芸術科学会論文誌 Vol. 17, No. 4, pp. 83-93 (2018)

JackTopGuitar: オーディオビジュアルパフォーマンスのための ギターと音声入力を使用したライブインタフェース

大谷泰斗 [1)]（非会員） **越智景子** [2)]（非会員） **大淵康成** [2)]（正会員）

1)東京工科大学大学院バイオ情報メディア研究科 2)東京工科大学メディア学部

JackTopGuitar: Combined Live Interface Using Guitar and Microphone for Audio-Visual Performance

Taito Otani[1)] (Non - Member) **Keiko Ochi**[2)] (Non - Member) **Yasunari Obuchi**[2)] (Member)

1) Graduate School of Bionics, Computer and Media Science, Tokyo University of Technology
2) School of Media Science, Tokyo University of Technology

g311702769@edu.teu.ac.jp ochikk@stf.teu.ac.jp obuchiysnr@stf.teu.ac.jp

アブストラクト

JackTopGuitar は，ギター・マイク・フットペダルを組み合わせたオーディオビジュアルパフォーマンスのためのデジタルミュージカルインタフェース(DMI)である．ボイスコマンド・ボイスジェスチャ・ギタージェスチャ・ペダル操作を組み合わせることで，投影映像や音楽シーケンサ・エフェクタなどの多様なパラメータをリアルタイムに操作することができる．これにより，即興性の高いオーディオビジュアルパフォーマンスが可能になる．演奏者は，伝統的なエレキギター演奏を踏襲した JackTopGuitar を使うことで，音声入力などを組み合わせたパフォーマンスシステムの制御を可能とし，演奏表現を拡張することができる．映像表現は，ギターや音声入力の演奏結果の単なる可視化ではなく，演奏と一体化したパフォーマンスの一部となる．

本稿では，JackTopGuitar の設計と実装について述べ，ライブパフォーマンスという過酷な環境下における音声認識率についての評価を行う．最後に，この DMI を利用したオーディオビジュアルライブパフォーマンスを行った後に，いくつかの問題についての議論と展望を述べる．

Abstract

JackTopGuitar is a digital musical interface (DMI) that uses an electric guitar, microphone and foot pedal for a computer-based audio-visual live performance. It consists of voice commands, voice gestures, guitar gestures, pedal controls, and their combined actions. The performer can project videos and control various parameters of sequencers and effectors. The interface realizes real-time audio-visual musical performances. JackTopGuitar is based on the traditional electric guitar playing style, and its performing expression is augmented by the extra functions such as speech recognition. Visual expressions are not just visualization of vocal and guitar plays but a part of the performance created by the music and performer's control.

In this paper, we describe the design and implementation of JackTopGuitar. We also conduct evaluation experiments of speech recognition under an adverse condition of live performance. Finally, we make an audio-visual performance using JackTopGuitar and discuss various issues found in the performance.

1. はじめに

近年，デジタル技術の発展により，電子音楽や実験音楽に限らずシーケンサやシンセサイザ，サンプラーといった音楽アプリケーションが広く使われるようになった．また，信号処理技術の進歩と品質の向上により，リアルタイムに複雑な計算と高次元パラメータの制御が可能となった．

これらの技術は，レコーディング現場のみならず，リアルタイム性が求められるライブパフォーマンスでも使用されるようになってきている．代表的な音楽アプリケーションには，Ableton Live，Cubase，Max，SuperColliderなどがある．これは，音楽パフォーマンスをデザインするためのツールとしてミュージシャン，アーティスト，作曲家の間で使われている．これらの音楽アプリケーションは，Musical Instrument Digital Interface (MIDI)やOpen Sound Control (OSC)といった規格に対応したノブやスライダ・パッドといった機器を自由にカスタマイズして，様々なパラメータを制御することができる．

例えば，Novation Launch Controlは，16個のノブと12個のパッドからなるインタフェースで，音楽アプリケーション上の様々なパラメータにノブやパッドを対応させることができる．ノブを回したり，パッドを叩いたりすることで，多くのパラメータを操作することができる．また，ラップトップに付属されたマウスパッドやキーボードと組み合わせパラメータとノブ・パッドの対応関係を切り替えることで，さらに多くのパラメータの操作が可能になる．これらの操作により，映像パラメータ操作，エフェクトパラメータの操作，サンプラー・シーケンスの再生停止などを駆使したライブパフォーマンスが実現される．

しかしながら，ノブやスライダは，手を使って操作を行うため，他の楽器を演奏しながらの制御はとても難しい．これが，ラップトップパフォーマンスと伝統的な楽器演奏の共存ができない決定的な理由である．このような音楽アプリケーションを駆使したラップトップパフォーマンスは，自由な表現力と手軽さを持つ一方で，伝統的な楽器演奏の持つ演奏のしやすさや，演奏表現を生かすことは難しい． また，典型的なラップトップパフォーマンスでは，シーケンサの再生停止，シーケンスパターンの切り替え，ディレイやフィルタといったエフェクトのオンオフやパラメータの操作が必要になるため，即興的な操作のためには，複数のパラメータの操作のためのインタフェースが必要になる．

伝統的なギター演奏では，片方の手で押弦，もう一方の手でピッキングし演奏するため，演奏の最中に手を使って，ノブやスライダを操作することは難しい． そこで，従来のギター演奏では，フットペダルを用いて，足を使ったパラメータの制御が行われてきた．この手法では，足を使って物理的に操作するため，確実な制御を可能にする．一方で，物理的な制限により，オン・オフ操作等の単純なパラメータ操作以外を行うためには，工夫が必要になる．

例えば，BOSS GT-1000は，10個のペダルスイッチと7個のノブと1個のExpressionペダル（フェーダのようなパラメータの操作ができる）を持つ．しかしながら，一般的なギター演奏を伴うパフォーマンスでは，演奏者は立ち上がった状態で演奏することが多く，片足のみで操作が要求されるため，一度に1つ以上のパラメータを操作することは難しい．そのため，一度に制御できるノブやスライダの数には物理的な制約が存在する．これが伝統的な楽器演奏においてラップトップパフォーマンスの自由度が発揮できない決定的な理由である．

別の手法として，楽器演奏のための動作自体をインタフェースとして用いる手法もある． これは，演奏ジェスチャの特徴を最大限に生かすことができるが，これだけでは限定的な範囲内でしか，パフォーマンスを最適化することができないなどの問題が挙げられる．

我々は，以前の研究で伝統的なエレキギター演奏を基にした音声認識を使ったDMIを開発した[1]．これは，音声認識を使うことでギターエフェクタを制御し，ギター音色を制御できるというものである．本稿では，このアイディアを拡張し，ボイスコマンドによる物理的な制約のないパラメータ選択手法に着目し，音楽・映像アプリケーションの持つ複数のパラメータの制御を可能とする伝統的なギター演奏の表現力を持つオーディオビジュアルパフォーマンスのためのインタフェースを提案する．これにより，伝統的なエレキギター演奏のスタイルを踏襲しながら，ラップトップパフォーマンスに見られる高次元のパラメータの操作によるサウンド表現を実現する．マイク入力やギター入力，フットペダル操作といった複数のインタフェースをモジュールとして組み合わせることで高次元のパラメータ操作を可能とする．

技術的にボイスコマンドは，発話から認識までに遅延が存在する．演奏にとって実行タイミングの正確な制御は極めて重要であるため，ボイスコマンドの実行タイミングは，伝統的なギター演奏に用いられてきたフットペダルを踏むことで決定する．ギター演奏者によってラップトップの持つパラメータを操作可能としたことで，伝統的なギター演奏の拡張を実現する．

これらのコンセプトに基づき，ライブパフォーマンスシステムのプロトタイプである“JackTopGuitar”を開発した． このJackTopGuitarでは，伝統的なギター演奏を生かしながら，ラップトップパフォーマンスの表現力に迫ることができる．

例えば，サンプラーやシーケンサを動的に展開させながら，それを伴奏にしたギター演奏や，ギター演奏に合わせた映像オブジェクトの操作，映像オブジェクトの動作に合わせてサンプラーを再生したり停止したり等ができる．これにより，従来のギター演奏では難しかった演奏者の音楽的な意思による映像表現と伝統的なギター演奏の音楽表現の拡張を目指す．

本インタフェースは音声認識機能を基盤とし，高次元のパラメータの制御を実現する．入力された音声からは，音声認識で用いる言語情報だけでなく，声の抑揚のような非言語情報も抽出し，これらをボイスジェスチャとして活用する．さらに，エレキギターの入力からは，音響解析を行いアタックやストロークなどを検出しパラメータ操作に活用する．また，パフォーマンスの状態をリアルタイムに反映した映像提示を行う．

今回のプロトタイプでは，ボイスコマンドにより映像・音楽・ギターエフェクタの持つパラメータの指定や，ボイスジェスチ

ャによるパラメータ操作を可能とした．また，ギタージェスチャによる映像オブジェクト制御によるオーディオビジュアルパフォーマンスを可能とする．

音声認識は，幅広い場面で実用化されるようになった技術であるが，騒音環境下での性能は必ずしも十分とは言えない．また，本来演奏者の声は，パフォーマンスの一部として演じられるものであり，それと並行する形で音声認識用のコマンドが発声されるという特殊条件下では，通常と異なる音声認識性能が得られる可能性もある．そこで本システムのユーザビリティを決める最重要要素として，音声認識機能の性能評価を行った．特に使用するマイクは，ステージ上に置かれたスタンドマイクを用いる方法とヘッドセットマイクを用いる方法が考えられるため，両者の比較検討を行った．これにより，ライブステージにおける音声認識には，ヘッドセットマイクよりスタンドマイクの方が優れていることが分かった．加えて，音響モデルの違いによる認識率の検証を行った．結果，日本語の音響モデルの方が安定した認識が可能であることがわかった．また，ギター経験者によるギタージェスチャ評価を行なった．

最後に，4名のギタリストによる本インタフェースを使ったオーディオビジュアルパフォーマンス(図1)を行い，インタビュー調査行い，得られた知見や問題点についての議論を行う．

図1. 演奏の様子 シーン「バブル」

2. 関連研究

2.1 ギター演奏の拡張

これまでギター演奏は演奏技術だけでなく音色エフェクト制御により，表現の拡張がされてきた．Robotically Augmented Electric Guitar[3] は，ロボットと人間が演奏を共有できる．アクチュエータにより動作するハンマーが弦を弾き，リズムパターンを生成する．演奏者はギター指板をおさえ演奏する．これは，ロボティクスによって演奏表現を拡張しており，機材を改良することによりギター演奏を拡張している例である．

BioStomp[4] は，ギター演奏者の腕の筋肉の動きを取得し，ギターエフェクタの物理的なノブを直接制御するシステムである．これは，ギター演奏における身体の動きをギターの音色に反映させるものである．このように演奏技術や音色表現の拡張だけでなく，新たにインタフェースを取り入れることで表現の拡張をしている．

2.2 オーディオビジュアルパフォーマンス

音楽と密な関係性を持った芸術的な映像提示のことをオーディオビジュアルパフォーマンスと呼ぶ[5]．Live Writing[6] は詩をリアルタイムで観客へ提示するオーディオビジュアルパフォーマンスである．イギリス出身のアーティストであるMax CooperはEMERGENCE[7] など，多くのオーディオビジュアル作品を公開している．また，「UTP_」はピアノ，室内楽団，エレクトロニクスに抽象的な映像表現を加えた作品である[8].

2.3 音声インタフェース

演奏者の音声情報を使ったインタフェースを使ったパフォーマンスについては，いくつかの議論が存在している．Fascianiら[9] は，物理的に制限のあるサウンドシンセサイザの音色をボイスジェスチャにより，制御するためのボイスインタフェースを提案している．Stowell[10] は伝統的なボイスパーカッションのスキルを使ってビートミックスができるインタフェースを提案している．これは高レベルの音声信号処理のおかげで高い表現力を実現している．Igarashiら[11] は，声の高さや大きさといった簡単な特徴量を使った非言語インタラクションを提唱している．また，ワウワウエフェクトをボイスジェスチャにより操作可能にしたものが存在する[12].

音声入力は，自然なインタフェースとして，様々な環境で利用されているが，デジタルミュージカルインタフェースの分野では，遅延の問題のため，ほとんど利用されてこなかった．しかしながら，Deacon[13] は音声認識とボイスジェスチャを組み合わせたシステムを提案している．これは，音声入力の長さにより，制御する対象が変化するインタフェースで，短い発話はボイスコマンドとして利用される．中程度の長さの発話は，シンセサイザの音色の選択に使われ，長い発話は音高やノート（音の長さ）の制御に使われる．

3. 設計と実装

この章では JackTopGuitar の設計と開発について述べる．図 2 は，システム全体の概要図である．本システムは，メイン(図 2 上)，サウンド(図 2 右下)，ビジュアル(図 2 左下)の 3 つのパートに分けられる．これらはそれぞれ Max・Ableton Live・openFrameworks で開発した．メインアプリケーションは，ギター，マイク，フットペダルからの入力をギター信号解析モジュール，音声解析モジュール，音声認識モジュールで解析する．そして，それぞれの出力を統合モジュールに集め，音楽アプリケーションや映像アプリケーションと連携させる．この統合モジュールをカスタマイズすることで，特定のパフォーマンスを想定した機能の組み合わせを実現することができる．メインアプリケーションと映像アプリケーションは OSC で通信している．メインアプリケーションと音楽アプリケーションは MIDI 規格で通信を行なっている．また，フットペダルとの通信にも MIDI を用いた．

3.1 インタフェースデザイン

JackTopGuitarのインタフェースは，エレキギター，マイク，フットペダルからなる伝統的なエレキギター演奏を基に設計した．

これらのインタフェースに協調性を持たせることで，ラップトップの持つ演奏能力をエレキギター演奏へ持ち込む．マイク入力は音声認識と音響解析により，言語情報と非言語情報を取り出す．演奏者はボイスコマンド（言語情報）とボイスジェスチャ（非言語情報）を使い分けることでパラメータの制御が可能となる．さらに，ギター演奏からの入力をギタージェスチャとして，パラメータ制御に取り入れる．これにより両手が塞がってしまうギター演奏中においても，高次元のパラメータの制御が可能となる．

しかしながら，音声認識は，技術的な問題から遅延を避けることができない．また，音楽演奏においてタイミングは極めて重要である．これらの問題を解決するため，ギター演奏において慣れ親しまれたフットペダルに着目し，このペダルを踏むことでボイスコマンドを実行できるようにした．また，ボイスジェスチャだけでなく，ギターピックアップからの音響信号を解析しギタージェスチャとして，パラメータ制御ができる．

ボイスジェスチャとギタージェスチャは，それぞれ出力に特徴が存在する．例えば，ボイスジェスチャでは安定したピッチや持続的な音量キープは難しいが，音量ダイナミクスのコントロールに向いており頻繁なパラメータ操作や増減を大きいパラメータができる．一方でギタージェスチャでは，安定したピッチの出力や持続的な音量キープに向いているといった特徴が挙げられる．ギタージェスチャを使うことで，特定の値を出力したい場合や，一定の時間，同じ値を出力し続けたいといった場合のパラメータの操作を可能にする．

さらに，ビジュアライズされたオブジェクトの持つ動きや位置，個数などをパラメータとしてエフェクトやシーケンスなどに割り当てることができる．

例えば，音楽アプリケーションにはシーケンサ，サンプラーエフェクト等を構築し，それぞれのパラメータをボイスコマンド，ボイスジェスチャ，ギタージェスチャによって操作することができる．

今回のプロトタイプでは，6つのトラックにシーケンスパターンを読み込み，それぞれのボイスコマンドにより最大31パターンのシーケンストラックを呼び出すことができる．また，ギターエフェクトでは，ボイスコマンドにより読み込まれた個別のエフェクトのオン・オフや，ボイスジェスチャにより同時に最大２つのパラメータの操作ができる．エフェクトパラメータは最小値0から最大値127の範囲で操作することができる．加えて6つの映像シーンを切り替えやビジュアルオブジェクトの操作ができる．

3.2 ギタージャスチャ

ギター演奏を通じてインタフェースを制御することをギタージェスチャと呼ぶ．ギター演奏にラップトップの操作を加えるためには，ノブやスライダといったパラメータ操作の代わりとなるジェスチャを導入する必要がある．ギター演奏の特性上，両手が塞がってしまう．さらに，本来のギター演奏以外のジェスチャを演奏者に要求すると，演奏品質の低下につながるという観点から，ギター演奏自体をジェスチャとして利用することにした．これをギター信号解析モジュールとしてシステムに実装した．このモジュールはピッチ，アタック，長さ，弦を弾く強さ，持続的な強いストロークを検出することができる．

ギタージェスチャは，ギター入力の信号解析を行うことで取得する．ピッチは，フレームサイズ内の基本周波数によって定義される．今回のプロトタイプにおけるフレームサイズは約43ms(理論値）とした．アタックは，ギター入力が閾値以下の状態から閾値以上になった場合に検出される．長さはアタックを検出してから，信号レベルが閾値以下になるまで区間により定義される．弦を弾く強さは，ギター入力のフレームサイズ内のピークの値から定義される．持続的な強いストロークは，ギター信号解析モジュールだけで定義せず，0.8以上の強度になるとオブジェクトが生成される．一定以上の個数になった場合にトリガーが送信され，持続的な強いストロークとなる．オブジェクトは，約5秒経過すると消滅する．今回のプロトタイプでは，50個以上のオブジェクトが生成された場合にトリガーが送信されるように設定した．

また，エレキギターからの信号レベルはピックアップの種類の影響を大きく受けるため，GUIによりそれぞれの検出器の閾値を調整できる．それぞれの検出されたパラメータは，ギタージェスチャとして統合モジュールへ送られる．

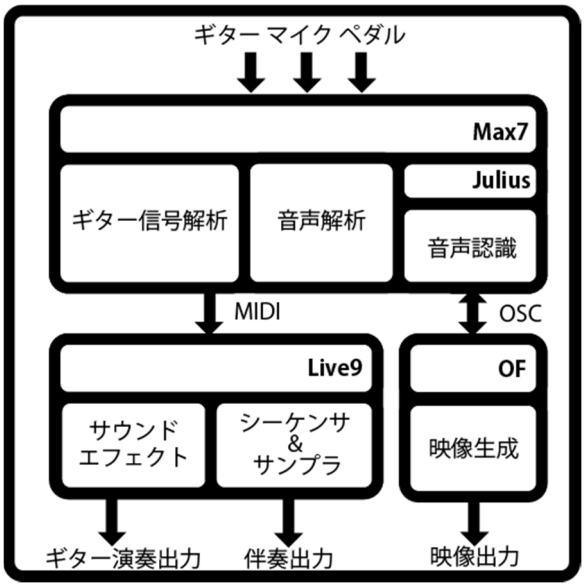

図 2. システム概念図

3.3 ボイスジェスチャ

演奏者の発話の音響的情報を使って，インタフェースを制御することをボイスジェスチャと呼ぶ．マイク信号はマイクの種類や演奏者の声質，パフォーマンス環境の影響を大きく受けるため，音量とピッチ検出の閾値を調整する必要がある．これらの値は，GUI により調整できるようにした．また，ハイパスフィルターを通すことで低域のノイズ成分を調整できるようにした．

ボイスジェスチャは，マイクからの入力信号の音量と音高から定義される．音量はフレーム毎のピーク値によって定義される．音高はフレーム毎の基本周波数によって定義される．

このモジュールの出力は，ボイスジェスチャのパラメータとして統合モジュールへ送られる．音量は入力信号から最低値 0，最高値 1 として扱う．量子化ビット数は，マイクの接続されたオ

ーディオインタフェースに依存する. 今回のプロトタイプでは, 16 ビットで処理をしている. 検出されたピーク値は, 最低値 0, 最大値 127 の整数値として変換し出力する.

3.4 音声認識

音声認識エンジンにはオープンソースの音声認識エンジンであるJulius[14] を利用し，音声からボイスコマンドを抽出した. Japanese Newspaper Article Sentences(JNAS) と Corpus of Spontaneous Japanese(CSJ)を基にしたGMM-HMM音響モデルを利用した. これらはJulius Dictation-kitとして公開されている. 音声認識エンジンは，Maxにより書かれたメインアプリケーションとは別スレッドで実行した. マイクからの音声は，ソケットを通じてMaxから送られ，Juliusは認識結果をMax内のshellインタフェースを通して送信されるように実装した. ボイスコマンドは単体で発音されることを想定し，孤立単語認識である.

JackTopGuitarでは, 認識結果をボイスコマンドとして取り扱う. 音声区間検出はJulius側で実行する. 音声区間が検出されると，既に登録されたフレーズのうち最大尤度となったものをメインアプリケーションへ返す. 音声区間を検出し認識結果を返すという一連の処理には，音楽的に許容できない遅延が存在する. ボイスコマンドの実行するタイミングは，フットペダルから行えるようにした.

ボイスコマンドは, 単語辞書に事前にフレーズを登録することで，多くのパラメータの柔軟な制御できる. 単語辞書には，ボイスコマンドと発音を登録する必要があり，ボイスコマンドと操作したいパラメータをマッピングできる. これはメインアプリケーションのGUIから登録することができる. ボイスコマンドは，サンプラー，シーケンサー，シンセサイザ，エフェクタ，シーンセレクト，映像オブジェクトの操作など，ビジュアルとサウンド両方の対応関係を登録できる.

3.5 統合モジュール

統合モジュールは，ボイスジェスチャ・ボイスコマンド・ギタージェスチャ・フットペダルを，楽器やエフェクタ，映像表現といったアプリケーションのパラメータとの関係性を定義するモジュールである. アプリケーションパラメータは，ギターエフェクト，シーケンサコントロール，サンプラー，ビジュアルフィードバックがある. ギターエフェクトは，音楽アプリケーション上のエフェクトパラメータの制御ができる. サンプルコントロールには，サンプルの再生，停止，および長さの設定ができる. ビジュアルコントロールには，オブジェクトの形状と数，スケール，色，モーション規則，生成タイミング，シーン制御のパラメータがある. このモジュールの設定は，パフォーマンスの内容に従ってカスタマイズすることができる.

3.6 ビジュアルコントロール

ビジュアルコントロールは, 主に演奏者の背後に投影された映像の生成と制御を行う.

従来のオーディオビジュアルパフォーマンスシステムでは, 音楽を解析し，生成規則によって自動的に映像が生成されるものや，ライブビデオをキャプチャしプレイバックや画面分割などの効果を与えるもの，ライブビデオ信号から異なる映像を生成するもの，用意した映像素材を読み込みミックスやエフェクトをかけるもの，リアルタイムでプログラミングしながら，映像を生成したり，合成したりするものがある[15]. 映像素材を使う場合には素材の切り替え，生成規則によって生成される映像の場合は，関数に与えるパラメータを操作する必要がある. 一般的に曲の展開やビートに合わせてシーンを変更やエフェクトのパラメータの操作が要求される.

本論文で提案する手法では, 生成規則によって映像生成されるタイプと映像素材を読み込みエフェクトかけるタイプを組み合わせたオーディオビジュアルパフォーマンスシステムで，動的な映像生成画面と映像素材による画面の切り替えに加え，映像エフェクトや生成規則へ与えるパラメータを操作することができる. 映像は，生成規則のパラメータによりオブジェクトの生成やサイズ, 形, 数, 動きなどを制御することができる. また，演奏者が視覚効果を積極的に制御しているとみなし，視覚制御の結果を JackTopGuitar に送信することもできる. 例えば，パーティクルオブジェクトを，ギターのストロークに従って生成させ，これらの数値が上限に達すると，音楽アプリケーション上のサンプルが再生されるなどのカスタマイズが可能となる. また，OSC によりパラメータ通信することで，描画用 PC を別に用意して連携させることができ，処理負荷を分散させることができる. これにより高負荷な映像表現が可能となる.

表 1. ボイスコマンド一覧

ボイスコマンド	対象パラメータ
TEST	ボイスコマンドのテスト用
CHECK	テストコマンドを終わる
GUITAR	ギタートラックをオンオフ
NEXT	映像シーンを次へ送る
GO	音楽シーケンスを次へ送る
BACK	音楽シーケンスを前へ戻す
COLOR	映像シーンの色を変更
START	音楽シーケンスを再生
STOP	音楽シーケンスを停止
EFFECT ON	ギターエフェクトをオン
EFFECT OFF	ギターエフェクトをオフ
BEAT	テンポ入力をオンオフ
BUBBLE	映像シーンをバブルにする
SPARK	映像シーンを spark にする
PEDAL CONTROL	選択パラメータの変更をオン
DISTORTION	歪みギターエフェクト
DELAY	ディレイエフェクト
CHORUS	コーラスエフェクト
BUBBLE DOWN	バブルシーンにてオブジェクトを下方向へ動かす
RAIN SOUND	雨のシーンに切り替え

4. 評価

ボイスコマンド認識に使われる音声認識部分およびギタージェスチャによる評価を行った．音声認識は，一般的な環境下における音声認識の認識率に関する報告は，多くされてきたが，ライブコンサートを想定した音声認識率の定量的な評価はほとんど行われていないという理由から，ライブコンサートでの利用を目的とした音声認識インタフェースの認識精度に関する評価を行う．まず，マイクの違いによる認識テストを行い，次に音響モデルの違いによる認識テストを行う．さらに，演奏者の主観評価によるギタージェスチャによる入力パラメータ変化のテストを行う．

4.1 マイクの種類による認識率の評価

実際のライブパフォーマンスでは，ギター演奏以外にドラムやボーカル，ベース，モニタスピーカなどが鳴り響いている環境が予想される．このような音は，背景演奏として音声認識精度に影響することが考えられる．そこで，マイクの種類の違いによる認識精度を調査する．

本インタフェースは，ライブステージでの利用を想定しているため，ライブステージで一般的に使われているスタンドマイクとヘッドセットマイクの認識率についてそれぞれ評価した．

また，このような環境下において，認識が難しいボイスコマンドを評価した．

4.1.1 実験手順

この実験では，使用するマイクによる認識率の差，および，背景演奏音の有無による認識率の差を評価した．ボイスコマンドは，パフォーマンスで使う20単語を選び，評価実験用の単語辞書を作成した．選択されたボイスコマンドは表1の通りである．大学内の防音室において，12名被験者（うち被験者5と11は女性平均年齢22歳）に，1つずつボイスコマンドを読み上げてもらい，その音声を2種類のマイクで収録した．被験者には，収録を始める前に実験の手順と20個それぞれのボイスコマンドの読み方を確認してもらった．その後，PC画面に表示されるボイスコマンドを順番に読み上げてもらった．収録はスタンドマイク，ヘッドセットマイクの順に行った．収録の時の注意点として，図3のように口をマイクに近づけるように指示し，発話してもらった．なお，言い間違いなどがあった場合には，間違えたボイスコマンドを一呼吸おいてから発話し直すように指示を与えた．

マイクは，多くのライブ現場や放送等で利用されているダイナミックタイプを選択し，スタンドマイクのSure SM58とヘッドセットマイクのSure WH20XLRを使った．オーディオインタフェースは，Focusrite Scarlet2i4をPCに接続し録音した．12名の被験者の中には，英語のネイティブスピーカは含まれず，発音に関して特に指示は行わなかった．音声は48kHz/16bitで収録し，ボイスコマンドごとに前後0.5秒の間隔で切り出した．

次に，ライブステージでの利用を想定した評価を行うため，ボイスコマンドが収録された音声データに，別途録音した背景演奏を加算した．背景演奏の収録は，観客のいないリハーサル中のライブハウスのステージにて行った．実際の使用状態を想定し，ボイスコマンドを収録した際と同じスタンドマイクとヘッドセットマイクを設置し，同じ楽曲を順番に計2回演奏した．背景演奏の収録では，ドラム・ベース・シンセサイザにより構成されたボーカルの含まれない楽曲[16]を演奏した．ノイズを加えるため，収録した背景演奏から振幅のはっきりした区間を選び，20個に分割した．背景演奏を加算する際に，ボイスコマンドの発話の途中で途切れることがないようにするため，背景演奏が収録音声よりも長くなるように分割した．背景演奏は，発話者ごとの各ボイスコマンドにランダムに加算した．スタンドマイクで収録した背景演奏は，スタンドマイクで収録したボイスコマンドの発話に，ヘッドセットマイクで収録したボイスコマンドは，ヘッドセットで収録した背景演奏に加算した．全ての音声ファイルのSN比の平均は，1.17dBであった．

図3. 収録の時のマイクセッティングの様子

4.1.2 マイク種類別の認識の結果

ボイスコマンドのみ条件(Clean)と背景演奏を加えた条件(Noisy)におけるマイクによる認識率の違いの結果を述べる．図4は，スタンドマイクにおけるCleanと背景演奏を加えたNoisyの被験者ごとの誤認識率を表している．図5は，ヘッドセットマイクにおけるCleanとNoisyの被験者ごとの誤認識率を表している．図4と図5の横軸は被験者番号である．図4のスタンドマイクを利用したCleanでの平均誤認識率は，5.8%となった．スタンドマイクのNoisyでは平均誤認識率は，30.8%となった．同様にヘッドセットマイクにおけるCleanの誤認識率の平均は，12.5%となり，Noisyでは，51.2%となった．スタンドマイクの方が，ヘッドセットマイクより高い認識率であった．次に図6と図7に単語別の誤認識率を表す．ボイスコマンドの「pedal Control」と「Rain Sound」は，誤認識率が低い．反対に「Back」「Start」「Stop」の誤認識率はスタンドマイクでも，ヘッドセットでも誤認識率は60%を超える結果となった．「Back」では「Bubble」や「Bubble down」「Spark」，「Start」では「Stop」「Spark」「Next」の誤認識，「Stop」では「Next」「Effect Off」などの誤認識が見られた．

4.2 音響モデルによる認識率の評価

デジタル音楽用語のほとんどは，英語由来のものが多いため，パラメータをボイスコマンドとして採用すると英単語が多く

なる．しかしながら，発話者が日本語発話者であるため，日本語音響モデルと英語音響モデルの音声認識率の差異について検証を行う必要がある．そこで，音響モデルの違いによる音声認識率の差異について調査し，その結果について述べる．

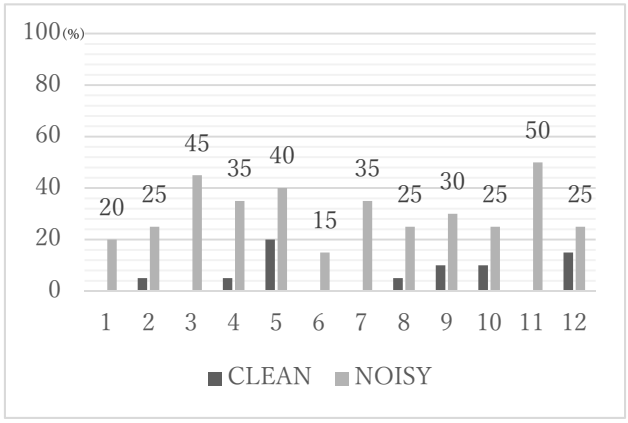

図4. 被験者別スタンドマイク音声認識の誤認識率

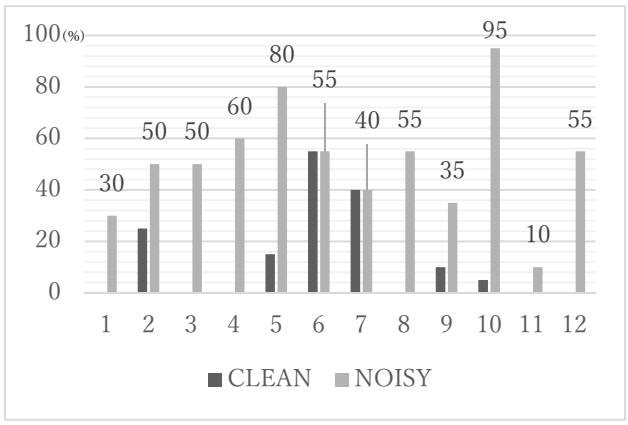

図5. 被験者別ヘッドセットの音声認識の誤認識率

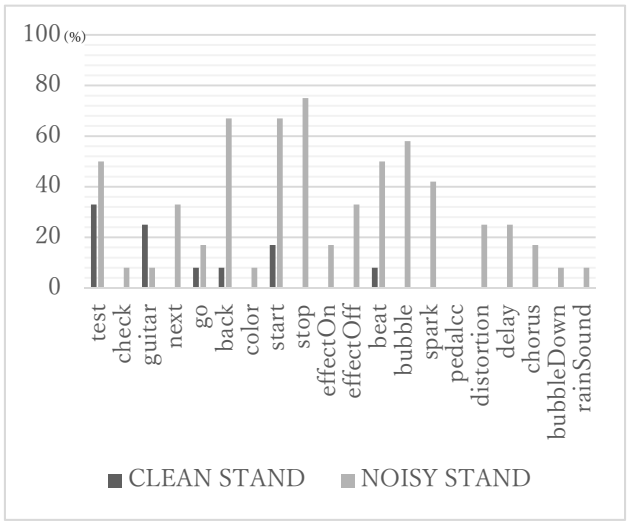

図6. コマンド別スタンドマイク音声認識の誤認識率

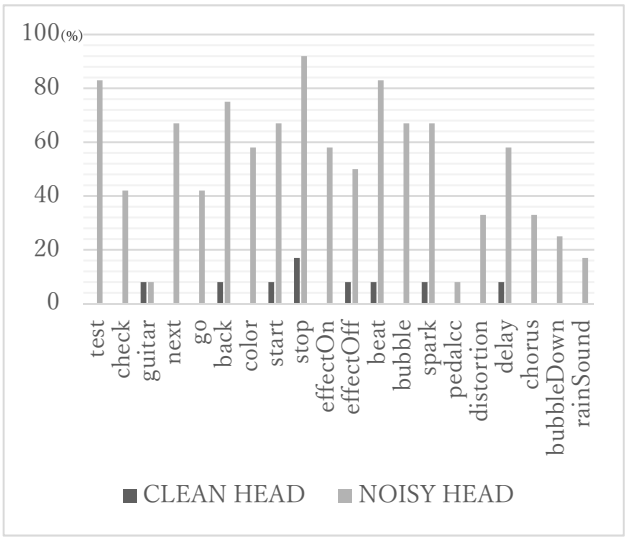

図7. 単語別ヘッドセットの音声認識の誤認識率

4.2.1 実験手順

認識させる音声データセットは，実験1と同様のものを使った(12名の発話者ごとに20個のボイスコマンドとそれぞれに背景演奏を加算した計480個の音声データセット)．英語音響モデルは，VoxForge Acoustic model[17] を利用した．また，表1にあげた20個のボイスコマンドの英語発音を記述した単語辞書を作成した．

4.2.2 英語音響モデルの認識率

実験1では，日本語音響モデルを使った際の全体の認識の平均誤認識率は22%であった．英語音響モデルを使った音声認識エンジンにおける480個全てのボイスコマンドの認識率の平均誤認識率は52%となった．ボイスコマンドのみのスタンドマイクの平均誤認識率は44%，ヘッドセットマイクは43%となった．背景演奏を加えたスタンドマイクの平均誤認識率は63%，ヘッドセットマイクは59%となった．単語別に見ると「Guitar」「Effect On」「Rain Sound」といったボイスコマンドが80%を超える平均誤認識率となった．ライブで使用するコマンドを想定した日本語話者による英単語の発話の認識には，日本語音響モデルのほうが適していることがわかった．

4.3 ギタージェスチャの評価

ギタージェスチャによるパラメータ入力を評価を行うためテストを行った．ギター経験者に実験を依頼し，入力した感覚と実際の結果との違和感について調査を行った．

4.3.1 実験手順

ギター経験者である男性4名(21から23歳)にギタージェスチャによるパラメータ入力テストを行った． 6弦のみを開放で弾いた場合，1弦の12フレット目を弾いた場合，Eメジャーコードで強く弾いた場合・弱く弾いた場合，Eメジャーコードで強く持続的にストロークした場合，Eメジャーコードで強弱をつけてストロークした場合を試してもらい，パラメータの動きの違和感を調べた．今回は，ギタージェスチャによって気泡に似たオブジェクトを生成する「バブル」のシーンで実験を行った．

4.3.2 結果

表 2 に結果を表す. (1) 6 弦のみを開放で弾いた場合は，4 名中 2 名が予想通りの結果だったと示した. (2) 1 弦の 12 フレット目を弾いた場合は，3 名が予想通りの結果だったと示した. (3) E メジャーコードを持続的に強く弾いた場合は，4 名が予想通りの結果だったと示した. (4) E メジャーコードを強弱をつけて引いた場合は，4 名が予想とは違う結果となったと示した. (5)E メジャーコードで弱く弾いた場合，4 名が予想とは違う結果となったと示した．弱く弾いた際に想像よりもバブルが生成されずないという意見を得た. (6)E メジャーコードで強く弾いた場合，4 名が予想通りの結果となったと示した.

持続的な弾き方や，強く弾いた場合は，予想通りの挙動を確認できたが，強弱のついた弾き方と実際のパラメータ変化による表現には改善の必要があることがわかった．より繊細なギタージェスチャ入力のため，演奏者適応や正規化の処理などを検討する必要があることがわかった.

表 2. ギタージェスチャ実験結果

	被験者 A	被験者 B	被験者 C	被験者 D
(1)	○	×	×	○
(2)	○	×	○	○
(3)	○	○	○	○
(4)	×	×	×	×
(5)	×	×	×	×
(6)	○	○	○	○

○予想通り　×違和感あり

5. パフォーマンスによるユーザ評価

我々は自然現象をテーマにコンテンツの制作, 楽曲制作を行い, JackTopGuitar を使ったエレキギター演奏経験者 4 名にオーディオビジュアルライブパフォーマンスを依頼し，ライブパフォーマンスを行ってもらった.

評価実験の結果を受け，マイクにはスタンドマイクを使用した．また音声認識には，日本語音響モデルを使用した．スタンドマイクは 4 章の実験と同じ，ダイナミックマイクの SURE SM58 を使用した. オーディオインタフェースとして Focusrite Scarlet2i4 を使った．エレキギターは，ハムバッカータイプの Yamaha RS-502 を使用した．また Roland SY-300 をフットペダルとして使用した．サウンド生成には，Ableton Live を使用し，シーケンサ，サンプラー，エフェクトを構築した．Live プロジェクトには展開とループパターンを作成した．それぞれのシーンはドラム，ベース，電子ピアノ，シンセサイザ(コード，パッド)で構成された 9 つのシーケンスパターンを作成した. これらのパターンの再生・停止・再構成は，ボイスコマンドから行えるようマッピングした．映像アプリケーションでは，ジェネラティブなアニメーションとプレレンダリングされたビデオを用意した. これらは, 自然現象をモチーフとしたもので,「雨」「バブル」「風」「石」といったシーンを用意した．それぞれの外観は制御パラメータによって制御できるように映像アプリケーションに読み込んだ.

図 9 は「石」のシーンである．矩形がグリッドパターンで投影され，ギタージェスチャに対応してパターンが点滅・移動する．また，演奏と結びついた視覚的なパフォーマンスの例として, 図 10 最上部は「バブル」のシーンである．ギターの持続的なストロークによりバブルが生成され，演奏者の背景に蓄積されていく．蓄積されたバブルは，演奏者が「Bubble Down」のボイスコマンドを与えると，落下し，サンプラーを鳴らしながら破裂し，音楽表現に影響を与える．さらに，ボイスコマンド「Color」を使うと気泡の色を変えたりすることができる. 図10 に映像シーンごとのビジュアルオブジェクトの変化を表す．上から「バブル」「雨」「風」「石」を表している.

ギターエフェクトは，Ableton Live 上に「低歪み」「高歪み」「ディレイ」「フランジャー」「コーラス」を用意した．これらはボイスコマンドにより呼び出しができる．例えば，演奏者が「ディレイ」とボイスコマンドを発話するとディレイ効果をアクティブにすることができる．ディレイ時間やかかり具合といったパラメータは，ボイスジェスチャで制御できる.

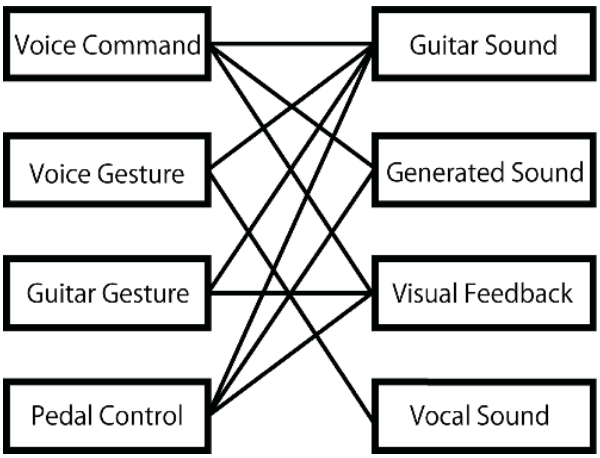

図 8. モダリティーの組み合わせ

5.1 ボイスコントロール

単語辞書には，ボイスコマンドの誤認識を避けるため，複数の発音パターンを登録した．ボイスコマンドは英語を基本としているが，日本語の話者が発声しているため，日本の音響モデルを使用した．ギターのパフォーマンスへの影響を考慮して，自然で短い言葉を使用した．表 1 にボイスコマンドとパラメータの対応関係の例を示す.

5.2 統合モジュールの設定

図8は，複数のモダリティの複合機能を示す．ギターサウンドは，ボイスコマンドとボイスジェスチャで制御できる．音声効果の種類はボイスコマンドによって選択され，そのパラメータはボイスジェスチャによって調整される.シーケンサは，ボイスコマンドによって，再生・停止・次のシーケンスへの切り替えなどの制御ができる．ビジュアルコントロールモジュールは，ボイスコマンド，ギタージェスチャ，フットペダルによって制御ができる．シーンの選択，オブジェクトの生成，色の選択，およびイベントのアクティブ化は，演奏者が制御する．「雨」のシーンの例は, 以下の通りである. このシーンを選択すると，雨の音のサンプルが再生される．波面などのビジュアルオブジ

ェクトは，強いギターのストロークの持続時間に応じて生成され，より多くの波面が作られると，ブラー効果が適用され激しさを映像的に表現する．

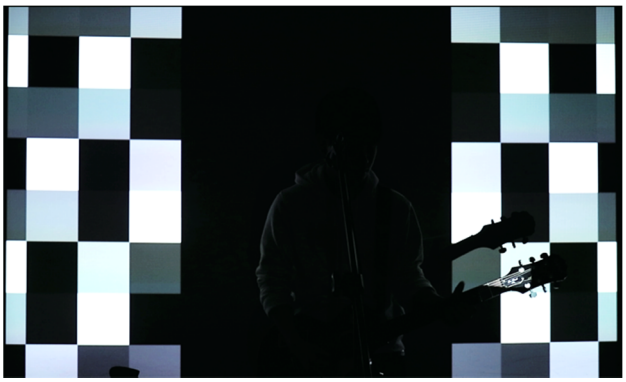

図9. シーン「石」の演奏の様子

図10 シーンごとのビジュアルオブジェクトの変化

5.3 インタビュー

ギター演奏者４名に本インタフェースを使ったライブパフォーマンスを依頼しインタビュー調査を行った．被験者は 21 歳から 23 歳（全て男性）のギター経験者で，ギター演奏経験は 1 年から 10 年間であった(表 2)．本番前 30 分程度の練習時間を設け，5 分程度の即興パフォーマンスをしてもらった．BPM は 118 に固定して行った． パフォーマンス後，すぐにインタビューを行った．

ボイスコマンドの操作が演奏の妨げになったかという問いに対して，「ならなかった」「ほとんどならなかった」「少しなった」「とてもなった」「ものすごくなった」の５段階で，演奏歴 10 年の A 以外の被験者は「少しなった」と答えた．被験者 B より「ギターを弾きながら歌っている程度」との意見を得た．被験者 A は，「とてもなった」と回答した．

ボイスコマンド操作のどんな点が演奏の妨げとなったかという質問に対して，「ボイスコマンドの誤動作があった場合，演奏の妨げになった」や「ボイコマンドが正しく認識されているのかペダルを踏んで実行するまでわからないので，次の展開が気になった」「認識遅延を考慮してボイスコマンドを言わないといけなかった」といった回答があった．また「next と言った時に変化がないと次の小節までまたないといけなかった」や「ペダルを踏むタイミングは分かりやすいが，ボイスコマンドをどのくらいの声量でしゃべるべきか難しい」といった回答があった．映像がギターエフェクトと同じようにペダル操作で扱えるのは楽しいといった回答や，映像によって演奏スタイルを変えた，演奏の強弱をつけられる映像シーンがなかったといった回答があった．演奏を続けることで，ライブペイントのように映像が塗られていくシーンが欲しいといった回答があった．また，どんなボイスコマンドをいうか忘れることはなかったが，迷うことはあったといった回答を得た．ボイスコマンドが増えた場合に，覚えるのは難しいといった回答があった．

表 3. 被験者（演奏者）一覧

被験者	年齢	性別	演奏歴(年)
A	23	男	10
B	21	男	6
C	21	男	3
D	22	男	1

5.4 考察

パフォーマンスでは，ボイスコマンドにより映像と音楽シーケンスを同じように操作することができた．しかしながら，ボイスコマンドを実行するまで，何もフィードバックがないため，認識がうまくいっているのかいないのか判断が難しいという問題があった．これにより，実行するためのフットペダルを連続して複数回踏む操作を誘発してしまっていた．さらに，機敏な操作が必要とされる演奏展開において，シーンの切り替えが間に合わないなどの問題があった．また，ボイスコマンドによるパラメータの操作は，演奏を少し邪魔することがわかった．被験者 B は「ギターを弾きながら歌っている程度だった」と回答している．本インタフェースを用いたオーディオビジュアルパフォーマンスの実演では，誤認識による操作ミスがあったが，30 分程度の練習でボイスコマンド・ペダル操作を使ったシステム操作への慣れがみられた．システムへの熟練度をあげることでより優れたオーディオビジュアルパフォーマンスが実現されることが期待される．また，練習による慣れは，ボイスジェスチャ操作でも見られた．被験者 A は，ボイスジェスチャによるパラメータ制御を摩擦音のような発声をすることで安定して音量によるパラメータ操作を実現していた．

また，未解決の問題は，音声認識辞書に登録するボイスコマンドの選択である．認識精度に加えて，発声されたフレーズの印象が演奏自体に影響することが考えられる．

6. まとめ

本論文では，JackTopGuitarと呼ばれる伝統的なギター演奏を基盤としたオーディオビジュアルパフォーマンスのためのデジタルミュージカルインタフェース(DMI)の紹介とライブコンサートを想定した音声認識インタフェースの認識テストを行った．このDMIは，ギターエフェクトの制御に音声認識インタフェースを導入することで音楽表現の拡張をした以前の我々の取り組み[1] を，ラップトップの持つ表現力をギター演奏に持ち込むことでエレキギター演奏を基にしたオーディオビジュアルパフォーマンスを実現するインタフェースとして発展させた．

先行研究では，ボイスコマンドによるギターエフェクタの音色の制御を実現した．本稿では，新しいDMIとして，マイク，エレキギター，フットペダルを組み合わせたオーディオビジュアルパフォーマンスのためのJackTopGuitarの設計と実装について述べた．本インタフェースは，ギター音色・音楽アプリケーション・映像アプリケーションの制御を可能にした．また，伝統的なギター演奏スタイルは保持したままの映像表現の制御を可能にした．さらに，サウンド表現へのフィードバックを可能にした．また，ライブパフォーマンスを想定した音声認識インタフェースの認識率の評価を行った．背景演奏の有無による差，およびスタンドマイクとヘッドセットマイクが認識率へ及ぼす影響，音響モデルの違いによる影響について，認識実験を行った．ライブパフォーマンス中のような音声入力に背景演奏が加えられた環境では，スタンドマイクの方が高い認識率が得られた．また，背景演奏が加えられた環境では，「Pedal Control」や「Rain Sound」のようなボイスコマンドは高い認識率を得やすいことが分かった．反対に「Back」「Start」「Stop」のようなボイスコマンドは認識が難しいことが分かった．

最後に，本インタフェースを用いたオーディオビジュアルパフォーマンスを4名の演奏者に演奏してもらい，インタビュー調査を行った．そのうち，いくつかの問題点について議論した．即興演奏に映像的なインスピレーションを与えるが，ボイスコマンドの誤認識のため，シーン展開に追いつかないことなどがわかった．音声認識の遅延の問題から採用したペダル操作によるボイスコマンドの実行は，慣れが必要なものの，演奏者4名とも30分程度の練習で操作を把握し，被験者Bより「ギターを弾きながら歌っている程度だった」との回答を得た．インタフェースへの成熟度による，さらなる演奏表現の拡張が期待できる．

また，認識精度と演奏自体への影響を考慮したボイスコマンドの選択が必要であることが分かった．

今後は，音声認識率および音楽表現力の向上を目指す．認識率の向上を検討するにあたって，認識結果を修正するフットペダル等を付け加えるなどを検討している．また，演奏下での音声認識精度を保ちつつ，パフォーマンスを考慮したボイスコマンドの選択に関する調査も進めていきたい．また，カメラや加速度センサー等を用いた演奏ジェスチャの利用も検討したい．

最後に，ライブパフォーマンスにデジタル技術を取り込むプロジェクトを通して，ミュージシャン・アーティスト・作曲家の創作ための表現技術の発展に貢献していきたい．

参考文献

[1] T. Otani and Y. Obuchi, “Voice controllable multimodal performance system,” in Proceedings of the 14th annual conference of Asia Digital Art and Design Association, 2017, pp. 107–110.

[2] 秋田祐哉，三村正人，河原達也．会議録作成支援のための国会審議の音声認識システム．電子情報通信学会論文誌，Vol. J93-D, No.9, pp.1736 -1744, 2010.

[3] T. Ogata and G. Weinberg, “Robotically augmented electric guitar for shared control,” NIME, 2017, pp. 487–488.

[4] C. Erdem, A. Camci, and A. Forbes, “Biostomp: A biocontrol system for embodied performance using mechanomyography,” NIME, 2017, pp. 65–70.

[5] M. Faulkner, VJ: Audio-Visual Art and VJ Culture: Includes DVD. Laurence King Publishing, 2006.

[6] S. W. Lee, G. Essl, and M. Martinez, “Live writing : Writing as a real-time audiovisual performance,” NIME, 2016, pp. 212–217.

[7] Max Cooper, Emergence, http://emergence.maxcooper.net/, 2017.

[8] Alva Noto, Ryuichi Sakamoto, Ensemble Modern, UTP_, 2008, CD+DVD, Raster-Noton.

[9] S. Fasciani and L. Wyse, “A voice interface for sound generators: adaptive and automatic mapping of gestures to sound,” NIME, 2012.

[10] D. Stowell and M. Plumbley, “Making music through real-time voice timbre analysis: machine learning and timbral control,” Ph.D. dissertation, 2010.

[11] T. Igarashi and J. F. Hughes, “Voice as sound: using non-verbal voice input for interactive control,” in UIST ’01 Proceedings of the 14th annual ACM symposium on User interface software and technology, 2001.

[12] A. Loscos and T. Aussenac, “The wahwactor: a voice controlled wah-wah pedal,” NIME, 2005, pp. 172–175.

[13] J. Deacon, “The development of a software tool that employs vocals for the control of musical elements in a live performance,” Ph.D. dissertation, University of Limerick, 2014.

[14] A. Lee and T. Kawahara, “Recent development of open-source speech recognition engine julius,” in Asia-Pacific Signal and Information Processing Association. Asia-Pacific Signal and Information Processing Association, 2009, pp. 131–137.

[15] Paul Spinrad and Melissa Ulto. The VJ Book Inspirations and Practical Advice for Live Visuals Performance, A Feral House Book, 2005, pp. 166-180.

[16] Taito, Harmpty, (https://itunes.apple.com/jp/album/harmpty-single/1436651746) , 2018.

[17] VoxForge Acoustic model (http://www.voxforge.org/), 2018.

大谷　泰斗

1995年生．2017年東京工科大学メディア学部メディア学科早期卒業．2018年東京工科大学大学院バイオ情報メディア研究科メディアサイエンス専攻博士前期課程在籍．サウンドデザイン・メディア表現技術に興味を持つ．

越智　景子

国立障害者リハビリテーションセンター研究所流動研究員(2011-2014)．同客員研究員(2014-2016)．国立情報学研究所特任研究員(2015-2016)．同特任助教(2016-2017)．2017 年より東京工科大学メディア学部助教．博士(情報理工学)．音声合成，韻律，音声分析，音声インタフェース，言語訓練の研究に従事．

大淵　康成

1966年生．1990年東京大学大学院理学系研究科物理学専攻修士課程修了．1992年同博士課程中退．1992年より2015年まで(株)日立製作所中央研究所および基礎研究所勤務．その間，Carnegie Mellon University客員研究員(2002-2003)，早稲田大学客員研究員(2005-2010)，クラリオン(株)(2013-2015)．2015年より東京工科大学メディア学部教授．博士(情報理工学)．