Vol.2025-MUS-142 No.1
2025/3/6

String Mixer：統計的撥弦楽器音合成のための弦単位位置埋め込み

小口 純矢[1,a)]　森勢 将雅[1,b)]

概要：人工的に器楽演奏の音響信号を合成する技術を楽器音合成という．とりわけ楽譜からその演奏楽器音を出力する score-to-audio タスクにおいては，楽器演奏を楽譜によって記述される音声言語と見なすことで音声合成技術を利用した手法が提案されている．Deep Performer はその代表例であるが，polyphonic mixer と呼ばれる処理によって多重音の合成を可能にしつつ，音符の音色変化を線形な変化としてモデル化している．一方で撥弦楽器に合成の対象とする楽器の種類を絞れば，その過渡的な音色変化は物理的に記述することができる．そこで，本研究は polyphonic mixer における線形な位置符号化を指数関数的な減衰モデルによって改良した string mixer を提案し，より実際の撥弦楽器が持つ音色変化に即したモデル化を行う．実験的評価では，一般的な 4 弦のエレクトリックベースギターのデータで訓練されたモデルを小規模な 5 弦のデータでファインチューニングした場合に，string mixer はより高品質な楽器音を合成できることが示唆された．

キーワード：楽器音合成，撥弦楽器，位置符号化，DNN

String Mixer: string-wise positional encoding for statistical plucked string instrument sound synthesis

KOGUCHI Junya[1,a)]　MORISE Masanori[1,b)]

Abstract: The technology of synthesizing musical instrument performance sounds artificially is referred to as musical instrument sound synthesis. In particular, for the score-to-audio task, which generates musical instrument sounds from a given musical score, methods utilizing speech synthesis techniques have been proposed by interpreting musical instrument performance as a speech language described by notation. Deep Performer is a representative example of such an approach. It enables polyphonic sound synthesis through a process called the polyphonic mixer, while modeling the timbral variations of notes as linear transitions. On the other hand, if we limit the target instruments to plucked string instruments, their transient timbral variations can be physically modeled. Therefore, this study proposes the string mixer, which improves the linear positional encoding in the polyphonic mixer by incorporating an exponential decay model, allowing for a more realistic representation of the timbral variations inherent to plucked string instruments. Experimental evaluations suggest that when a model trained on data from a standard four-string electric bass guitar is fine-tuned on a small-scale dataset of a five-string bass, the string mixer achieves higher-quality instrument sound synthesis compared to conventional approaches.

Keywords: Instrument sound synthesis, plucked string instrument, positional encoding, DNN

1. はじめに

楽器音を人工的に生成・変換・加工する技術を楽器音合成 (musical instrument sound synthesis) といい，特に楽

[1] 明治大学
Meiji University
[a)] korguchi@meiji.ac.jp
[b)] mmorise@meiji.ac.jp

譜情報を入力としその演奏楽器音を合成するものを score-to-audio と呼ぶ [1, 2]．Score-to-audio 技術を活用することで，音楽制作における作編曲のプロセスを効率化するだけでなく，身体的・環境的な要因から収録が困難な場合でも豊かな器楽演奏を楽曲へ盛り込むことができる．音楽教育においては，手本となる実施例や演奏を学習者へフィードバックすることで，理論と実践の橋渡しをスムーズに行うことができるため，学習効果の向上が期待できる [3–5]．演奏支援システムにおいて伴奏を自動で生成することで，より本番に近い環境での練習が可能になる [6]．加えて，score-to-audio 技術が新たな音楽性を創造する側面にも注目したい．TR-808 [7] のように，ドラムセットの音を収録・加工し任意のパターンで再生するドラムマシンは，ヒップホップやハウスミュージックといった音楽ジャンルの個性を形成する要素として機能してきた [8]．また，VOCALOID [9] に代表される歌声合成技術によって，ユーザーは人間による歌唱を必要とせずに楽曲制作が可能となり，プロ・アマチュアを問わず制作した楽曲を動画プラットフォーム上などで公開するムーブメントが活発である [10]．これらの例のように，score-to-audio 技術は新しい音楽性を提供する点で重要な役割を果たしている．

現在，製品化されている技術の多くは素片接続 [11] や物理モデリング [12] のように決定論的なアプローチに基づく．一方で，人間による楽器演奏のような複雑なプロセスをこれらの手法によって再現するには一般に困難であり，自然な演奏を模擬するためには膨大かつ精緻な制御パラメータのプログラミングを必要とする [13]．この問題を解決するため，楽譜とその演奏楽器音との関係を深層学習を用いて統計的にモデル化する試みが行われている．これには，楽譜情報から人間の演奏によるゆらぎを付与する手法（パフォーマンス・レンダリング，演奏表情付け）を解く例 [14–17] や，単音の楽器音を合成する手法（audio sample synthesis） [18–20]，指定した音符列に忠実な演奏楽器音を合成する手法（MIDI-to-audio） [21–24] が提案されている．これらは人間らしい演奏と楽器の音響的な振る舞いのモデリングをそれぞれ個別の問題として解いている．

他方，音声を対象として，テキストあるいは歌詞と旋律を入力しそれぞれその発話 [25–30]・歌唱 [31–35] を合成する音声合成技術が目覚ましい発展を遂げている．これらは高い自然性を持つ音声の合成を可能にしつつ，イントネーション・話速・強弱といった要素を制御できる．この成功を受けて，その技術を楽器音合成技術へ応用する手法が登場した．これには，単一のモデルによって楽譜の入力から楽器音波形の合成を一貫して行う end-to-end なフレームワークを利用するもの [36, 37] や，多話者合成モデルを多楽器の合成へ応用したもの [38] などがある．中でも Deep Performer [2] は FastSpeech [39] のアーキテクチャを用いることで，これまで独立して解かれてきた演奏生成と音響合成のモデル化を統合し，高い自然性と制御性の両立を可能としている．Deep Performer は多重音の合成への拡張するために，FastSpeech における length regulator を polyphonic mixer として拡張する．polyphonic mixer は，オンセットと音価に基づいてゼロ埋め・複製された複数の音符の埋め込みベクトルを加算する．このとき，各音符の埋め込みベクトルには，音価に応じた相対的なフレーム位置を，線形に変化する重み付けする音符単位位置符号化（note-wise positional encding; NWPE）が施される．これにより，音の立ち上がりや減衰などの音色変化がモデル化される．

Deep Performer が音符の音色変化が線形であることを仮定する一方で，合成の対象とする楽器の種類をその発音機構に絞れば，その音色変化は物理的に記述することができる．そこで，本研究は撥弦楽器に着目し，polyphonic mixer における線形な位置符号化を指数関数的な弦振動の減衰モデルによって改良した string mixer を提案する．これにより，撥弦時から減衰までの過渡的な実際の撥弦楽器が持つ音色変化に即したモデル化が良い帰納バイアスとして働くことで合成音の品質向上を期待する．実験的評価では，合成音における特徴量の誤差に基づく客観評価および聴取実験に基づく主観評価によって，一般的な 4 弦のエレクトリックベースギターのデータで訓練されたモデルを小規模な 5 弦のデータでファインチューニングした場合に，string mixer はより高品質な楽器音を合成できることを示す．

1.1 ベースライン：Deep Performer

Deep Performer は DNN に基づく score-to-audio フレームワークであり，1) 楽譜情報を入力として実際の演奏に基づくゆらぎを持ったオンセット・音価を出力する演奏モデルと，2) 楽譜情報と演奏モデルのオンセット・音価を入力としてその演奏メルスペクトログラムを出力する音響モデル，3) メルスペクトログラムから楽器音波形を合成する波形生成モデルからなる．

1.1.1 演奏モデル

演奏モデル（**図 1**）における入力は，音符のインデックスを i として，楽譜情報を構成する要素のトークン（本研究では，音高 p_i，オンセット o_i，音価 d_i，ベロシティ v_i，奏法 t_i）である．一つの音符に対して各トークンは同じ次元を持つ複数のベクトルとして並列に線形層を通して埋め込みベクトル $\mathbf{e}$ へ変換される．

$$\mathbf{e}_i^{(p)} = W^{(p)} p_i, \tag{1}$$
$$\mathbf{e}_i^{(o)} = W^{(o)} o_i, \tag{2}$$
$$\mathbf{e}_i^{(d)} = W^{(d)} d_i, \tag{3}$$
$$\mathbf{e}_i^{(v)} = W^{(v)} v_i, \tag{4}$$
$$\mathbf{e}_i^{(t)} = W^{(t)} v_i \tag{5}$$

その後，音符の埋め込みベクトル $\mathbf{e}_i^{(\text{note})}$ としてこれらは加算される．

$$\mathbf{e}_i^{(\text{note})} = \mathbf{e}_i^{(p)} + \mathbf{e}_i^{(o)} + \mathbf{e}_i^{(d)} + \mathbf{e}_i^{(v)}. \tag{6}$$

こうして得られた音符列 $\left\{\mathbf{e}_1^{(\text{note})}, \ldots, \mathbf{e}_N^{(\text{note})}\right\}$ に対して，正弦波位置埋め込みベクトル $\mathbf{e}_i^{(\text{pos})}$ を足し合わせる．

$$\mathbf{z}_i^{(0)} = \mathbf{e}_i^{(\text{note})} + \mathbf{e}_i^{(\text{pos})}, \quad i = 1, 2, \ldots, N \tag{7}$$

上記で得られた $\mathbf{z}_i^{(0)}$ は，エンコーダと呼ばれる複数の feed-forward Transformer ブロック [39] から構成される DNN によって楽譜の特徴を反映した特徴量ベクトルに変換される．具体的にはエンコーダの各層を通して，

$$\mathbf{z}_i^{(\ell)} = \text{Encoder}^{(\ell)}\left(\mathbf{z}_1^{(\ell-1)}, \ldots, \mathbf{z}_N^{(\ell-1)}\right), \quad \ell = 1, 2, \ldots, L_{\text{enc}} \tag{8}$$

と変換され，最終層の出力，

$$\mathbf{z}_i^{(\text{enc})} = \mathbf{z}_i^{(L_{\text{enc}})}, \tag{9}$$

に対し，演奏者による個性とテンポによる演奏スタイルの変化を考慮し，演奏者コード c と BPMb の埋め込みベクトルである $\mathbf{e}^{(c)}$, $\mathbf{e}^{(b)}$

$$\mathbf{e}^{(c)} = W^{(c)} c, \tag{10}$$
$$\mathbf{e}^{(b)} = W^{(b)} b, \tag{11}$$

を加算することで，

$$\mathbf{z}_i^{(\text{enc}+)} = \mathbf{z}_i^{(\text{enc})} + \mathbf{e}^{(c)} + \mathbf{e}^{(b)}. \tag{12}$$

を得る．

最終的に，$\mathbf{z}_i^{(\text{enc}+)}$ に線形層を通し，STFT フレーム上でのオンセット開始フレームインデックス（以下 $\hat{o}_i$ ）と音価に相当するフレーム数（以下 $\hat{d}_i$ ）をに相当する実数を出力する．

$$\left[\hat{o}_i, \hat{d}_i, \hat{s}_i\right]^\top = W^{(\text{out})} \mathbf{z}_i^{(\text{enc}+)} \tag{13}$$

後述する音響モデルへ入力する際には，これらは丸め処理を行ってフレーム単位の自然数へ変換される．

学習時には，正解のオンセット $\tilde{o}_i$ と音価 $\tilde{d}_i$ および弦番号に対し，以下によって定義される平均二乗誤差（MSE）を用いる．

$$\mathcal{L}_{\text{Perform}} = \frac{1}{N} \sum_{i=1}^{N} \left[(\hat{o}_i - \tilde{o}_i)^2 + \left(\hat{d}_i - \tilde{d}_i\right)^2 + (\hat{s}_i - \tilde{s}_i)^2 \right]. \tag{14}$$

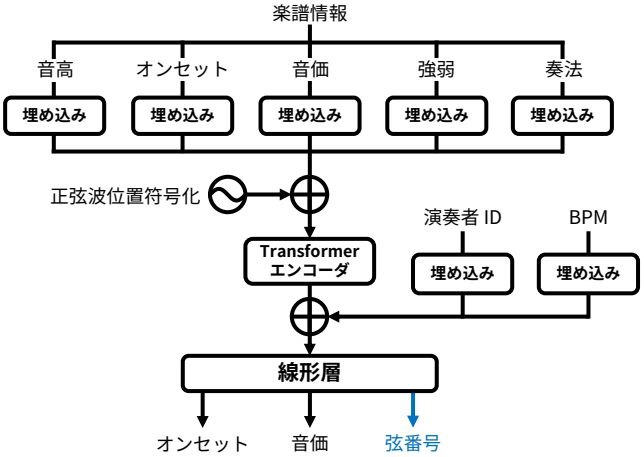

図 1 本研究において改良した Deep Performer 演奏モデル．提案手法ではオンセットと音価に加えて弦番号を推定する．

1.1.2 音響モデル

音響モデル（**図 2**）は，演奏モデルから出力されたオンセットと音価，および楽譜情報に基づいて，メルスペクトログラムを推論する．演奏モデルと同様に，まず各音符の楽譜情報のトークンをエンコードし，演奏者識別（ID）とテンポの埋め込みベクトルを加えて音符埋め込みベクトルを得る．音符埋め込みベクトルは次に，polyphonic mixer と呼ばれる音価とオンセットによってベクトルをゼロ埋め・複製する機構を経て Transformer デコーダに渡され，メルスペクトログラムを出力する．Polyphonic mixer は，FastSpeech [39] における length regulator を多重音に拡張した手法である．Length regulator は，演奏モデルに基づいて各音符が持つ音価 $\hat{d}$ に対応するフレーム数だけ $\mathbf{z}_i^{(\text{note})}$ を複製する．Polyphonic mixer はこれに加え，以下のように演奏モデルからの出力であるオンセットに従ってゼロベクトルで埋めながらシフトさせ，全ての音符を並列に加算することで多重音を表現する．

$$\mathbf{Z}_i^{(\text{shift})}[n] = \begin{cases} \frac{n-\hat{o}_i}{\hat{d}_i} \mathbf{w} \odot \mathbf{z}_i^{(\text{note})}, & \hat{o}_i \le n < \hat{o}_i + \hat{d}_i, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{15}$$

$$\mathbf{Z}^{(\text{mix})}[n] = \sum_{i=1}^{N} \mathbf{Z}_i^{(\text{shift})}[n]. \tag{16}$$

ここで n はフレームのインデックスを表す．また，エンコードされた音符埋め込みベクトルをフレーム数だけ複製する際には，音符単位位置埋め込み（Note-wise positional encoding; NWPE）が適用される．この NWPE は以下のように適用される：

ここで $\odot$ はアダマール積を表し，$\mathbf{w}$ は学習可能なベクトルであり，$\mathbf{z}_{\text{frame}} \approx \mathbf{z}_{\text{note}}$ となるように小さな乱数で初期化される．音符埋め込みベクトル $\mathbf{z}_{\text{note}}$ に対して，a をその音符の音価の長さを 1 とした場合の内の相対位置として $[0, 1]$ の範囲で与える．これにより NWPE は時間的な遷移にともなって異なる重み付けがなされ，音色の変化を識別

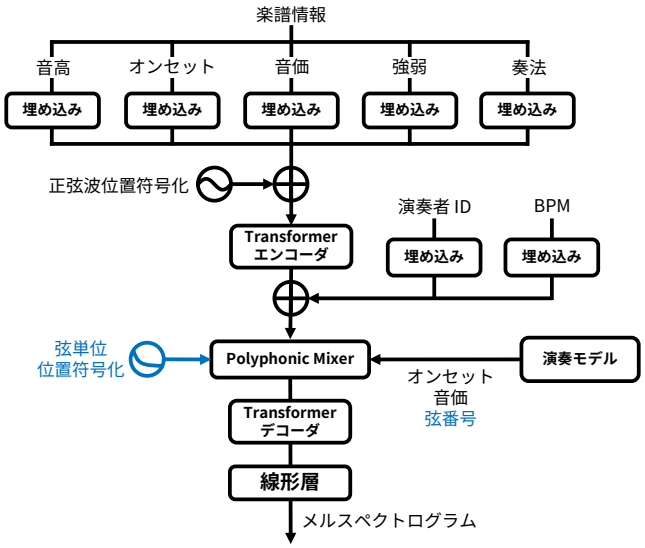

図 2 Deep Performer における音響モデル．提案手法では音符単位位置符号化の代わりに弦単位位置符号化を用いる．

的に学習するように誘導する．音響モデルは音符間だけでなく音符内の時間的な音色変化をモデル化できるようになり，自然性の向上が期待される．

最後に，エンコーダ同様，多層の Transformer によって構成されたデコーダと線形変換 $W^{(\mathrm{mel})}$ を経てフレーム単位のベクトル系列はメルスペクトログラム m に変換される．

$$\mathbf{z}_i^{(\ell)} = \mathrm{Decoder}^{(\ell)}\left(\mathbf{Z}^{(\mathrm{mix})}[n]\right), \quad \ell = 1, 2, \ldots, L_{\mathrm{dec}}, \tag{17}$$

$$\hat{\mathbf{m}}_n = W^{(\mathrm{mel})}\mathbf{z}_f^{(\mathrm{dec}, L_{\mathrm{dec}})} \tag{18}$$

学習のための損失関数は，メルスペクトログラムの平均二乗誤差である．

$$\mathcal{L}_{\mathrm{Acoustic}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{\mathbf{m}} - \tilde{\mathbf{m}})^2 . \tag{19}$$

1.2 波形生成モデル

デコーダから出力されたメルスペクトログラムを波形生成モデルによって楽器音波形に変換する．このような DNN はニューラルボコーダと呼ばれ，合成の対象とする音データが十分に得られない場合にはあらかじめ他のデータで訓練された事前学習モデルの重みを初期値としてから学習を行う（ファインチューニング）ことが多い．

一方で，エレキベース音は音声やその他の楽器とは異なる低音部を担当する楽器であり，様々な音色的奏法を使い分けるため，その合成には多様な音色に対して安定した品質が要請される．そのため，事前学習モデルの重みを利用するにしても，訓練音声データとは分布の異なる楽器音に対して音質劣化のない頑健なモデルを利用する必要がある．そこで，本研究においては BigVGAN [40] を利用する．BigVGAN は敵対的生成ネットワーク（generative adversarial network; GAN） [41] に基づくニューラルボコーダであり，メルスペクトログラムによって条件付けされ波形を生成する．その他多くのボコーダの合成品質は，学習データに大きく依存することが知られており，データセットに含まれない話者や録音環境のメルスペクトログラムを用いた合成品質が大幅に低下する問題があった．BigVGAN は，オーディオ品質を向上させるために，生成モデルに周期的な活性化関数とアンチエイリアス表現を導入することでこの問題に対処している．さらに，BigVGAN は多くのニューラルボコーダよりもモデルパラメータの数を増加させている．これにより，BigVGAN は学習データに含まれない話者，言語，録音環境，歌声，音楽，楽器など，さまざまな分布外のデータに対しても高品質な波形を出力できる．

事前学習済みのモデルをエレキベースデータベースでファインチューニングし，合成音の品質を向上させる．これにより，さまざまな音高や奏法のバリエーションを持つエレキベース音に対する頑健性を期待する．

2. 提案手法：弦単位位置符号化（string mixer）

撥弦楽器は弦が指ないしプラスチックの小片（ピック）などで弾かれることで撥音される．そこで，polyphonic mixer における線形な位置符号化を以下のように弦によって異なる指数関数によって減衰する符号化に置き換える．

$$\mathbf{Z}_i^{(\mathrm{shift})}[n] = \begin{cases} \frac{n-\hat{o}_i}{\hat{d}_i} e^{-\boldsymbol{\alpha}\frac{\hat{s}_i}{N_s}n+\boldsymbol{\beta}} \odot \mathbf{z}_i^{(\mathrm{note})}, & \hat{o}_i \le n < \hat{o}_i + \hat{d}_i, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{20}$$

N_s は弦の本数を表し，$\boldsymbol{\alpha}$，$\boldsymbol{\beta}$ はそれぞれ学習可能なパラメータである．弦が太いほど $\frac{\hat{s}_i}{N_s}$ は大きくなるため，強く重み付けされることで減衰が緩やかになる．これにより，実際の弦振動に即した位置符号化が音符ベクトルになされ，音色変化をより忠実にモデル化することが可能になる．

3. 実験的評価

本研究の有効性を検証するため，合成音における特徴量の誤差に基づく客観評価および聴取実験に基づく主観評価実験を行った．

4. データセット

Deep Performer および提案手法による改良モデルの学習に使用する撥弦楽器としてエレクトリックベースギター（以降弦の数によって 4 弦ベース，5 弦ベースのように呼称する）を用いる．学習に用いるデータセットには 4 弦ベースのフレーズを収録したもの [42] を用いるほか，弦の種類による合成音の差異を適切に評価するために，5 弦の音の

データセットを新たに構築した．それぞれ 60 から 120 beat per minute; BPM で 4〜8 小節程度のフレーズを 60 フレーズ（約 22 分）収録した．演奏は 10 年以上のアマチュア奏者が行い，オーディオインターフェースは RME ADI-2 Pro FS R [43] ，楽器は Mayones Jabba Custom EP5 [44] を使用した．48 kHz サンプリング，24 bit 量子化の wav ファイルとして収録したものを，実験には 24 kHz にダウンサンプリングして用いた．楽譜上のオンセット・音価は動的時間伸縮と混合ガウスモデルに基づく音色変換の反復アルゴリズム [42] によって与えられ，弦番号は演奏者がフレーズ作成時に手動で与えた．

4.1 実験条件

比較に用いる楽器音は, 収録音（ground truth）, BigVGAN による分析合成音（BigVGAN），従来の Deep Performer（polyphonic mixer），そして提案手法（string mixer）による改良モデルによる合成音である．

モデルの学習にはデータセットを訓練用・検証用・評価用にそれぞれ 80 %，10 %，10 %の比率でランダムに分けて用いた．音響モデル及び演奏モデルの諸条件を**表 1**, **表 2**, **表 3** に示す．

表 1 演奏モデルにおけるネットワーク構造の設定.

エンコーダの層数	3
マルチヘッド注意のヘッド数	2
マルチヘッド注意の中間ノード数	64
CNN の中間ノード数	256
CNN のフィルタサイズ	9, 1
最大系列長	1,000
最大音価 [s]	96

表 2 音響モデルにおけるネットワーク構造の設定.

エンコーダの層数	3
デコーダの層数	6
マルチヘッド注意のヘッド数	2
マルチヘッド注意の中間ノード数	128
DNN の中間ノード数	512
CNN のフィルタサイズ	9, 1
最大系列長	1000
最大音価 [s]	96
メルスペクトログラムのビン数	100

メルスペクトログラムからの波形生成には BigVGAN を用いた．音声データを用いた学習済みのモデル（bigvgan_base_24khz_100band）を著者の GitHub リポジトリ [45] から取得し, 新たにエレキベース音を用いてファインチューニングを行った．諸条件を**表 4** および**表 5** に示す．

表 3 演奏モデルと音響モデルのハイパーパラメータ.

バッチサイズ	16
ドロップアウト率	0.2
AdamW における β_1	0.9
AdamW における β_2	0.98
AdamW における ϵ	10^{-9}
学習率アニーリングステップ数 (演奏モデル)	1,000
学習率アニーリング率 (演奏モデル)	0.5
勾配クリッピングの閾値	1.0
Warm up のステップ数 (演奏モデル)	1,000
Warm up のステップ数 (音響モデル)	4,000
学習のステップ数 (演奏モデル)	10,000
学習のステップ数 (音響モデル)	100,000

表 4 BigVGAN の設定.

サンプリング周波数 [Hz]	24,000
メルスペクトログラムのビン数	100
ホップサイズ	256
窓長	1024
セグメントサイズ	8,192

表 5 BigVGAN のハイパーパラメータ.

AdamW における β_1	0.9
AdamW における β_2	0.98
AdamW における ϵ	10^{-8}
荷重減衰率	0.01
バッチサイズ	32
学習ステップ数	100,000

4.2 結果と考察

実験結果を**表 6** に示す．4 弦ベースの合成音について，メルスペクトログラムの誤差は提案手法が優れるものの，主観評価において顕著な差は見られなかった．提案手法は合成の精度を高めうるが，知覚できるほどの品質の差異をもたらさなかった可能性がある．他方，よりデータの少ない 5 弦ベースの合成音については，客観評価だけでなくその音色の MOS が有意に高い結果となった．良い帰納バイアスとして働くことで少量のデータによるファインチューニングであっても高い精度で推論を行うことができたものと考えられる．

5. おわりに

本研究は string mixer を提案し，従来の線形な音色変化を仮定した音符の位置符号化に対して，撥弦楽器音の音色変化をその減衰特性に沿った符号化を行うことによって，合成音の品質向上を達成した．今後の予定としては，ダイナミクス変化のみならず時間に伴って高域が失われるといった周波数特性の変化の考慮すや，他の楽器の撥音機構に基づいた位置符号化の開発などが挙げられる．

謝辞 本研究は, JSPS 科研費 JP22J22158 の支援によって行われた．

表 6 主観評価の結果.

	4 strings				5 strings (finetune)			
	Pitch accuracy	Timbre	Noise level	Overall	Pitch accuracy	Timbre	Noise level	Overall
Ground truth	3.98	4.45	3.51	4.27	3.82	4.51	3.47	4.19
BigVGAN	3.13	3.24	2.52	3.01	2.98	2.75	1.57	2.90
Polyphonic mixer	2.43	1.94	1.33	2.01	1.26	2.11	1.27	1.79
String mixer (proposed)	2.54	1.89	1.33	2.16	1.93	2.57	1.45	2.31

参考文献

[1] Wang, B. and Yang, Y.-H.: PerformanceNet: Score-to-Audio Music Generation with Multi-Band Convolutional Residual Network, *Proceedings of American Association for Artificial Intelligence Symposium Series (AAAI 2019)*, Vol. 33, No. 01, pp. 1174–1181 (online), DOI: 10.1609/aaai.v33i01.33011174 (2019).

[2] Dong, H.-W., Zhou, C., Berg-Kirkpatrick, T. and McAuley, J.: Deep Performer: Score-to-Audio Music Performance Synthesis, *Proceedings of 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2022)*, pp. 951–955 (online), DOI: 10.1109/ICASSP43922.2022.9747217 (2022).

[3] Haldren, J.: Toward a more comprehensive understanding of sound design in music technology education, *Journal of Technology in Music Learning*, Vol. 2, No. 1, pp. 25–35 (2006).

[4] Webster, P. R.: Creative thinking in music: Advancing a model, *Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009)*, pp. 279–284 (2009).

[5] Standage, D.: A survey of the use of musical instrument digital interface (MIDI) in the secondary music classroom, *Music Education Research*, Vol. 14, No. 4, pp. 409–425 (2012).

[6] Sagayama, S., Nakamura, T., Nakamura, E., Saito, Y., Kameoka, H. and Ono, N.: Automatic music accompaniment allowing errors and arbitrary repeats and jumps, *Proceedings of 167th Meeting of the Acoustical Society of America (ASA 2014)*, Vol. 21, No. 1, Acoustical Society of America, p. 035003 (online), DOI: 10.1121/1.4904932 (2014).

[7] Roland: Roland TR-808, `https://www.roland.com/jp/products/rc_tr-808/`. (Accessed on 20 Nov. 2024).

[8] Roland: Four Decades, One Sound TR-808 誕生から40周年, `https://www.roland.com/jp/promos/roland_tr-808/`. (Accessed on 20 Nov. 2024).

[9] Kenmochi, H. and Ohshita, H.: VOCALOID-commercial singing synthesizer based on sample concatenation, *Proceedings of the 8th Conference of the International Speech Communication Association (INTERSPEECH 2007)*, pp. 4009–4010 (2007).

[10] Kenmochi, H.: VOCALOID and Hatsune Miku phenomenon in Japan, *Proceedings of First Interdisciplinary Workshop on Singing Voice (InterSinging 2010)*, pp. 1–4 (2010).

[11] Schwarz, D.: Concatenative sound synthesis: The early years, *Journal of New Music Research*, Vol. 35, No. 1, pp. 3–22 (2006).

[12] Smith, J. O.: Physical modeling using digital waveguides, *Computer music journal*, Vol. 16, No. 4, pp. 74–91 (1992).

[13] Roads, C.: *Computer Music Tutorial*, chapter 2, MIT Press (1996).

[14] Oore, S., Simon, I., Dieleman, S., Eck, D. and Simonyan, K.: This time with feeling: Learning expressive musical performance, *Neural Computing and Applications*, Vol. 32, No. 4, pp. 955–967 (2020).

[15] Maezawa, A., Yamamoto, K. and Fujishima, T.: Rendering Music Performance With Interpretation Variations Using Conditional Variational RNN, *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR 2018)*, pp. 855–561 (2018).

[16] Jeong, D., Kwon, T., Kim, Y. and Nam, J.: Graph neural network for music score data and modeling expressive piano performance, *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, PMLR, pp. 3060–3070 (2019).

[17] Jeong, D., Kwon, T., Kim, Y., Lee, K. and Nam, J.: VirtuosoNet: A Hierarchical RNN-based System for Modeling Expressive Piano Performance, *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR 2019)*, pp. 908–915 (2019).

[18] Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D. and Simonyan, K.: Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders, *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, pp. 1068–1077 (2017).

[19] Défossez, A., Zeghidour, N., Usunier, N., Bottou, L. and Bach, F.: SING: Symbol-to-Instrument Neural Generator, *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, pp. 9041–9051 (2018).

[20] Engel, J., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C. and Roberts, A.: GANSynth: Adversarial Neural Audio Synthesis, *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*, pp. 1–17 (2019).

[21] Manzelli, D., McVicar, M., Kumar, K. and Smaragdis, P.: Conditioning Deep Generative Raw Audio Models for Structured Automatic Music, *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, pp. 2467–2476 (2018).

[22] Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J. and Eck, D.: Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset, *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*, pp. 1–12 (2019).

[23] Engel, J., Hantrakul, L., Gu, C. and Roberts, A.: DDSP: Differentiable Digital Signal Processing, *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*, pp. 1–19 (2020).

[24] Hayes, A., Engel, J., Roberts, A., Hantrakul, L., Gu, C. and Roberts, A.: Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders, *Proceedings of the 9th International Conference on Learning Representa-*

tions (ICLR 2021), pp. 1068–1077 (2021).

[25] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. and Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio, *Proceedings of the 9th ISCA Speech Synthesis Workshop (SSW 9)*, pp. 125–125 (2016).

[26] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., Saurous, R. A., Agiomyrgiannakis, Y. and Wu, Y.: Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions, *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, pp. 4779–4783 (2018).

[27] Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z. and Liu, T.-Y.: FastSpeech 2: Fast and High-Quality End-to-End Text to Speech, *Proceedings of the 9th International Conference on Learning Representations (ICLR 2021)* (2021).

[28] Kim, J., Kong, J. and Son, J.: Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech, *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 5530–5540 (2021).

[29] Casanova, E., Shulby, C., Silva, J. F., Junior, R., Ponti, M. A., Gama, J. P., Shillingford, B., Coile, A., Prenger, R., Catanzaro, B. and Valle, R.: YourTTS: Towards Zero-Shot Multi-Speaker TTS and Zero-Shot Voice Conversion for Everyone, *Proceedings of the 38th International Conference on Machine Learning (ICML 2022)*, pp. 255–270 (2022).

[30] Wang, C., Chen, S., Wu, Y., Zhang, Z., Chen, L. Z., Liu, S., Wang, Y., Li, Z., Liu, J., Wu, H., Li, P., Wei, F. and Zhou, M.: Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers, *arXiv preprint arXiv:2301.02111* (2023).

[31] Blaauw, M. and Bonada, J.: A Neural Parametric Singing Synthesizer Modeling Timbre and Expression from Natural Songs, *Applied Sciences*, Vol. 7, p. 1313 (2017).

[32] Chen, J., Tan, X., Luan, J., Qin, T. and Liu, T.-Y.: Hifisinger: Towards high-fidelity neural singing voice synthesis, *arXiv preprint arXiv:2009.01776* (2020).

[33] Ren, Y., Tan, X., Qin, T., Luan, J., Zhao, Z. and Liu, T.-Y.: DeepSinger: Singing Voice Synthesis with Data Mined From the Web, *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Association for Computing Machinery, pp. 1979—-1989 (online), available from ⟨https://doi.org/10.1145/3394486.3403249⟩ (2020).

[34] Hono, Y., Hashimoto, K., Oura, K., Nankaku, Y. and Tokuda, K.: Sinsy: A Deep Neural Network-Based Singing Voice Synthesis System, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 29, pp. 2803–2815 (2021).

[35] Liu, J., Li, C., Ren, Y., Chen, F. and Zhao, Z.: Diffsinger: Singing voice synthesis via shallow diffusion mechanism, *Proceedings of the 36th AAAI conference on artificial intelligence (AAAI)*, Vol. 36, No. 10, pp. 11020–11028 (2022).

[36] Cooper, E., Wang, X. and Yamagishi, J.: Text-to-Speech Synthesis Techniques for MIDI-to-Audio Synthesis, *Proceedings of 11th ISCA Speech Synthesis Workshop (SSW 11)*, pp. 130–135 (online), DOI: 10.21437/SSW.2021-23 (2021).

[37] Shi, X., Cooper, E., Wang, X., Yamagishi, J. and Narayanan, S.: Can Knowledge of End-to-End Text-to-Speech Models Improve Neural Midi-to-Audio Synthesis Systems?, *Proceedings of 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023)*, pp. 1–5 (2023).

[38] 中迫酒菜：Melisma: 楽譜ラベルに基づく単一モデルによる DNN 多楽器・多歌唱者音合成システム, 情報処理学会研究報告音楽情報科学研究会（MUS），Vol. 2024-MUS-141, No. 2, pp. 1–6 (2024).

[39] Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z. and Liu, T.-Y.: FastSpeech: Fast, robust and controllable text to speech, *Proceedings of Advances in Neural Information Processing Systems (NeurIPS 2019)*, Vol. 32 (2019).

[40] Lee, S., Ping, W., Ginsburg, B., Catanzaro, B. and Yoon, S.: BigVGAN: A Universal Neural Vocoder with Large-Scale Training, *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, (online), available from ⟨https://openreview.net/forum?id=iTtGCMDEzS⟩ (2023).

[41] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative Adversarial Nets, *Proceedings of the 28th Conference on Neural Information Processing Systems (NeurIPS 2014)* (Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. and Weinberger, K., eds.), Vol. 27, Curran Associates, Inc., pp. 1–9 (2014).

[42] Koguchi, J. and Morise, M.: Phoneme-inspired playing technique representation and its alignment method for electric bass database, *Proceedings of 16th International Symposium on Computer Music Multidisciplinary Research (CMMR 2023)*, pp. 170–177 (2023).

[43] RME: ADI-2 Pro FS R, `https://www.rme-audio.de/adi-2-pro-fs-be.html`. (Accessed on 11 Feb. 2025).

[44] Mayones: Jabba Custom EP 5, `https://mayones.com/page/jabba-custom-ep-5-2019/`. (Accessed on 11 Feb. 2025).

[45] NVIDIA: BigVGAN [Source code], `https://github.com/NVIDIA/BigVGAN`. (Accessed on 31 July 2023).