Vol.2016-MUS-112 No.20
2016/8/1

音声分析合成システムWORLDにより実時間音声合成を実現するための拡張と実装例

森勢 将雅[1,a)]

概要：筆者らは，Channel Vocoder の構造を踏襲した高品質音声分析合成システムについて研究を続けており，合成結果の品質向上，演算コストの低減や，音声合成技術を必要とするアプリケーションへ流用可能な関数群を提供してきた．本発表では，筆者が開発している音声分析合成システム WORLD について，音声パラメータ（ここでは基本周波数 (F0)，スペクトル包絡，非周期性指標をまとめて音声パラメータと呼称する）から音声波形を合成する処理部の実時間化を実現する実装アイディアと実装例について述べる．本実装では実時間音声合成用の構造体を定義し，いくつかの関数群により音声パラメータの逐次登録と N サンプル単位での合成を実現することとした．任意のフレーム数分の音声パラメータを逐次構造体にリンクする関数と N サンプル単位で波形を合成する関数により，音声パラメータを逐次リンクしながら波形を出力する機能を実現した．この機能により，音声合成ソフトウェアについて，ユーザのアクションをその場で合成結果に反映させるインタラクション機能の実現や，音声を用いた電子楽器への応用が期待される．本稿では，WORLD を実時間化するための実装の詳細と，実装したプログラムの合成速度について評価した結果，および応用アプリケーションに利用する際の注意点について述べる．

キーワード：音声分析，音声合成，実時間処理

1. はじめに

高品質な音声分析合成技術は，統計的音声合成 [1], [2] や歌声合成 [3]，声質変換 [4]，音声の統計的性質の解析や制御 [5] など，幅広い分野で利用されている．実時間で声質を制御して出力するアプリケーション [6] の場合，分析と合成の両方を実時間で完了する必要がある．一方，統計的音声合成や音声の統計的性質の解析に関しては，大量の音声を分析する必要はあるが，必ずしも実時間で分析を完了させる必要は無い．音声パラメータ（ここでは，基本周波数 (F0)，スペクトル包絡，非周期性指標をまとめて音声パラメータと呼称する）を逐次生成し，その音声パラメータから波形を実時間合成する機能を実装することで，アプリケーションの利便性を大幅に向上できる利点がある．

筆者らは，高精度な音声分析合成システムとして TANDEM-STRAIGHT [7], [8] と WORLD [9] を提案してきた．TANDEM-STRAIGHT は MATLAB と C 言語によるライブラリ [*1] として実装されており，WORLD は MATLAB と C++のソースコードを公開している [*2]．どちらも，合成に関しては音声パラメータを引数として合成された波形を出力とする実装になっており，音声パラメータから波形を逐次合成することが可能な実装とはなっていない．この制限は，オフライン処理で事足りるアプリケーションへの組み込みや，音声知覚の実験刺激生成では問題とはならない一方，合成結果をインタラクティブに操作するアプリケーションなどで利用することができないことを意味する．

本研究では，この問題に対処するため，音声合成部について N サンプル毎に波形を出力する機能を有する合成器の実装を目指した検討をしている．実装は C++で行い，インタラクティブに音声を操作するアプリケーションへの応用が容易となるよう工夫している．本稿では，従来の合成関数と比較して品質を損なわない仕様での実装のコンセプトと実装した関数群について説明し，音声合成にかかる時間について，音声パラメータから一括して波形を合成する従来の WORLD を対象とした評価を実施する．以下では，各アルゴリズムの概要について述べ，実時間合成において重要となる項目を整理する．

[1] 山梨大学，山梨県甲府市武田 4-3-11.
University of Yamanashi, 4-3-11, Takeda, Kofu, 400–8511, Japan.
[a)] mmorise@yamanashi.ac.jp

[*1] http://ml.cs.yamanashi.ac.jp/straight/

[*2] http://ml.cs.yamanashi.ac.jp/world/

ⓒ 2016 Information Processing Society of Japan

2. WORLD：基盤となる音声分析合成システム

2016年7月現在，WORLD [9] には，PLATINUM版 [10] とD4C版が存在するが，本実装はD4C版を対象に行っている．D4C版WORLDは，Vocoder [11] のアイディアを踏襲し，音声からF0，スペクトル包絡，非周期性指標の3つのパラメータを推定する．F0は，声帯振動間隔のうち最短の区間の逆数から与えられるパラメータであり，スペクトル包絡は，音声波形から求めたパワースペクトルからF0に起因する微細変動を取り除いたパラメータである．非周期性指標は，音声のパワーと音声中に含まれる非周期性成分のパワーの比として定義される．WORLDでは，それぞれのパラメータを独自の方法を提案して推定しており，特許性に問題が生じずソースコードは修正BSDライセンス下で自由に利用できる利点がある．

2.1 DIO: 高SNRの音声を対象とした高速・高精度なF0推定法

F0は，音声を構成する要素として最も基礎的なパラメータの1つであり，現在までに数多くの推定法が提案されている [12]．相関ベースの方法 [13] やCpestrum法 [14] が代表的であり，YIN [15] やPYIN [16] のように相関ベースの方法を改良した方法，あるいはスペクトルの特徴に着目した方法に改良を加えたSWIPE [17] など，高度化がなされている．また，調波構造の基本波抽出を用いた方法 [18] は，高SNRの音声が対象となるが，高速・高精度な推定が可能である．WORLDで採用しているDIO [19] は，声帯振動の時刻検出を用いた方法 [20] と基本波抽出法を改良して作られた方法であり，高SNRの音声に対象は限定されるが，高速かつ高精度なF0推定を実現する．

2.2 CheapTrick: 高精度なスペクトル包絡推定法

音声からのスペクトル包絡推定についても歴史は長く，線形予測分析 (LPC: Linear Predictive Coding) [21] やCepstrum [22] が提案されている．ただし，これらの方法は高品質音声合成向きではなく，主に音声符号化や音声認識で利用する方法として発展してきた経緯がある．高品質な音声合成に利用される方法では，STRAIGHT [23] やTANDEM-STRAIGHT [7], [8]，F0適応多重フレーム統合分析 [24] が提案されてきた．WORLDでは，筆者が開発したCheapTrick [25], [26] を採用している．CheapTrickは，ピッチ同期分析 [27] とCepstrum法の考え方を改良した方法であり，高品質音声合成を目指した既存の方法と比較して計算コストが低く，高品質な音声が合成できる利点がある．

2.3 D4C: 音声の揺らぎに頑健な非周期性指標推定法

音声は，声帯振動から成る周期的な成分だけではなく，非周期的な雑音成分も含む．高品質な音声合成を目指すため，mixed excitation [28] など，非周期的な成分を導入する方法が検討されてきた．非周期性指標は，音声波形に含まれる非周期的な成分のパワーと音声波形全体のパワーとの比として定義されるパラメータである．また，このパワー比は周波数により異なるため，帯域毎に与えることが多い [29]．非周期性指標推定の方法もいくつか検討されており [30], [31]，高品質音声合成を目的としたものも提案されている [32], [33]．D4C版WORLDでは，D4C [34] という群遅延に基づくパラメータから推定する方法を採用している．

2.4 音声パラメータからの波形合成

音声合成処理では，声帯振動が生じる時刻をF0軌跡から算出する．算出された時刻それぞれについてスペクトル包絡と非周期性指標を取り出し，周期性成分は最小位相応答，非周期性成分はホワイトノイズを励起信号として最小位相応答を畳み込むことで算出する．なお，F0の存在しない無声区間については，F0を500 Hzと設定し，全ての成分が非周期的であるという前提で処理をする．無声区間の合成区間もF0情報が利用されるが，本稿ではどちらも合わせて「合成イベント時刻」と呼称する．

実時間合成を行う際は，F0軌跡から合成イベント時刻を逐次求める処理をどのように実装するかが課題となる．声帯振動時刻さえ求めることができれば，スペクトル包絡と非周期性指標を取り出して合成する処理について，従来の実装をそのまま流用することが可能となる．

2.5 実時間合成への要求事項

具体的な目標は，音声パラメータからNサンプル単位で合成する機能の追加である．本研究では，現在の合成関数で出力される音声波形と等価な品質の音声合成を実現することも要求事項とする．ここでは，C++で実装しているWORLDに，実時間音声合成を行うための構造体と関数群を用意する形で実装する．本稿で説明するプログラムはGitHubで公開しており [*3]，修正BSDライセンス下で誰もが利用可能である．なお，本実装ではクラスを利用していないが，これは，C言語の機能のみを利用したいというリクエストがあるためである．

3. 実時間合成の実装例

ここでは，WORLDにより得られた音声パラメータから実時間合成を行う実装例について述べる．本実装は，実時間合成に必要となるWorldSynthesizer構造体と，6つの関

[*3] https://github.com/mmorise/World

ⓒ 2016 Information Processing Society of Japan

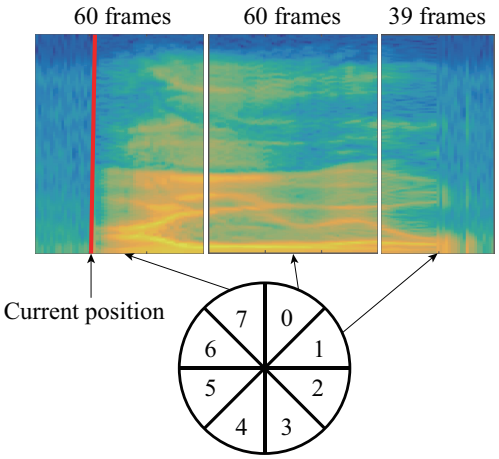

図 1 Synthesizer 構造体と音響パラメータの関係．下部の円は音声パラメータへリンクするためのポインタであり，この図では 8 つのリングバッファとして実装されている．Current position は，現在までに合成された波形を示す時刻である．この時刻に基づいて，合成に利用されることが無くなった音声パラメータへのリンクは自動的に破棄される．

数を駆使して実時間合成を実現する．

3.1 実時間合成用の構造体の導入

構造体を用いた実時間合成について，図 1 を用いて説明する．図 1 の下部にある 8 等分された円は，音声パラメータへのポインタを有するリングバッファである．本実装では，音声パラメータをリングバッファにリンクすることで，音声パラメータを逐次追加することが可能である．また，N サンプル毎に合成することで，図中の Current position が N サンプル分シフトする．シフトすることで以後アクセスされることが無いことが確定した音声パラメータは，自動的にリングバッファから外れるように実装している．

実時間合成処理は，WORLD により音声パラメータが求められた後に，以下のステップで行う．

(1) InitializeSynthesizer 関数で構造体を初期化
(2) AddParameters 関数で音声パラメータを構造体に追加
(3) Synthesize2 関数で N サンプル分の波形を合成
(4) DestroySynthesizer 関数でメモリを解放

(3) は 1 回につき N サンプルしか合成されないため，追加されている音声パラメータから合成可能なサンプル数分合成されるまで処理をループする．1 フレームずつリンクしつつ同時に逐次合成するアプリケーションを実装する場合，(2), (3) を繰り返して処理することとなる．以下では，具体的な流れについて順に説明する．

3.2 InitializeSynthesizer()：構造体の初期化

WorldSynthesizer 構造体は，はじめに本関数を用いて初期化される．構造体は，サンプリング周波数，1 回の合成により得られるサンプル数，ポインタ数（図 1 だと 8），WORLD で分析した際の分析シフト幅と FFT 長などの情報を有する．これらの情報は，初期化後に変更することはできない．

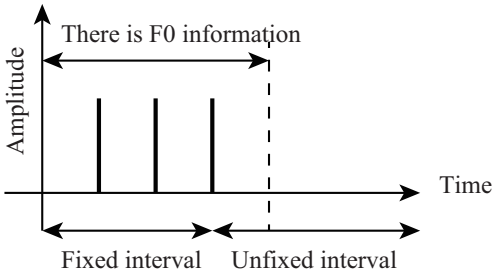

図 2 F0 の存在する区間と合成可能な区間（Fixed interval）との関係．この図は，声帯振動が生じる時刻にパルスを配置した例を示す．最後の声帯振動の次に生じる声帯振動の時刻は未知のため，合成可能な区間は，最後の声帯振動が生じた時刻までである．F0 がその時刻より先まで与えられたとしても，合成イベント時刻が確定するまでは合成可能とはならない．

3.3 AddParameter()：音声パラメータを構造体にリンク

構造体が初期化された後は，AddParameter 関数により音声パラメータをリングバッファにリンクする．音声パラメータが引数となるが，特色は，任意のフレーム数をまとめてリンクことが可能な仕様で実装されている点にある．図 1 の例では，60 フレーム，60 フレーム，39 フレームの音声パラメータをそれぞれリンクしている．すでにリングバッファがフルの場合は，何も行われずにエラーを示す戻り値が得られる．この関数が呼ばれる毎に，合成イベント時刻が修正され，現在合成可能な音声波形のサンプル数が自動的に更新される．

3.4 Synthesis2()：N サンプル毎に合成

現在時刻から N サンプル以上の合成が可能である場合，Synthesis2 関数により N サンプルの合成が実施される．1 回合成されるたびに，合成完了したサンプル数と AddParameter 関数でリンクされた音声パラメータの時刻から，今後アクセスされることが無い音声パラメータをリングバッファから自動的に削除する処理も行う．合成可能なサンプル数が N 未満の場合，本関数は何もせず合成がなされなかったことを示す戻り値が得られる．

AddParameter 関数では，任意のフレーム数を 1 回の呼び出しでリンクするため，逐次音声パラメータを追加しつつ合成する場合でも，AddParameter と Synthesis2 の呼び出し回数は 1 対 1 とはならない．また，合成可能なサンプル数は，図 2 のように，音声パラメータが存在する時刻とは一致しないことに注意が必要である．

3.5 DestorySynthesizer()：構造体のメモリを解放

合成処理の終了後は，構造体で内部的に確保したメモリ

ⓒ 2016 Information Processing Society of Japan

を解放する必要がある．DestorySynthesizer 関数は，構造体内部で利用した全てのメモリを解放する．一方，サンプリング周波数等の条件が等しく，現在リンクされている音声パラメータのみ破棄したい場合は，RefreshSynthesizer 関数を呼び出すことで，リンクされた音声パラメータのみ初期化することが可能である．

3.6 IsLock()：構造体の状態を確認

本実装では，初期化の段階で音声パラメータをリンクするリングバッファのバッファ数を指定することが可能である．この仕様は，図 3 のように構造体に音声パラメータをリンクすることも合成することもできない状態（本稿ではこれを「ロック状態」と呼称している）になる危険性がある．一度この状態にはまった場合，RefreshSynthesizer 関数か DestorySynthesizer 関数で構造体をリセットすることを余儀なくされる．IsLock 関数は，ロック状態を検出するために実装された関数であり，この関数が True を返した場合は，何らかの手段で構造体をリセットする必要がある．

ロック状態を抑止する対策としては，以下の 3 つが有効な手段となる．

- 構造体の初期化時にリングバッファのバッファ数を多く確保する
- AddParameter 関数呼び出し時にリンクするフレーム数を増やす
- AddParameter 関数でリンクする前に F0 の下限を確定させる

F0 が低いほど次の声帯振動が生じる時刻までの間隔が長くなり，ロック状態を引き起こしやすくなる．3 番目の手段は，インタラクティブな操作により F0 が極端に低い場合ロック状態を引き起こす問題を抑止する効果がある．また，合成可能性のある F0 を事前に確定させ，フレームシフトと毎回リンクするフレーム数から必要となるリングバッファのバッファ数を決定することで，ロック状態は確実な回避が可能である．

4. 合成速度に関する評価

本実装では，合成イベント時刻を従来の合成関数と完全に一致させているため，合成結果は従来の WORLD と概ね一致する．実質的な差は非周期性成分の合成におけるホワイトノイズ生成部であり，これが品質に与える影響は存在しないといえる．品質評価が不要であるため，本稿では合成速度に関する評価を中心に行い，有効性について論じる．

4.1 評価に用いる音声と実験条件

声帯振動の回数と FFT 回数が比例関係にあり，FFT が合成処理の実質的なボトルネックになるため，声帯振動回数の多い F0 が高い音声ほど合成に時間がかかる．今回は，1 つの目安として女性発話音声を対象に分析を行い，結果

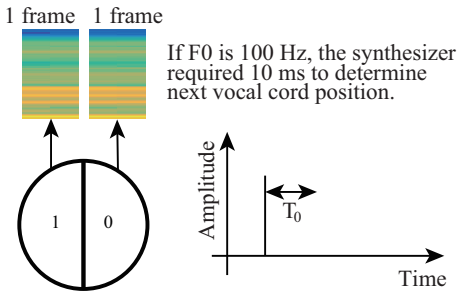

図 3 Synthesizer 構造体のロック状態．例えば，F0 が 100 Hz の場合次の声帯振動が生じるまで 10 ms 程度先の音声パラメータを必要とするが，リンクされている音声パラメータのフレーム数が 10 ms 分に満たない場合生じる．AddParameters 関数で音声パラメータを追加できず，Synthesis2 関数で合成もできないため，構造体をリフレッシュすることでリセットする必要がある．

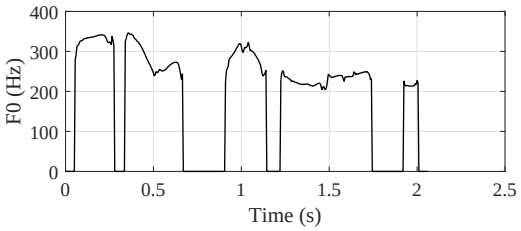

図 4 実験に用いた音声の F0 軌跡．女性発話で「コーヒーにミルクを入れますか」と発音している．

から有効性を考察する．分析に用いた音声は，約 2 秒の「コーヒーにミルクを入れますか」という文章の発話音声であり，サンプリング周波数は 48 kHz である．図 4 は分析された F0 軌跡であり，下限が 204 Hz で上限が 346 Hz である．また，FFT 長は 2,048 サンプルに設定している．

実時間合成部に関しては，1 回 Synthesis2 関数を呼び出すたびに 256 サンプル (5.33 ms) 合成されるように構造体を初期化した．本稿では，Real time factor (RTF) を用いた速度の評価を，従来の音声パラメータから一括して合成した場合，および本実装で N サンプル毎に合成した場合について比較する．また，Synthesis2 関数 1 回あたりにかかる処理時間も計測し，分布を確認することでアプリケーションへ組み込む際の注意点について述べる．実験には，Let's note CF-SX2 (i7-3540M 3.00 GHz, 16 GB メモリ) を利用した．

4.2 現 WORLD の合成関数と比較した速度の評価

まず，音声波形全体の合成にかかった時間について示す．図 5 は，従来の WORLD と，本稿で実装した実時間合成について，横軸を RTF とした棒グラフである．実時間処理を導入することにより，RTF は 15.7%低下している．これは，Synthesis2 関数を呼び出すごとに，内部パラメータの調整を行う処理が含まれるためである．しかしながら，それでも RTF は 0.1 未満におさえられているため，イン

ⓒ 2016 Information Processing Society of Japan

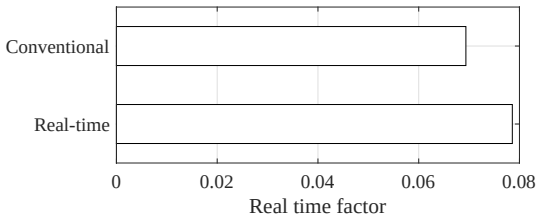

図 5 2 秒の女性声を合成した例に対する RTF の結果．実時間処理は従来の合成処理と比較して，15.7%速度が低下している．

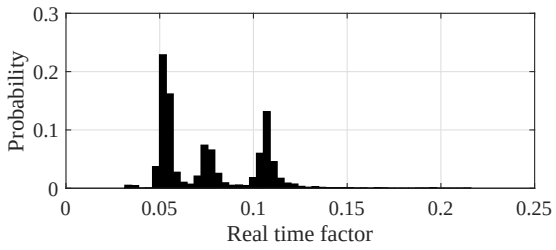

図 6 Synthesis 関数 1 回あたりの処理時間（RTF）のヒストグラム．ヒストグラムには，主に 3 つのピークが観測できる．今回の条件では，5.3 ms の間に最低 1 回，最大 3 回の声帯振動時刻が存在し，各ピークは声帯振動時刻の数に比例している．

タラクティブに音声を合成するアプリケーションへの組み込みに問題は生じないといえる．

次いで，Synthesis2 関数を 1 回呼び出す毎にかかる時間を分析する．図 6 は，1 回関数を呼び出す毎に 5.33 ms 合成する条件下において，1 回あたりの RTF を求めヒストグラムとして表示したものである．同条件で 10 回実行することで，他のプロセスによる CPU 利用率変動の影響を低減している．

ヒストグラムには，主に 3 つのピークを観測することができる．これは，1 回の合成関数呼び出しで合成される 5.33 ms 中に含まれる声帯振動回数に対応するピークである．WORLD では，声帯振動を伴わない無声区間は，F0 が 500 Hz と設定しており，F0 が 500 Hz の無声音区間では，5.33 ms の間に 3 回分の波形生成を行う可能性がある．一方，F0 の下限が 204 Hz であることは，声帯振動の最大間隔がおよそ 235 サンプルであることを示す．これは，256 サンプル毎に合成を行う場合，1 回の関数呼び出し中に最低 1 回の波形生成処理が含まれることを示す．ただし，3 回の声帯振動波形を生成する最悪の条件でも RTF は 0.1 程度であることから，本実験で用いたノート PC 程度のスペックであっても，実時間合成は問題無く実現できるといえる．

4.3 考察

本実装は，音声波形を一括で合成する場合と比較して 15.7%の速度低下が認められたが，RTF の観点から実時間合成は可能であるといえる．AddParameters 関数は，Synthesis2 関数と比較して処理時間が誤差の範囲であったため，音声パラメータを逐次リンクしつつ合成する実装においても，合成にかかる処理時間に差は生じない．音声合成の処理時間は声帯振動を合成する回数に比例するため，概ね合成対象となる音声パラメータの平均 F0 と比例関係にある．無声区間については F0 を 500 Hz と仮定しているが，本評価結果は，1 kHz を上回る高さの音声でも実時間処理が可能であることを示している．なお，現在の有声音の合成は，周期性成分と非周期性成分逐次処理で合成しているが，並列で合成することも可能であるため，速度の更なる最適化は可能である．特に，実時間で声質を変換し出力するアプリケーションの実装では，合成処理のみに CPU を占有させることは好ましくないため，RTF を可能な限り下げる最適化は，声質変換にかける時間を確保する重要な意味がある．

実時間合成に関して，本実装により要求事項は満足したと考えているため，次なる課題は実時間分析であるといえる．WORLD は，入力サンプル数から出力される音声パラメータのフレーム数が一意に決定できないことが問題である．実時間分析機能が実装されれば，現状の品質を維持した realtime STRAIGHT の改良版ともいえるアプリケーションや，実時間声質変換アプリケーションへの応用が期待できる．実時間分析に関しても，分析の信号処理理論については現在の WORLD の実装を流用できるため，実装の仕様を考えることが重要な課題となる．現在は仕様策定中であり，実時間合成と同様に分析器に相当する構造体を用意することで，従来の分析法と等価な品質での実時間分析を目指している．現在，実時間分析にも対応した WORLD をコードネーム「TenebrariusWORLD」として開発中であり，完成し次第 GitHub でリリースする見通しである．

5. おわりに

本稿では，音声分析合成システム WORLD で得られた音声パラメータを用いて実時間音声合成を実現する方法について説明した．実装例では，realtime STRAIGHT のように品質が劣化することが無く，WORLD の品質をそのまま実時間合成することを可能にした．プログラムのソースコードも配布しており，修正 BSD ライセンスを採用しているため，STRAIGHT Library と比較しても使いやすいといえる．合成速度は，現在の WORLD よりも 15.7%の低下が認められたが，1 フレーム単位でパラメータを与え逐次合成する場合においても，実時間処理が可能であることを示した．

次のステップでは，実時間声質変換の鍵となる分析合成を実現するため，実時間分析を行う拡張が必要となる．本実装を拡張することで，実時間で歌声を加工しつつ演奏するような電子楽器の実現にも取り組むことを計画している．実時間分析合成のアプリケーションは実環境で動作させることも想定されることから，ある程度雑音を含む音声

ⓒ 2016 Information Processing Society of Japan

から高精度な F0 を推定可能な方法も必要といえる．

謝辞 本研究は，科研費 15H02726, 16H05899, 16K12511, 16K12464 の支援を受けて実施された．

参考文献

[1] Zen, H., Tokuda, K. and Black, A. W.: Statistical parametric speech synthesis, *Speech Communication*, Vol. 51, No. 11, pp. 1039–1064 (2009).

[2] Koriyama, T., Nose, T. and Kobayashi, T.: Statistical parametric speech synthesis based on gaussian process regression, *IEEE Journal of Selected Topics in Signal Processing*, Vol. 8, No. 2, pp. 173–183 (2014).

[3] Nakamura, K., Oura, K., Nankaku, Y. and Tokuda, K.: HMM-based singing voice synthesis and its application to Japanese and English, *in Proc. ICASSP2014*, pp. 265–269 (2014).

[4] Ohtani, Y., Toda, T., Saruwatari, H. and Shikano, K.: Maximum likelihood voice conversion based on GMM with straight mixed excitation, *in Proc. ICSLP*, pp. 2266–2269 (2006).

[5] Kobayashi, K., Toda, T., Doi, H., Nakano, T., Goto, M., Neubig, G., Watiasri, S. S. and Nakamura, S.: HMM-based singing voice synthesis and its application to Japanese and English, *IEICE Trans. Inf. & Syst.*, Vol. E97-D, No. 6, pp. 1419–1428 (2014).

[6] Banno, H., Hata, H., Morise, M., Takahashi, T., Irino, T. and Kawahara, H.: Implementation of realtime STRAIGHT speech manipulation system, *Acoust. Sci. & Tech.*, Vol. 28, No. 3, pp. 140–146 (2007).

[7] Kawahara, H., Morise, M., Takahashi, T., Nisimura, R., Irino, T. and Banno, H.: TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation, *in Proc. ICASSP2008*, pp. 3933–3936 (2008).

[8] Kawahara, H. and Morise, M.: Technical foundations of TANDEM-STRAIGHT, a speech analysis, modification and synthesis framework, *SADHANA - Academy Proceedings in Engineering Sciences*, Vol. 36, No. 5, pp. 713–728 (2011).

[9] Morise, M., Yokomori, F. and Ozawa, K.: WORLD: a vocoder-based high-quality speech synthesis system for real-time applications, *IEICE Trans. Inf. & Syst.*, Vol. E99-D, No. 7, pp. 1877–1884 (2016).

[10] Morise, M.: PLATINUM: A method to extract excitation signals for voice synthesis system, *Acoust. Sci. & Tech.*, Vol. 33, No. 2, pp. 123–125 (2012).

[11] Dudley, H.: Remaking speech, *J. Acoust. Soc. Am.*, Vol. 11, No. 2, pp. 169–177 (1939).

[12] Hess, W.: *Pitch determination of speech signals*, Springer-Verlag (1983).

[13] Ross, M., Shaffer, H., Cohen, A., Freudberg, R. and Manley, H.: Average magnitude difference function pitch extractor, *IEEE Transactions on acoustic, speech, and signal processing*, Vol. ASSP-22, No. 5, pp. 353–362 (1974).

[14] Noll, A.: Short-time spectrum and “cepstrum” techniques for vocal pitch detection, *J. Acoust. Soc. Am.*, Vol. 36, No. 2, pp. 269–302 (1964).

[15] Cheveigné, A. and Kawahara, H.: YIN, a fundamental frequency estimator for speech and music, *J. Acoust. Soc. Am.*, Vol. 111, No. 4, pp. 1917–1930 (2002).

[16] Mauch, M. and Dixon, S.: PYIN: A fundamental frequency estimator using probabilistic threshold distributions, *in Proc. ICASSP2014*, pp. 659–663 (2014).

[17] Camacho, A. and Harris, J. G.: A sawtooth waveform inspired pitch estimator for speech and music, *J. Acoust. Soc. Am.*, Vol. 124, No. 3, pp. 1638–1652 (2008).

[18] Kawahara, H., Morise, M., Nisimura, R. and Irino, T.: Higher order waveform symmetry measure and its application to periodicity detectors for speech and singing with fine temporal resolution, *in Proc. ICASSP2013*, pp. 6797–6801 (2014).

[19] 森勢将雅，河原英紀，西浦敬信：基本波検出に基づく高 SNR の音声を対象とした高速な F0 推定法，電子情報通信学会論文誌 D， Vol. J93-D, No. 2, pp. 109–117 (2010).

[20] Yegnanarayana, B. and Murty, K.: Event-based instantaneous fundamental frequency estimation from speech signals, *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 17, No. 4, pp. 614–624 (2009).

[21] Atal, B. S. and Hanauer, S. L.: Speech analysis and synthesis by linear prediction of the speech wave, *J. Acoust. Soc. Am.*, Vol. 50, No. 2B, pp. 637–655 (1971).

[22] Oppenheim, A. V.: Speech analysis-synthesis system based on homomorphic filtering, *J. Acoust. Soc. Am.*, Vol. 45, No. 2, pp. 458–465 (1969).

[23] Kawahara, H., Masuda-Katsuse, I. and de Cheveigné, A.: Restructuring speech representations using a pitch-adaptive timefrequency smoothing and an instantaneous-frequency-based F0 extraction, *Speech Communication*, Vol. 27, No. 3–4, pp. 187–207 (1999).

[24] Nakano, T. and Goto, M.: A spectral envelope estimation method based on F0-adaptive multi-frame integration analysis, *in Proc. SAPA-SCALE 2012*, pp. 11–16 (2012).

[25] Morise, M.: CheapTrick, a spectral envelope estimator for high-quality speech synthesis, *Speech Communication*, Vol. 67, pp. 1–7 (2015).

[26] Morise, M.: Error evaluation of an F0-adaptive spectral envelope estimator in robustness against the additive noise and F0 error, *IEICE Trans. Inf. & Syst.*, Vol. E98-D, No. 7, pp. 1405–1408 (2015).

[27] Mathews, M. V., Miller, J. E. and David, E. E.: Pitch synchronous analysis of voiced sounds, *J. Acoust. Soc. Am.*, Vol. 33, No. 2, pp. 179–186 (1961).

[28] McCree, A. and Barnwell, T.: A mixed excitation LPC vocoder model for low bit rate speech coding, *IEEE Trans. on Speech Audio Processing*, Vol. 3, No. 4, pp. 242–250 (1995).

[29] Lin, W., Koh, S. N. and Lin, X.: Mixed excitation linear prediction coding of wideband speech at 8 kbps, *in Proc. ICASSP'00*, Vol. 2, pp. 1137–1140 (2000).

[30] Griffin, D. W. and Lim, J. S.: A new model-based speech analysis/synthesis system, *in Proc. ICASSP1985*, Vol. 10, pp. 513–516 (1985).

[31] Griffin, D. W. and Lim, J. S.: Multiband excitation vocoder, *IEEE Trans. on Acoust., Speech, and Signal Processing*, Vol. 36, No. 8, pp. 1223–1235 (1988).

[32] Kawahara, H., Morise, M., Takahashi, T., Banno, H., Nisimura, R. and Irino, T.: Simplification and extension of non-periodic excitation source representations for high-quality speech manipulation systems, *in Proc. INTERSPEECH2010*, pp. 38–41 (2010).

[33] Kawahara, H. and Morise, M.: Simplified aperiodicity representation for high-quality speech manipulation systems, *in Proc. ICSP2012*, pp. 579–584 (2012).

[34] 森勢将雅：帯域毎の非周期性指標推定法とその誤差評価，電子情報通信学会技術研究報告，Vol. 115, No. 99, pp. 13–18 (2015).

ⓒ 2016 Information Processing Society of Japan