Vol.2021-DPS-189 No.5
2021/12/20

AutoVCを用いた ゼロショットリアルタイム声質変換手法の提案

鈴木 大志[†1]　鷹合 大輔[†1]　中沢 実[†1]

概要：声質変換とは，人物の声の声質のみを別人の声質に変換する技術である．その中でも，ゼロショット声質変換は，変換モデルの学習した音声にない声質間での変換が可能な手法である．AutoVC は，ゼロショット声質変換モデルで，入力話者の声のメルスペクトログラムと入出力話者の話者埋め込みベクトルを入力する事で，話者らの声を学習しているか否か関わらず，出力話者の声質のメルスペクトログラムを出力する．これを，音声波形に復元する際に，音既存手法では WaveNet や Griffin-Lim などの多くの計算時間を要する手法を用いておりリアルタイムな声質変換の弊害となっている．そこで，本研究ではメルスペクトログラムに代えて，スペクトル包絡を用いた．そして，波形の復元は WORLD を用いる事でリアルタイムな声質変換を実現した．

キーワード：Voice Conversion, Zero-Shot, Real-Time, Deep Learning

A Study of Zero-Shot Real-Time Voice Conversion Method Using AutoVC

1. はじめに

近年，深層学習を用いた声質変換技術の発展により，人物の声を高精度で別の特定の人物の声質に似た声質に変換することが可能になってきた．この中でも，AutoVC[1] をはじめとした，ゼロショット声質変換技術という，学習データに含まれるか否かに関わらず声質変換が可能な技術を用いることで，変換の際に，変換目標話者の声質を表す埋め込みベクトル，変換元話者の声質を表す埋め込みベクトルなどを入力すると，変換元話者の音声の声質のみを指定した変換目標話者の声質へ変換することが可能である．このような声質変換は，実際にアプリケーションとして，配信での利用やボイスチャットや電話などのリアルタイムで使用されることが考えられる．このような場合において，コミュニケーションを阻害しない速度での声質変換が求められ，これをリアルタイムな声質変換と定義する．また，ゲームなどと同時に動作させたり，スマートフォンなどの計算資源の少ない環境での動作も求められる．しかし，多くのゼロショット声質変換モデルではこの問題を解決する事は出来ない．その要因として，メルスペクトログラムをはじめとした，声質変換に用いた特徴量を音声波形に変換する際に，WaveNet[2] や Griffin-Lim[3] などの処理に時間のかかる手法を用いたり，WaveGlow[4] のような計算コストのかかる深層学習モデルを用いた手法を使うことがその原因となっていると考えた．そこで，本研究では AutoVC の入力にメルスペクトログラムではなく，スペクトル包絡を用いた．そして音声波形への変換時に WORLD[5] を用いて，深層学習手法を用いない高速な復元が可能になる手法を提案する．

本研究における貢献内容は，AutoVC の問題点である変換時の韻律の不安定さの解消，コミュニケーションにおいて支障のない程度の遅延の低減を目的とする．

本論文の構成は次の通りである．2. で関連研究について述べ， 3. で提案手法について述べ， 4. で評価実験について述べ, 5. でまとめについて述べる．

2. 関連研究

2.1 声質変換の仕組み

声質変換とは，ある人物の発した声の声質のみを別の話

†1 現在，金沢工業大学
Presently with Kanazawa Institute of Technology

者の声質に変換することのできる技術である．その中にもいくつかの手法があり，隠れマルコフモデルなどを用いた統計的手法や深層学習を用いた手法などが知られている．今回研究に用いた声質変換モデルは，深層学習を用いた手法の中でもオートエンコーダという技術を用いている．

その中でも画期的であった手法として，非並列データ用いた学習が可能な声質変換手法がある．それ前の手法では，変換元の話者の音声と変換先の話者の音声の発声タイミングや発話内容などを細かく合わせないと学習できなかった．それに対し，非並列データ用いた学習が可能な声質変換手法では，発声タイミングや発話内容が違っていても学習が可能で，CycleGAN を用いた手法 [11] などがある．

声質変換手法の中の分け方として，(1) 一対一の声質変換，(2) 多対多の声質変換，(3) ゼロショット声質変換などがある．(1) の場合，特定の人物 A の声質を特定の人物 B の声質に変換できる．ただし，A 以外の人物の声を入力として用いることはできない上，B 以外の声質での出力はできない．この方式については，CycleGAN-VC[11] などが知られている．(2) の場合, 学習に用いる人物を複数人にでき，手法によってはその双方向を変換できる手法である．しかしながら，学習に用いていない声質への変換はできない．この方式については，SterGAN-VC[12] などが知られている．(3) の場合，学習にその声質の音声含まれているかの有無に関わらず変換が可能となる．ただし，どのような声質変換にするかを示す値を入力しなければならない．

2.2 AutoVC

2.2.1 AutoVC とは

AutoVC は，非並列データ用いて学習された，多対多の AutoEncoder ベースの声質変換モデルで，入力には入力話者のメルスペクトログラムと，入力話者を声質を表すワンホットベクトル，出力話者の声質を表すワンホットベクトルを入力することで，入力されたメルスペクトログラムに含まれる入力話者の声質を出力話者を示すワンホットベクトルで指定した人の声質に変換することができるモデルである．学習時の入力は入力話者のメルスペクトログラムと入力話者のワンホットベクトルを入力とし，出力話者のワンホットベクトルは入力話者と同じものを入れることで，入力したメルスペクトログラムと出力したメルスペクトログラムの平均二乗誤差を損失関数として用いる．また，中間層にボトルネック構造があり，ボトルネックの出力と，出力されたメルスペクトログラムを再度入力に入れた時のボトルネックの出力の平均二乗誤差も損失関数として利用している．AutoVC は入力にワンホットベクトルではなく，話者の声質を表す埋め込みベクトルを用いて学習することで，学習に用いていない声質間の変換が可能なゼロショット声質変換モデルとして用いることができる．

図 1 は，ゼロショット時の AutoVC のモデル構造を示している．(a) は Content Encoder と言い, 一般的な AutoEncoder でいうエンコーダにあたる部分を指す. ただし, Down1 と Down2 の部分がボトルネック構造になっていることが特徴である．また，$E_s(\bullet)$ は，(b) と同様である．(b) は Style Encoder と言い，話者埋め込みベクトルを生成する機構である．ただし，ゼロショット以外での利用の際は，これを用いず，直接ワンホットベクトルを入力する．(c) は Decoder といい，一般的なオートエンコーダのデコーダと同様の機能を持つ．(d) は，Spectrogram Inverter と言い，PostNet 及び WaveNet で構成されたメルスペクトログラムを音声波形に変換するための機構である．(e) 及び (f) はそれぞれ Down1 と Down2 のボトルネック構造を示している．

本研究でも AutoVC を利用しているが，WaveNet 及び Style Encoder はインターネットで公開されている学習済みモデルを利用し，それぞれ wavenet vocoder[13] と Resemblyzer[14] を利用している．

2.2.2 話者埋め込みベクトル

話者埋め込みベクトルとは，話者認証などの分野で用いられる，声質間の類似度を判断するための固定長のベクトルのことである．AutoVC においては図 1 の (b) の Style Encoder 部分で生成されるベクトルである．word2vec[6] と同様に，浅いニューラルネットワークを学習させることで，音声データを話者埋め込みベクトルに変換するモデルを作成できる．

2.2.3 AutoVC の問題点

AutoVC の問題点として, 韻律情報をうまく変換できず音程が大きく上下に揺らぐことがある [7]．これは，AutoVC のボトルネック層で入力話者の韻律情報がそのままデコーダーに引き継がれ，入力話者の基本周波数 (以下 F0) と出力話者の F0 との間で反転を繰り返してしまうためだと考えられている．

また，変換後のメルスペクトログラムを音声波形に復元する際に，自己回帰モデルである，WaveNet を用いているためにリアルタイム化ができていない．

2.3 既存の解決手法

2.3.1 F0-conditioned AutoVC

AutoVC の韻律が維持できない問題点を解決した手法として，F0-conditioned AutoVC[7] が知られている．この手法では, 韻律の維持が困難であることに対し, 以下のような手法を用いて解決している．あらかじめ，入出力話者の F0 の対数平均と対数分散を算出する．Logarithm Gaussian (LG) normalized transformation[8] を用いて, 入力話者の F0 を出力話者の F0 の分布に変換する．この特徴量をデコーダーに補助特徴量として入力することで，韻律の維持を可能にしている．この手法の場合，韻律の維持は可能になっているが，F0 を補助特徴量にしていることにより計

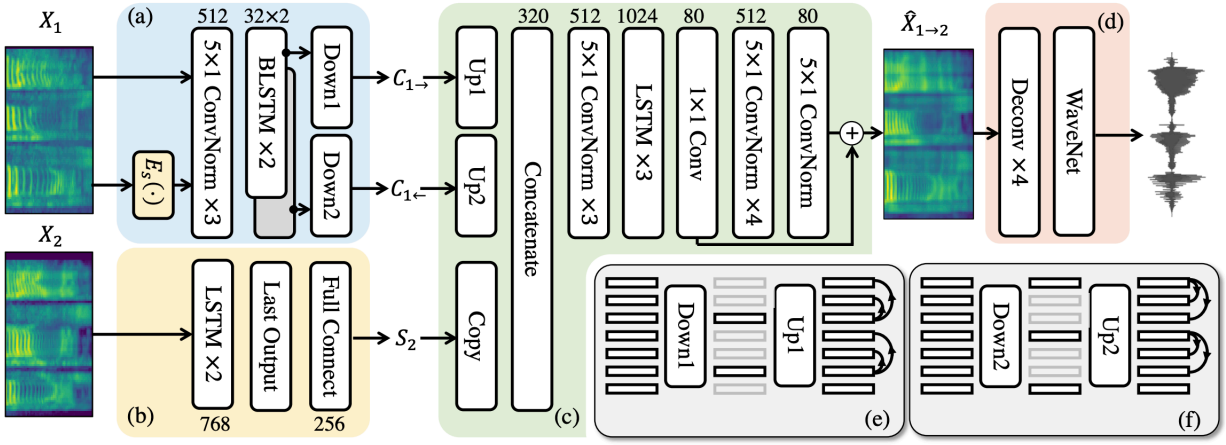


図 1 AutoVC モデル構造 [1]Figure 3. より引用

算量が増えてしまいリアルタイム化が困難になっている．

2.3.2 ConVoice

ゼロショット声質変換手法のリアルタイム化した手法として，ConVoice[9] が知られている．この手法では，メルスペクトログラムを音声波形への復元をリアルタイムに行うために自己回帰モデルではなく，Flow-based 生成モデルである，WaveGlow を用いてリアルタイム化を行っている．この手法のデメリットとして，リアルタイム化のために深層学習手法を用いているため，計算資源の少ないスマートフォンなどでの動作が困難である．

3. 提案手法

3.1 提案手法の概要

まず，準備段階として，入出力話者の話者埋め込みベクトルと F0 の対数平均と対数分散をあらかじめ生成しておく．次に，前処理として WORLD を用いて音声波形を F0，スペクトル包絡，非周期性指標の三つの特徴量に分解する．F0 は 2.3 節で述べた手法と同様の手順で F0 の分布を出力話者の分布に変換する．スペクトル包絡は正規化した後，スペクトル包絡用の AutoVC を用いて声質変換をし，正規化した値を元に戻す．最後に，変換した F0，変換したスペクトル包絡，変換していない非周期性指標を WORLD を用いて音声合成し，波形を生成する．

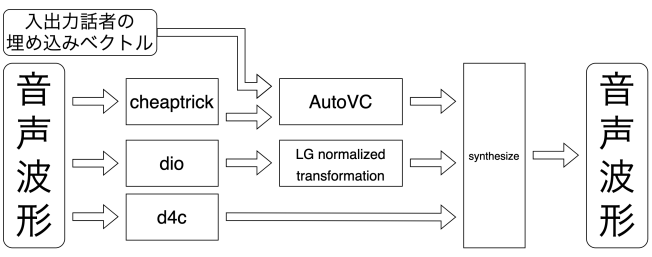


図 2 提案手法

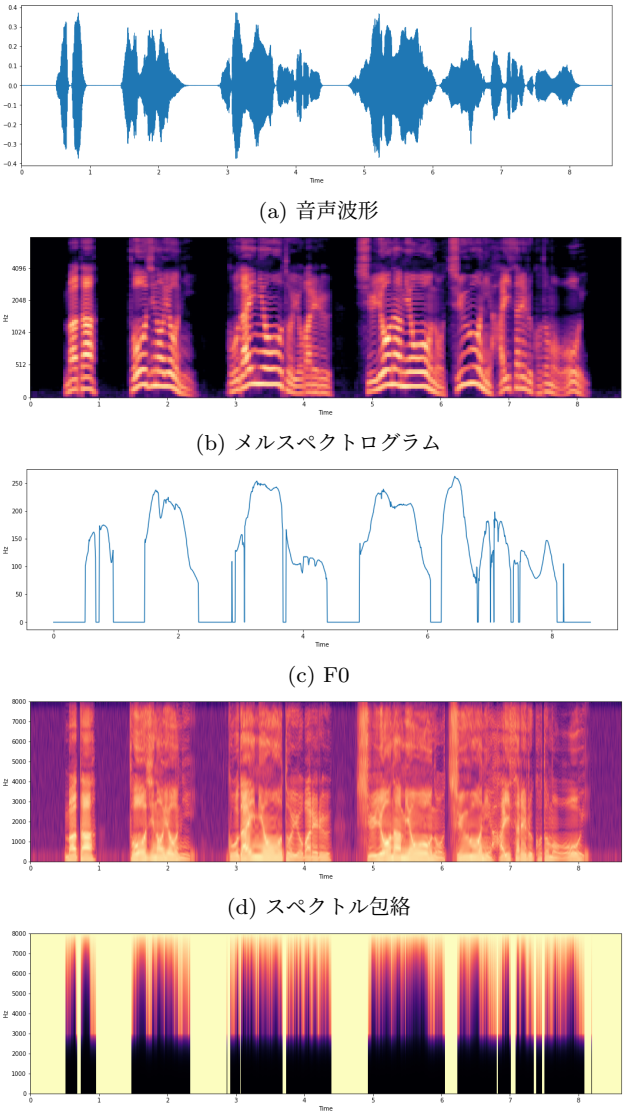
この手法を用いることで，音声波形への復元の際に低速であったり，多くの計算資源の消費してしまうという問題を解決し，高速化を実現できる．また， AutoVC での変換をスペクトル包絡に対してのみとすることによって，F0 の情報が失われる心配がなく安定した韻律の変換を実現する．

3.2 特徴量

本研究で用いる主な特徴量は (1) 音声波形， (2) メルスペクトログラム，(3)F0, (4) スペクトル包絡，(5) 非周期性指標の 5 種である (図 3)．(1) はアナログ音声を 16kHz モノラル 16bit でサンプリングしたデータ系列である．(2) は (1) を短区間フーリエ変換したものをメルフィルタバンクとかけることで，人間の聴覚特性に基づいた情報の圧縮をした特徴量である．(3) は WORLD で用いる特徴量で基本周波数ともいい，本研究では声の音程を扱うために用いている．(4) は WORLD で用いる特徴量で，音声スペクトルの大まかな形を表す包絡線のことであり，本研究では声質情報や言語情報を扱うために用いている．(5) は WORLD で用いる特徴量で，声のかすれなどを扱うために用いている．

3.3 WORLD

WORLD は，肉声と区別できないほど高い品質での音声合成を実現するために作られた音声分析合成システムである．そのうち本研究では，(1)F0 を推定する Dio 関数，(2)F0 の推定を補助する StoneMask 関数，(3) スペクトル包絡を推定する CheapTrick 関数，(4) 非周期性指標を推定する D4C 関数，(5)F0，スペクトル包絡，非周期性指標の三特徴量から音声波形を生成する Synthesis，の 5 つの関数を用いている．



(a) 音声波形

(b) メルスペクトログラム

(c) F0

(d) スペクトル包絡

(e) 非周期性指標

図 3 本論文で用いる 5 種の特徴量

3.4 データセット

本研究では，JVS (Japanese versatile speech) corpus[10] を用いて学習及び評価実験を行なった．

このコーパスは，100 人のプロフェッショナル話者によって parallel100(話者間で共通する読み上げ音声 100 発話)，nonpara30(話者間で全く異なる読み上げ音声 30 発話)，whisper10(ささやき声 10 発話)，falsetto10(裏声 10 発話) があり，本研究では parallel100 を用いて学習と 評価検証を行なった．

3.5 モデル構造

本研究では，図 1 をベースに入力する特徴量を変更した．それによりいくつかのパラメータに変更を加え 3 つのモデルを作成した．変更内容として入力をする特徴量をメルスペクトログラムからスペクトル包絡に変更したため周波数方向の次元数が 80 から 513 に増加した，それにより図 1 の (c) の部分の Conv や ConvNorm のフィルタ枚数を 80 から 513 に変更した (以下，提案手法 I)．また，更なる精度向上のために，提案手法 I の変更に追加して，図 1 の (a) と (c) の畳み込み層のフィルタ枚数のうちフィルタ枚数が 512 枚のものを 1024 枚と 2 倍のフィルタ数に変更したモデルを作成した (以下，提案手法 II)．他にも提案手法 I の変更に加え，図 1 の (e)，(f) のボトルネック部分を通過できる数字の量を 2 倍にしたモデルも作成した (以下，提案手法 III)．これにより，提案手法 I，提案手法 II，提案手法 III と既存の AutoVC(以下，既存手法) の計 4 つのパターンを検証した．

4. 評価実験

4.1 評価方法

本研究では， (1) MOS(Mean Opinion Score) of speech quality, (2) MOS of speaker similarity, (3) RTF(Real Time Factor) の三つの指標を用いて性能を評価する．

(1) は，目標音声と変換した音声を比較して，どの程度音声が劣化しているかを 5(劣化がわからない)〜1(劣化が非常に気になる) の間で何人かに主観的に評価してもらい，その平均点を算出する手法である．

(2) は，目標音声変換した音声を比較して，どの程度声質が似ているかを 5(非常に似ている)〜1(全く似ていない) の間で何人かに主観的に評価してもらい，その平均点を算出する手法である．

(3) は，変換にかかった時間 ÷ 変換する音声の長さ によって算出する．これは CPU 環境と GPU 環境でそれぞれ計測した．

4.2 評価手順

4.2.1 アンケートによる MOS の評価

MOS of speech quality 及び MOS of speaker similarity の計測のため，16 人に対して，192 ページからなるアンケートで調査した．

アンケートは，図 4 に示す Web サイトを作成し，以下のような流れで行なった．

(1) アンケートの趣旨とやり方，取得した情報の取り扱いについて説明する．
(2) アンケートを始める
(3) 変換目標となる人の音声を聞く (図 4 では音声 1 と表記)．
(4) 別の人の声を変換目標の人の声質に変換した音声を聞く (図 4 では音声 2 と表現)．
(5) MOS of speech quality の調査を回答する (図 4 では質問 1 と表現)．
(6) MOS of speaker similarity の調査を回答する (図 4 では質問 2 と表現)．
(7) その他の気づいたことや感想を任意で回答する (図 4

では質問 3 と表現).
(8) 提出ボタンを押して次のアンケートへ進む.
(9) まだアンケートを続ける場合は (3) に戻り，同じ形式の別のアンケート回答する.

調査終了後，通信環境などの影響で送信できていない回答があることが判明したため，アンケートごとに回答数にむらがあり，最も少ないアンケートで 11 回答，最も多い回答で 16 回答となっている.

図 4 Web アンケート画面 (1 ページ目)

4.2.2 速度計測による RTF の調査

以下の環境でプログラムを動作させて速度を計測した.

- OS: Ubuntu 20.04
- GPU: Nvidia GeForce RTX 3080
- CPU: Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz

計測は CPU 環境で実行した場合と GPU 環境で実行した場合の 2 パターンを計測した．取得した情報は，変換に使用した音声の長さ，変換に要した時間全体，変換に要した時間のうち深層学習モデルで変換している時間をそれぞれ計測し，それを元に RTF を算出する．今回計測した RTF は 3 種類ある，(1) 変換全体の RTF，(2) モデルの変換での RTF， (3) モデルの変換以外の RTF である．(1) の場合，実際に使用する際などにどの程度遅延が発生するかの指標として用いる．(2) の場合，深層学習モデルによる遅延がどの程度かを示す指標として用いる．(3) の場合，前処理や波形生成などによる遅延がどの程度かを示す指標として用いる．これらを，CPU 環境と GPU 環境のそれぞれで測定する.

4.3 実験結果

4.3.1 MOS of speech quality

図 5 は，アンケートの質問 1 の回答を手法別に分けた時の MOS of speech quality を示したものである．横軸は変換手法を表す.「提案手法 I」は， 3.5 節で説明した，WORLD を用いた AutoVC である.「提案手法 II」は「提案手法 I」の改良版であり畳み込み層のフィルタ数を既存手法の 2 倍にしたモデルを用いている.「提案手法 III」は「提案手法 I」の改良版であり，ボトルネックを通過する次元数を 2 倍にしたモデルを用いている.「既存手法」は既存の AutoVC を用いており，波形合成は WeveNet で行っている．縦軸は MOS を表している．ここからわかる事として，既存手法に比べどの提案手法でも 2 倍以上の聞き取りやすさを実現していると言える．また，モデルのパラメータによってそれほど値に差異がないため，聴きやすさの面では WORLD を用いたことが有用であったと考えられる.

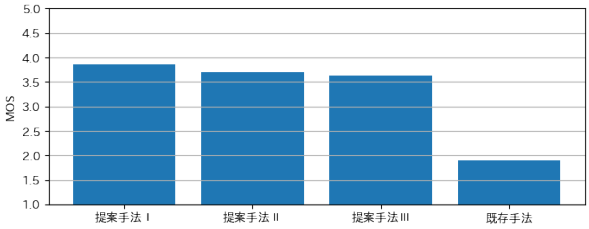


図 5 MOS of speech quality

4.3.2 MOS of speaker similarity

図 6 は，アンケートの質問 2 の回答を手法別に分けた時の MOS of speaker similarity を示したものである．図 5 と同様，横軸は変換手法を示し，縦軸は MOS を示す．ここからわかる事として既存手法に比べ，全体的に同程度の声質の類似度で変換できており，中でもフィルタ数が 2 倍の場合やや，類似度が高いことがわかる．このことから，特徴量をメルスペクトログラムからスペクトル包絡に変更し，F0 を別で合成したとしても，類似度の面において変換精度は劣化していないと言える.

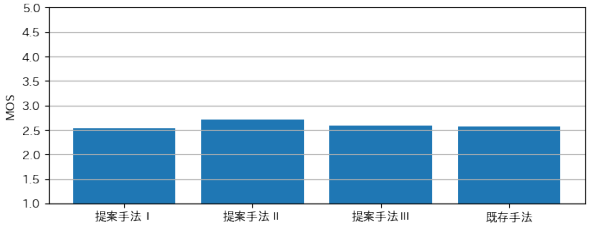


図 6 MOS of speaker similarity

4.3.3 RTF

図 7 は，GPU 環境で測定した RTF の平均である．図 5 と同様，横軸は変換手法を示す. 縦軸は RTF の平均であり，青い部分が深層学習単体の変換時間を用いた RTF，オレンジの部分はその他の部分の変換時間を用いた RTF，縦の長さは長さは総変換時間を用いた RTF になる．ここからわかる事として，全ての既存手法において RIF が 1.0 以下であり，リアルタイム化が達成できている．また，総変換時間のみを見ると，既存手法に対してどの提案手法でも 300 倍以上の速度向上がみられる．ただし，モデルの変換時間のみを見ると変換時間が長くなっており，これは入力する特徴量を変えたことで，モデル全体の処理量が多くなったことが要因である.

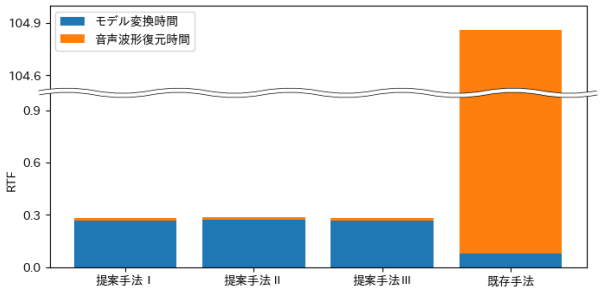


図 7 RTF(GPU)

図 8 は，CPU 環境で測定した RTF の平均である．CPU 環境でも提案手法全てで RIF が 1.0 以下であり，リアルタイム化が達成できている．ただし，GPU での実行に比べ，既存手法 II の RTF が大きくなっている．原因として，畳み込み層のフィルタを増やしたため，並列処理の困難な CPU 環境では長く時間がかかってしまったと考えられる．それに対して，WaveNet の場合，元々自己回帰モデルであるため並列化を行えないので，GPU の恩恵を受けることが難しく，むしろ CPU では高速化している．

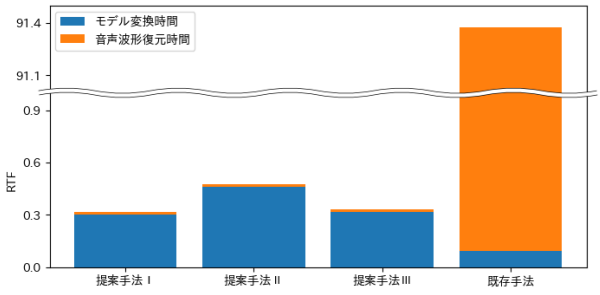


図 8 RTF(CPU)

5. まとめ

本研究では，先行研究の AutoVC というゼロショット声質変換モデルの達成できていなかった，韻律の安定的な変換，リアルタイム化を実現した．提案手法では，WORLD を用いて音声波形を，F0，スペクトル包絡，非周期性指標に分解して，そのうちスペクトル包絡のみを，変換することで，F0 に依存しない声質変換を可能にし，結果的に韻律の安定的な変換を可能にした．また，音声波形に変換する際には，WORLD の Synthesis 関数を用いることで高速な変換が可能にし，リアルタイム化が可能となった．提案手法の有効性を検証するために，アンケート及び実行時間計測を行い MOS of speech quality，MOS of speaker similarity，RTF を測定した．その結果，音声品質は既存手法に比べ高く，話者類似度は同程度，実行速度も速く，リアルタイム化が可能になった．

今後の計画として，より変換時の類似度の向上のため CTCLoss などの損失関数の導入や VAE や Transformer などの層の導入を検討する．また，社会実装に向けて iOS や Android での実行を可能にしたり，使用時の体験向上のための使いやすい UI/UX を検討する．

参考文献

[1] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, Mark Hasegawa-Johnson: ”AUTOVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss”, 入手先 ⟨https://arxiv.org/abs/1905.05879⟩ (2021.11.24).

[2] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu: ”WaveNet: A Generative Model for Raw Audio”, 入手先 ⟨https://arxiv.org/abs/1609.03499⟩ (2021.11.24).

[3] DANIEL W. GRIFFIN, JAE S. LIM: ”Signal estimation from modified short-time Fourier transform”, 入手先 ⟨https://ieeexplore.ieee.org/document/1164317⟩ (2021.11.24).

[4] Ryan Prenger, Rafael Valle, Bryan Catanzaro: ”WaveGlow: A Flow-based Generative Network for Speech Synthesis”, 入手先 ⟨https://arxiv.org/abs/1811.00002⟩ (2021.11.24).

[5] Masanori Morise, Fumiya Yokomori, Kenji Ozawa: ”WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications”, 入手先 ⟨https://doi.org/10.1587/transinf.2015EDP7457⟩ (2021.11.24).

[6] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean: ”Efficient Estimation of Word Representations in Vector Space”, 入手先 ⟨https://arxiv.org/abs/1301.3781⟩ (2021.11.24).

[7] Kaizhi Qian, Zeyu Jin, Mark Hasegawa-Johnson, Gautham J. Mysore: ”F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder”, 入手先 ⟨https://arxiv.org/abs/2004.07370⟩ (2021.11.24).

[8] Kun Liu, Jianping Zhang, Yonghong Yan: ”High Quality Voice Conversion through Phoneme-Based Linear Mapping Functions with STRAIGHT for Mandarin”, 入手先 ⟨https://ieeexplore.ieee.org/document/4406422⟩ (2021.11.24).

[9] Yurii Rebryk, Stanislav Beliaev: ”ConVoice: Real-Time Zero-Shot Voice Style Transfer with Convolutional Network”, 入手先 ⟨https://arxiv.org/abs/2005.07815⟩ (2021.11.24).

[10] Shinnosuke Takamichi, Kentaro Mitsui, Yuki Saito, Tomoki Koriyama, Naoko Tanji, Hiroshi Saruwatari: ”JVS corpus: free Japanese multi-speaker voice corpus”, 入手先 ⟨https://arxiv.org/abs/1908.06248⟩ (2021.11.24).

[11] Takuhiro Kaneko, Hirokazu Kameoka: ”Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks”, 入手先 ⟨https://arxiv.org/abs/1711.11293⟩ (2021.11.24).

[12] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, Nobukatsu Hojo: ”StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks”, 入手先 ⟨https://arxiv.org/abs/1806.02169⟩ (2021.11.24).

[13] r9y9/wavenet_vocoder: WaveNet vocoder (Github), 入手先 ⟨https://github.com/r9y9/wavenet_vocoder⟩ (2021.11.24).

[14] resemble-ai/Resemblyzer: A python package to analyze and compare voices with deep learning (Github), 入手先 ⟨https://github.com/resemble-ai/Resemblyzer⟩ (2021.11.24).