を作成しようとすると，複数の調波楽器の音量が抑制され，音量バランスが悪くなる楽曲も存在する．技術的難度とタブレット端末という画面サイズや計算リソースの制約を考慮して，適切なパート設定については今後検討したい．

音楽内容表示については，ギター演奏・歌唱のいずれにおいても，手本として参考になると評価された．ただし，今回利用した三曲ともに，自動解析誤りの修正がほぼ完全に完了したものであったので，手本に誤りが含まれる場合の評価は今後の課題である．また，ビート時刻・コード進行・歌声の音高に加えて，歌詞を同期表示してほしいという意見が得られた．これを実現するには，SongPrompter [6] のように，混合音中に含まれる歌声に対して歌詞を自動的に同期する技術を応用することが考えられる．

演奏認識機能についても，ギター演奏・歌唱のいずれにおいても，機能の必要性・スキル向上における有用性・楽しさの観点で有効性が示された．コード認識時の境界は任意にして，音高表示の時間幅はもう少し大きくした方がいいという意見が得られた．本システムでは原曲のコードの区間ごとにユーザのコードが推定される．そのため，原曲の通りに弾く場合は望ましいが，アレンジを加えるときにはコードの切り替わりが必ずしも原曲と同じにはならない．音高軌跡については，クオンタイズや平滑化すると見やすくなる一方で，ビブラートのような表現が可視化されなくなるトレードオフが考えられる．用途によって認識の時間ユニットの切り替える機能が考えられる．

認識精度は弾き方（コードの抑え方）や歌い方に左右され，個人差が大きかった．認識誤りが起こりやすいコードがあるという指摘もあり，スキル向上に用いるにはまだ改善の余地があるものの，ゲーム感覚で楽しむには十分であると考えられる．また，リズムを認識できるとより楽しく，有益であるという意見を得られた．今後の課題としては，ドラム演奏に関する認識機能の実装が考えられる．

表 1 アンケート結果：ギター練習支援.
Table 1 Questionnaire results of guitar performace assistance.

質問事項			平均	標準偏差
インタフェース全体の操作性			6	2
音量調節機能	適切だと思う音量設定 [%]	歌声	95	10
		伴奏音	56	35
		打楽器音	95	10
	この機能は必要だった		8	1
	スキル向上に役立った		7	2
	この機能を楽しめた		8	1
演奏認識機能	この機能は必要だった		8	1
	スキル向上に役立った		7	3
	この機能を楽しめた		8	1

表 2 アンケート結果：歌唱練習支援.
Table 2 Questionnaire results of singing assistance.

質問事項			平均	標準偏差
インタフェース全体の操作性			7	2
音量調節機能	適切だと思う音量設定 [%]	歌声	33	21
		伴奏音	100	0
		打楽器音	67	58
	この機能は必要だった		9	0
	スキル向上に役立った		7	1
	この機能を楽しめた		9	1
演奏認識機能	この機能は必要だった		8	1
	スキル向上に役立った		7	2
	この機能を楽しめた		8	1

5. 関連研究

本章では伴奏音生成，音楽鑑賞・創作・演奏，音源分離の三つの観点から関連研究について述べる．

5.1 伴奏音生成

自動伴奏システムの歴史は古く，30 年前ほど前にはすでに，伴奏パートの楽譜情報（MIDI データ）を使用する自動伴奏システム [7,8] が開発されている．Tekin ら [9] や Pardo ら [10] は，一定の範囲内での弾き直しや弾き飛ばし，さらにはテンポの揺らぎに対応しながら伴奏を同期再生できる楽譜追従システムを提案している．最新の研究成果として，中村ら [11,12] は，任意箇所の弾き直し・任意箇所へのジャンプや，楽譜上には音符列が正確に定義されていないトリルや分散和音に対しても頑健に追従できる自動伴奏システム Eurydice を開発している．

上記のような MIDI データに基づく自動伴奏システムとは異なり，高品質な伴奏の音楽音響信号をユーザの演奏に同期させる試み [13,14] もなされているが，ピアノのような多声楽器の演奏追従は困難である．Mauch ら [6] は，ユーザ演奏への追従機能はないものの，任意の音楽音響信号から伴奏音（ドラムとベース）を生成する演奏支援システム SongPrompter を提案している．具体的には，原曲中のビート時刻とベースの音高の自動推定を行い，MIDI 音源を用いてドラムとベースのパートを合成している．また，与えらえたコード付き歌詞は自動で音響信号と同期し，音楽に合わせて歌詞とコードをスクロール表示することで，ユーザの歌唱やギター演奏を支援する．

提案するシステムは，原曲に含まれる高品質な伴奏音を再生できる点が特徴である．産業用途では，ユーザのギター演奏をリアルタイムに認識できる音楽ゲーム（例：ubisoft 社 Rocksmith）や，ユーザの歌声の音高を認識できるカラオケの採点機能は存在するが，あらかじめ伴奏音を準備しておく必要があった．提案するシステムでは，任意の楽曲に対する伴奏音の生成と，ユーザの歌唱・ギター演奏認識機能を統合した結果，臨場感のある音楽演奏を（場合によっては複数人で）楽しむことができるようになっている．伴奏音再生速度は原曲のままで変化させることができないが，MIDI データに基づく従来の自動伴奏システムのように，将来的には，ユーザの演奏に追従させるような拡張も可能であると考えられる．

5.2 音楽鑑賞・創作・演奏

近年，能動的音楽鑑賞インタフェース [15] が盛んに研究されている．「能動的」であるとは，音楽に合わせて演奏するだけではなく，音楽再生中に任意の編集を加える（一種の創作）ような，音楽を聞いて楽しむ上でのあらゆる能動的な体験も含む．例えば，後藤ら [3] は，音楽内容を自動推定して可視化することでユーザが楽曲の要素をより深く理解できるようになる Web サービス Songle を開発している（図 9）．吉井ら [16] は，音楽音響信号中に含まれるドラムパートのみを，あたかも MIDI データを編集するかのごとく直感的にカスタマイズできるシステム Drumix を提案している．糸山ら [17] は，楽譜情報を利用する音源分離手法を用いて，個別の楽器パートの音量をリアルタイムで調節できるシステムを提案している．安良岡ら [18] は，音響信号中の特定の楽器パートのフレーズを原曲の音色を保持しながら自由に編集できるシステムを提案している．深山ら [19] は，異なる音響信号中のコード進行の特徴を組み合わせて，新たなコード進行を生成することができるシステムを提案している．

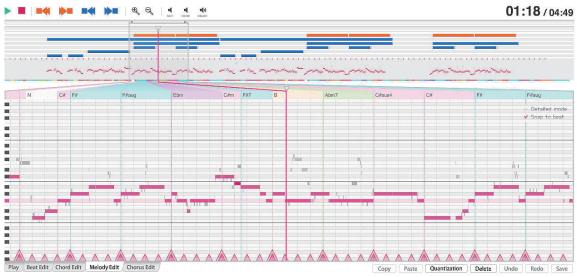

図 9 Songle の画面：楽曲の繰り返し構造・ビート時刻・コード進行・歌声の音高がブラウザ上で可視化されている．

Fig. 9 A screenshot of Songle: The repeated sections, beat times, chord progressions, and vocal pitches of music audio signals are visualized on the browser.

「音楽鑑賞・創作・演奏能力を拡張する」インタフェースも提案されている．例えば，Giraldo と Ramirez [20] は，ブレインコンピュータインターフェースにより検出した脳活動データを用いて，リアルタイムで演奏表現を制御するシステムを提案している．Mancini ら [21] は，モバイルデバイスとセンサーによりユーザの動きを解析し，リアルタイムでバーチャルオーケストラを操作するシステムを提案している．Chandra ら [22] は，モバイルデバイスで動作を検知し，ほとんど音楽経験のないユーザでもバンド演奏できるシステムを提案している．都築ら [23] は，複数のカバー曲をマッシュアップすることで，コーラス作成を補助するシステムを提案している．

提案するシステムは，歌唱や楽器演奏の練習を支援することが主目的であるが，2 章冒頭でも議論した通り，楽器パートの音量を制御するという観点では，能動的な音楽鑑賞であると解釈することもできる．一方，ユーザインタフェースについては，上記で紹介した従来研究と異なり，標準的なものを採用している．操作に伴う効果は過去の経験から十分に予測可能であるものの，従来不可能だと思われていた領域へ「操作対象を拡張する」ことで（MIDI 信号の操作から音楽音響信号の操作へ），魔法を使っているかのような不思議なインタラクティブ体験を提供する．

5.3 音源分離

近年，音楽音響信号の歌声・伴奏分離の進展は著しい．例えば，Rafii と Pardo [24] は，音楽スペクトログラム中の各小区間の繰り返し構造に着目することで，歌声を分離する手法 REPET を提案している．Liutkus ら [25] は，各時間・周波数ビンにおける音源成分が，その音源固有の近接カーネルに従って定義される周囲のビンから推定できると仮定することにより、REPET の考え方を一般化したカーネルベースモデリングを提案している．Huang ら [26, 27] は，教師ありまたは教師無しの歌声分離手法として，それぞれロバスト主成分分析 (RPCA) および Deep Neural Network (DNN) を利用する手法を提案している．歌声・伴奏分離の改良のため，Rafii [28] らは REPET に基づく歌声分離と歌声の音高推定を組み合わせる手法を提案している．同様に，RPCA と歌声の音高推定を組み合わせる手法が池宮ら [1] により提案されている．

調波楽器音・打楽器音分離にもいくつもの試みがある．吉井ら [16, 29] は，テンプレート適応・マッチングに基づき，ドラムのオンセット時刻を検出し，ドラム音を抑制する手法を提案している．Gillet と Richard [30] は，時間・周波数部分空間マスクを推定し，ウィーナーフィルタを用いる手法を提案している．Rigaud ら [31] は，短時間フーリエ変換における振幅変化に対するパラメトリックモデルを用いて，多声音楽からドラム音を抽出する手法を提案している．宮本ら [32] は，スペクトログラム上での調波楽器音と打楽器音の成分の異方性に着目し，コスト関数最小化としての定式化を試みている．Fitzgerald ら [2] も同様に，スペクトログラムの異方性に着目し，計算負荷の軽いメディアンフィルタを用いる手法を提案している．

提案システムでは，歌声分離のために，2014 年度の国際的な音楽解析コンテストである MIREX の歌声分離トラックで優勝した池宮らの手法 [1] を用いることにした．他の最新手法と比較して計算時間も十分に早いうえ，その大部分は特異値分解に占められており，GPGPU など特殊ルーチンを用いた高速化が容易である．研究に用いたタブレット端末である Nexus 9 では NVIDIA Tegra K1 を搭載しており，将来的には，実装の見直しにより，タブレット端末単体で解析から練習支援までスタンドアローンで動作できるようにすることを検討している．一方，打楽器音分離に関しては，Fitzgerald らの手法 [2] を用いることにした．スペクトルの異方性に基づく手法の分離精度は拮抗しており，採用手法は実装・計算時間・チューニングの容易さの点で優れている．

6. おわりに

本論文では歌声・調波楽器音・打楽器音分離とユーザ演奏のリアルタイム可視化に基づく音楽演奏練習システムについて述べた．本システムでは，歌声・伴奏音分離と調波楽器音・打楽器音分離の二手法を組み合わせた音源分離を行うことで，各楽器パートを任意の音量で再生することを可能にした．さらに，歌唱練習やギター演奏の効果的なサポートのために，Songle から取得したビート時刻・コード進行・歌声の音高を音楽再生に同期してスクロール表示するとともに，ユーザの歌声の音高やギターのコードをリアルタイムで推定することで，画面上に比較表示する．被験者実験により，各機能の必要性や有用性，個人の趣向や用途に合わせて任意に各音源の音量を設定できることの有効性を確認した．

今後は，DNN を用いてユーザ演奏の認識精度を改善するとともに，テンポの揺らぎなどユーザの演奏に自動追従する機能の開発を予定している．また，マイク 1 本でユーザが弾き語り（歌唱＋ギター演奏）を行う場合には，歌声とギター演奏をリアルタイムで分離しながら，歌声の音高推定とギターのコード認識を同時に行う機能も必要である．現在，開発したアプリケーションを一般公開することを目指して，音源分離アルゴリズムをタブレット上で動作するよう実装することに取り組んでいる．最近では，事前学習が必要であり，学習データへの過学習が懸念されるものの，運用時には従来手法の 100 倍以上高速に動作する DNN に基づく歌声分離も研究が進んでおり，音源分離のリアルタイム実行も視野に入ってきている．

謝辞 本研究の一部は，科研費 24220006，26700020，24700168 および OngaCREST プロジェクトの支援を受けた．

参考文献

[1] Ikemiya, Y., Yoshii, K. and Itoyama, K.: Singing Voice Analysis and Editing based on Mutually Dependent F0 Estimation and Source Separation, *Proc. ICASSP*, pp. 574–578 (2015).

[2] Fitzgerald, D.: Harmonic/percussive separation using median filtering, *Proc. DAFX*, pp. 246–253 (2010).

[3] 後藤真孝，吉井和佳，藤原弘将，Mauch, M.，中野倫靖：Songle: ユーザが誤り訂正により貢献可能な能動的音楽鑑賞サービス，*Proc. IPSJ Interaction*, pp. 1363–1372 (2012).

[4] Hermes, D. J.: Measurement of pitch by subharmonic summation, *JASA*, pp. 257–264 (1988).

[5] 中野倫靖，後藤真孝，　平賀譲：MiruSinger: 歌を「歌って/聴いて/描いて」見る歌唱力向上支援インタフェース，*IPSJ Interaction*, pp. 195–196 (2007).

[6] Mauch, M., Fujihara, H. and Goto, M.: SongPrompter: An accompaniment system based on the automatic alignment of lyrics and chords to audio, *Proc. ISMIR*, pp. 9–16 (2010).

[7] Dannenberg, R. B.: An On-line Algorithm for Real-time Accompaniment, *Proc. ICMC*, pp. 193–198 (1984).

[8] Vercoe, B.: The Synthetic Performer in the Context of Live Performance, *Proc. ICMC*, pp. 199–200 (1984).

[9] Tekin, M. E., Anagnostopoulou, C. and Tomita, Y.: Towards an intelligent score following system: Handling of mistakes and jumps encountered during piano practicing, *Proc. CMMR*, pp. 211–219 (2005).

[10] Pardo, B. and Birmingham, W.: Modeling form for on-

line following of musical performances, *Proc. Nat. Conf. Artif. Intell*, pp. 1018–1023 (2005).

[11] 中村栄太，武田晴登，山本龍一，齋藤康之，酒向慎司，嵯峨山茂樹：任意箇所への弾き直し・弾き飛ばしを含む演奏に追従可能な楽譜追跡と自動伴奏，情報処理学会論文誌，pp. 1338–1349 (2013).

[12] Nakamura, E., Cuvillier, P., Cont, A., Ono, N. and Sagayama, S.: Autoregressive hidden semi-Markov model of symbolic music performance for score following, *Proc. ISMIR*, pp. 392–398 (2015).

[13] Raphael, C.: Music Plus One: A system for flexible and expressive musical accompaniment, *Proc. ICMC*, pp. 159–162 (2001).

[14] Cont, A.: ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music, *Proc. ICMC*, pp. 33–40 (2008).

[15] Goto, M.: Active music listening interfaces based on signal processing, *Proc. ICASSP*, pp. 1441–1444 (2007).

[16] Yoshii, K., Goto, M., Komatani, K., Ogata, T. and Okuno, H. G.: Drumix: An audio player with real-time drum-part rearrangement functions for active music listening, *IPSJ Journal*, pp. 134–144 (2007).

[17] Itoyama, K., Goto, M., Komatani, K., Ogata, T. and Okuno, H. G.: Instrument Equalizer for Query-by-Example Retrieval: Improving Sound Source Separation Based on Integrated Harmonic and Inharmonic Models., *Proc. ISMIR*, pp. 133–138 (2008).

[18] Yasuraoka, N., Abe, T., Itoyama, K., Takahashi, T., Ogata, T. and Okuno, H. G.: Changing timbre and phrase in existing musical performances as you like: manipulations of single part using harmonic and inharmonic models, *Proc. ACM Multimedia*, pp. 203–212 (2009).

[19] Fukayama, S. and Goto, M.: HarmonyMixer: Mixing the Character of Chords among Polyphonic Audio, *Proc. ICMC-SMC*, pp. 1503–1510 (2014).

[20] Giraldo, S. and Ramirez, R.: Brain-activity-driven real-time music emotive control, *Proc. ICME* (2013).

[21] Mancini, M., Camurri, A. and Volpe, G.: A system for mobile music authoring and active listening, *Proc. Entertainment Computing*, pp. 205–212 (2013).

[22] Chandra, A., Nymoen, K., Voldsund, A., Jensenius, A. R., Glette, K. H. and Tørresen, J.: Enabling participants to play rhythmic solos within a group via auctions, *Proc. CMMR*, pp. 674–689 (2012).

[23] Tsuzuki, K., Nakano, T., Goto, M., Yamada, T. and Makino, S.: Unisoner: An Interactive Interface for Derivative Chorus Creation from Various Singing Voices on the Web, *Proc. SMC* (2014).

[24] Rafii, Z. and Pardo, B.: Music/Voice Separation Using the Similarity Matrix, *Proc. ISMIR*, pp. 583–588 (2012).

[25] Liutkus, A., Fitzgerald, D., Rafii, Z., Pardo, B. and Daudet, L.: Kernel Additive Models for source separation, *IEEE Transactions on Signal Processing*, Vol. 62, No. 16, pp. 4298–4301 (2014).

[26] Huang, P.-S., Kim, M., Hasegawa-Johnson, M. and Smaragdis, P.: Singing-voice separation from monaural recordings using deep recurrent neural networks, *Proc. ISMIR* (2014).

[27] Huang, P.-S., Chen, S. D., Smaragdis, P. and Hasegawa-Johnson, M.: Singing-voice separation from monaural recordings using robust principal component analysis, *Proc. ICASSP*, pp. 57–60 (2012).

[28] Rafii, Z., Duan, Z. and Pardo, B.: Combining rhythm-based and pitch-based methods for background and melody separation, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1884–1893 (2014).

[29] Yoshii, K., Goto, M. and Okuno, H. G.: Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression, *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 333–345 (2007).

[30] Gillet, O. and Richard, G.: Transcription and separation of drum signals from polyphonic music, *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 529–540 (2008).

[31] Rigaud, F., Lagrange, M., Robel, A. and Peeters, G.: Drum extraction from polyphonic music based on a spectro-temporal model of percussive sounds, *Proc. ICASSP*, pp. 381–384 (2011).

[32] 宮本賢一，亀岡弘和，小野順貴，嵯峨山茂樹：スペクトログラムの滑らかさの異方性に基づいた調波音・打楽器音の分離，日本音響学会秋季研究発表会講演集，pp. 903–904 (2008).