Vol.2021-MUS-132 No.22
2021/9/17

楽譜情報を援用した音楽音響信号に対する混合Differentiable DSPモデルの合成パラメータ推定

川村 真也[1]　中村 友彦[1]　北村 大地[2]　猿渡 洋[1]　高橋 祐[3]　近藤 多伸[3]

概要：Differentiable Digital Signal Processing（DDSP）は深層ニューラルネットワーク (deep neural network: DNN) を用いた楽器音生成モデルであり，中間表現として得られる基本周波数，ラウドネス，音色特徴量 (合成パラメータ) を変更することで，楽器音を柔軟に加工できる．しかし，DDSP は単旋律の単一楽器音を対象として設計されており，複数音源が混在した音響信号に対しては直接適用できない．また，事前に既存の音源分離手法を用いて各楽器音に分離すれば DDSP を適用できるものの，分離に伴う歪みや当該楽器以外の残留音により必ずしも DDSP が適切に動作するとは限らない．そこで，本稿では音源分離を介さずに混合音から直接各音源の合成パラメータを推定するため，事前学習済みの DDSP を複数用いた混合楽器音生成モデル (混合 DDSP モデル) を提案する．楽譜情報を援用しながら提案する混合 DDSP モデルからの生成音を混合楽器音にフィッティン グすることで，各楽器音に対応する合成パラメータを推定する．混合楽器音からの合成パラメータ推定実験により，提案法の有効性を示す．

1. はじめに

音楽は様々な楽器音によって構成されており，各楽器をユーザの意図通りに加工できる楽音加工システムの実現は，音楽情報処理における重要な問題の 1 つである．楽音加工システムは，対象とする楽器や目的によって様々である．例えば，ドラム音に特化した手法としては，音色置換システム [1,2] やドラムパターンの置換システム [3] が提案されている．一方，楽曲中の各楽器音を音量を個別に変更できるシステム [4,5] や，楽器音の音色やフレーズに対する置換システム [6] など，調波楽器音を主な対象とするシステムも提案されている．本稿では，特に複数の調波楽器で演奏された混合楽器音を扱う．

深層ニューラルネットワーク (deep neural network: DNN) を用いた音の合成はこれまで盛んに研究され，一例として直接波形を形成するモデル [7–9] その中でも，DNN に加算・減算合成シンセサイザの構造を取り入れた楽器音生成モデルとして differentiable digital signal processing（DDSP）が注目を集めている [10]．DDSP は，単旋律の単一楽器音から時刻毎の音色を表す特徴量（音色特徴量）を抽出するエンコーダと，音色特徴量と基本周波数（F_0），ラウドネスから当該楽器音の時間信号を出力するデコーダからなる．本稿では，デコーダの入力であるこれら 3 つの特徴量を合成パラメータと呼ぶ．デコーダに入力される合成パラメータがユーザに解釈しやすい形で提供されるため，人手で合成パラメータを適切に変更することで，音色や音高，音量を柔軟に加工できる．また，他の楽器音生成のための DNN に比べ少量のデータで学習できることが経験的に確認されており，比較的学習コストも低い．

DDSP は入力音響信号が単旋律かつ単一の調波楽器音であることを前提に設計されている．しかし，加工対象となる楽器音は複数の楽器音の中に含まれることも多く，より可用性の高い楽音加工を実現するためには，混合楽器音に適用できる手法が必要である．

単純には，既存の音源分離手法を用いて対象とする楽器音を分離し，分離音に DDSP を適用すれば良い．しかし，DNN の導入により音源分離性能は向上しつつあるものの [11]，音源分離は未だ困難な問題である．そのため，分離音には分離に伴う歪みや他楽器の残留音が含まれる場合が多く，分離音に対して必ずしも DDSP が適切に動作するとは限らない．実際に，4 節の実験で述べるように，既存の楽譜情報を援用した音源分離手法を用いた場合，分離音に DDSP を適用して得られた音響信号の音色は対象楽器音とは大きく異なっていた．また，対象楽器を分離できる DNN 音源分離手法を用意するためには，学習データと

[1] 東京大学
The University of Tokyo, Tokyo, Japan
[2] 香川高等専門学校
National Institute of Technology, Kagawa College, Kagawa, Japan
[3] ヤマハ株式会社
Yamaha Corporation, Shizuoka, Japan

して十分な量の対象楽器音データや混合されうる他の楽器音が必要であり，学習コストは高い．

本稿では，音源分離を介さず，混合楽器音から直接楽器音の合成パラメータを推定する方法を提案する．提案法では各楽器の音響信号が DDSP から生成されているとみなし，各 DDSP の出力の和として混合楽器音を表現する．このモデルを混合 DDSP モデルと呼ぶ．混合 DDSP モデルの出力を観測混合楽器音にフィッティングすることで，各楽器の合成パラメータを推定する．

提案法で推定する合成パラメータは，音色や音高，音量に対応する人間が解釈可能な形式で与えられる．そのため，楽譜情報などの音楽的情報が事前に得られれば，それらの情報を推定に反映できる．近年では，DNN を用いた高精度な自動採譜手法が提案されており [12]，採譜結果は有用な事前情報として利用できる．楽譜情報を援用することで音源分離性能が向上することは様々な研究で報告されており [13–16]，合成パラメータ推定においても有用な事前情報となりうる．そこで，楽譜情報を合成パラメータの初期値として反映させる方法を検討する．

最後に，2〜4 種類の楽器の混合楽器音と同一 3 楽器による混合楽器音からの合成パラメータ推定実験により，楽譜情報を援用した既存の音源分離手法を適用した後に DDSP を適用する手法に比べ，提案法が高精度に合成パラメータを推定することができることを示す．

2. 関連研究

2.1 DDSP

DDSP は，楽器音の音響信号を入力として，合成パラメータとして F_0，音色特徴量，ラウドネスを得ながら入力信号を再構成する DNN ベース楽器音生成モデルである（図 1 参照）．時間長 N の入力信号 $\boldsymbol{x} \in \mathbb{R}^N$ が DDSP に入力されると，入力信号から合成パラメータが計算される．当該信号時間長に対応するフレーム数を T，フレームインデックスを $t = 1, \ldots, T$ とする．また，フレーム t の F_0 を $f_t \in \mathbb{R}_{\geq 0}$，音色特徴量を $\boldsymbol{z}_t \in \mathbb{R}^D$，ラウドネスを $l_t \in \mathbb{R}$ で表す．F_0 は，事前に学習された CREPE [17] と呼ばれる F_0 推定用 DNN により算出する．音色特徴量は，入力信号からメル周波数ケプストラム係数（mel-frequency cepstral coefficient: MFCC）を算出し，それを入力とする DNN により $\boldsymbol{z}_t$ を得る．この音色特徴量を抽出する DNN をエンコーダと呼ぶ．ラウドネスは，入力信号のパワースペクトルに対し対数をとり，A 特性による重み付けを適用することで dB 単位で得る．

次に，DDSP は得られた $f_t, \boldsymbol{z}_t, l_t$ を DNN で構成されたデコーダに入力し，加算合成シンセサイザと減算合成シンセサイザによりそれぞれ調波成分と非調波成分を作成する．ここで，n は信号領域の時刻インデックスとする．加算合成シンセサイザは，K 倍音までの調波成分からなる信号 $\hat{\boldsymbol{h}} = [\hat{h}_1, \ldots, \hat{h}_N]^\top \in \mathbb{R}^N$ を出力する．調波成分 $\hat{h}_n$ は，f_t を参照しつつデコーダ出力である時刻 n での振幅 $a_n \in \mathbb{R}_{\geq 0}$ と各調波成分の相対振幅 $c_{k,n} \in \mathbb{R}_{\geq 0}$ を用いて以下のよう表せる．

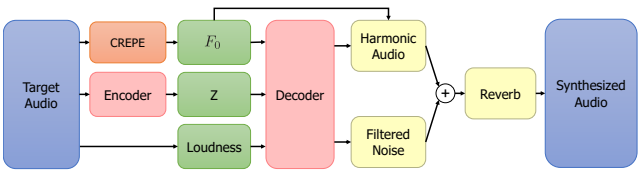

図 1: DDSP の構造

$$\hat{h}_n = a_n \sum_{k=1}^{K} c_{k,n} \sin\left(2\pi \sum_{m=1}^{n} k\tilde{f}_m\right) \tag{1}$$

ここで，$k = 1, \cdots, K$ は調波インデックスであり，$\tilde{f}_n$ は f_t を時間信号の解像度まで線形補完した F_0 を表す．また，$c_{k,n}$ は $\sum_k c_{k,n} = 1$ を満たす．

減算合成シンセサイザでは，周波数サンプリング法と窓関数法により設計したフレーム毎の有限長のインパルス応答（finite impulse response: FIR）をもつフィルタを用いてノイズをフィルタリングする．ノイズは各時刻で $[-1, 1]$ の範囲から一様ランダムにサンプルしたものを用いる．デコーダから出力されるフレーム毎の FIR フィルタの周波数応答を用いて，対応するフレームのノイズをフィルタリングし Hann 窓を乗算する．各フレームで得られたその結果を重畳加算することで，非調波成分を生成する．得られた調波成分と非調波成分を足し合わせ，それをインパルス応答長に対応するカーネルサイズをもつ畳み込み層により実装したリバーブを通した後で，合成音 $\hat{\boldsymbol{x}} \in \mathbb{R}^N$ が得られる．

DDSP のエンコーダとデコーダを構成する DNN のパラメータは，入力された楽器音を再構成できるように学習される．入力楽器音 $\boldsymbol{x}$ と合成音 $\hat{\boldsymbol{x}}$ の再構成誤差は，マルチスケールスペクトルロスにより計算する [10]．マルチスケールスケールスペクトルロスは，2 つの音響信号の誤差を複数の異なる時間周波数解像度の短時間 Fourier 変換（short-time Fourier transform: STFT）での誤差の和として算出する．用いる時間周波数解像度の種類を I とし，i 番目の時間周波数解像度に対応するフレーム長とフレームシフトで STFT を行う演算を $\mathcal{F}_i$ で表す．このとき，$\boldsymbol{x}$ と $\hat{\boldsymbol{x}}$ のマルチスケールスケールスペクトルロスは以下のように定義される．

$$L(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sum_{i=1}^{I} L_i(\boldsymbol{x}, \hat{\boldsymbol{x}}) \tag{2}$$

$$L_i(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \|\mathcal{F}_i\boldsymbol{x} - \mathcal{F}_i\hat{\boldsymbol{x}}\|_1 + \|\log(\mathcal{F}_i\boldsymbol{x}) - \log(\mathcal{F}_i\hat{\boldsymbol{x}})\|_1 \tag{3}$$

2.2 音源分離

混合楽器音に対する DDSP の前処理として，音源分離

手法を用いることができる．混合音から直接分離を行うこともできるが，より高精度な分離結果を得るため楽譜情報を援用する分離手法も提案されている [13–16]．近年は，DNN を用いる方法もあるものの [13–15]，十分な量の対象楽器音の学習データや混合されうる他の楽器音が必要であり学習コストが高い．それに対し，非負値行列因子分解（nonnegative matrix factorization: NMF）に基づく分離手法は比較的少ない学習データでも動作する [16]．当該手法では，事前に音楽データから NMF の基底を学習しておき，楽譜と実演奏のアライメントを行いつつ分離を行う．

また，提案法のように，DNN ベースの楽音生成モデルを混合したものを，混合音にフィッティングする音源分離手法も提案されている [18]．当該手法では，敵対的学習を用いて各楽器音の学習データから楽器音生成用 DNN を事前に学習する．その後，得られた各楽器音の DNN の出力の和を各 DNN の入力である潜在表現を最適化することで観測信号にフィッティングし，各楽器の推定値を得る．しかし，潜在表現の各次元が音楽的要素に対応している保証はなく，楽器加工にそのまま転用することは難しい．また，楽譜情報を潜在表現に反映させることも単純には難しい．

3. 提案手法

3.1 動機と方策

混合楽器音に含まれる楽器音を DDSP により加工する方法の 1 つは，事前に対象楽器音を音源分離手法により分離し，分離音に対して DDSP を適用する方法である．DDSP はクリーンな楽器音を用いて学習されるため，分離音に含まれる歪みや他楽器の残留音が加工性能に直結し，高精度な音源分離が必要となる．DNN の導入により音源分離手法の性能は向上しているものの [11]，歪みや他の楽器音が含まれないクリーンな楽器音は必ずしも得られない．そのため，DDSP が適切に動作する合成パラメータを分離音から得られるとは限らない．実際，4 節の実験で示すように，楽譜情報を援用した音源分離手法を用いた場合であっても，分離音から得られた F_0 とラウドネスはクリーンな楽器音から得られたものとは大きく乖離していた．より高精度な音源分離手法を模索することもできるが，多種多様な楽器音や混合条件下で動作することが求められる上に，楽器によっては十分な学習データを確保することも難しい．

そこで本稿では，混合楽器音から合成パラメータを求める問題を，音源分離問題と分離音からの合成パラメータ回帰問題に分解して扱うのではなく，観測された混合楽器音をよりよく説明するような各楽器音の合成パラメータを推定する逆問題として定式化する．具体的には，各楽器音が DDSP のデコーダ以降の部分から出力されているとみなし，それらの出力の和により混合楽器音を表現する混合 DDSP モデルを提案する．各デコーダの入力である合成パラメータを変数とみなし，事前学習した DDSP を用いて構成した混合 DDSP モデルを混合楽器音にフィッティングすることで，合成パラメータを混合楽器音から直接推定する．

3.2 混合 DDSP モデルを用いた合成パラメータ推定の定式化とパラメータ推定アルゴリズム

本節では，混合 DDSP モデルを用いた合成パラメータ推定問題の定式化を行い，パラメータ推定アルゴリズムを与える．混合楽器音に含まれる音源数を R とし，$r = 1, \ldots, R$ を音源インデックスとする．各 DDSP に対する合成パラメータと合成信号を区別するため，以下では $f_t, \boldsymbol{z}_t, l_t, \hat{\boldsymbol{x}}$ に下付き添え字 r を付け表す．r 番目の DDSP デコーダ以降の部分を DDSP_r で表すと，混合 DDSP モデルは以下のように表せる．

$$\hat{\boldsymbol{y}} = \sum_{r=1}^{R} \hat{\boldsymbol{x}}_r \tag{4}$$

$$\hat{\boldsymbol{x}}_r = \mathrm{DDSP}_r(\{f_{r,t}, \boldsymbol{z}_{r,t}, l_{r,t}\}_{t=1}^{T}) \tag{5}$$

これを用いて，混合 DDSP モデルを観測混合楽器音 $\boldsymbol{y} \in \mathbb{R}^N$ にフィッティングする問題は，$\boldsymbol{y}$ と $\hat{\boldsymbol{y}}$ のマルチスケールスペクトルロス $L(\boldsymbol{y}, \hat{\boldsymbol{y}})$ を最小化する問題として定式化できる（図 2 参照）．

$$\min_{\{f_{r,t}, \boldsymbol{z}_{r,t}, l_{r,t}\}_{r=1,t=1}^{R,T}} L(\boldsymbol{y}, \hat{\boldsymbol{y}}) \tag{6}$$

ここで，DDSP_r として事前学習した DDSP のデコーダ以降の部分を用いる．

式 (6) で与えられる最小化問題は，合成パラメータを適当に初期化した後，勾配降下法を用いて解くことができる．勾配降下法を用いるためには，ロス関数から変数に対する勾配が計算できる必要がある．L は $\hat{\boldsymbol{y}}$ に関して可微分であり，DDSP_r を構成する DNN や演算も全て可微分であるため，連鎖律により $f_{r,t}, \boldsymbol{z}_{r,t}, l_{r,t}$ の勾配は計算できる．また，この勾配計算は PyTorch や Tensorflow などの深層学習フレームワークを用いて容易に実装できる．本稿では，DDSP の学習とこの最小化問題を解くステップを区別するため，当該ステップをフィッティングと呼ぶ．

3.3 楽譜情報を援用した F_0 とラウドネスの初期値設計

勾配降下法は一般に初期値により得られる解が変わりうる．そのため，所望の解を得るには適切な初期値を与えることが重要である．本節では，MIDI データとして事前に得られた楽譜情報を援用した $f_{r,t}, l_{r,t}$ の初期値設計法を提案する．簡単のため，ここでは実演奏と楽譜情報の時間的対応づけは得られているものとする．

図 3 に，MIDI データとそれに対応する F_0 とラウドネスの初期値の例を示す．音源 r に対応する時刻 t の MIDI ノートナンバーを $p_{r,t} = -1, 0, \ldots, 127$ で表す．ただし，$p_{r,t} = -1$ は無音，すなわち音符がない場合を表す．この

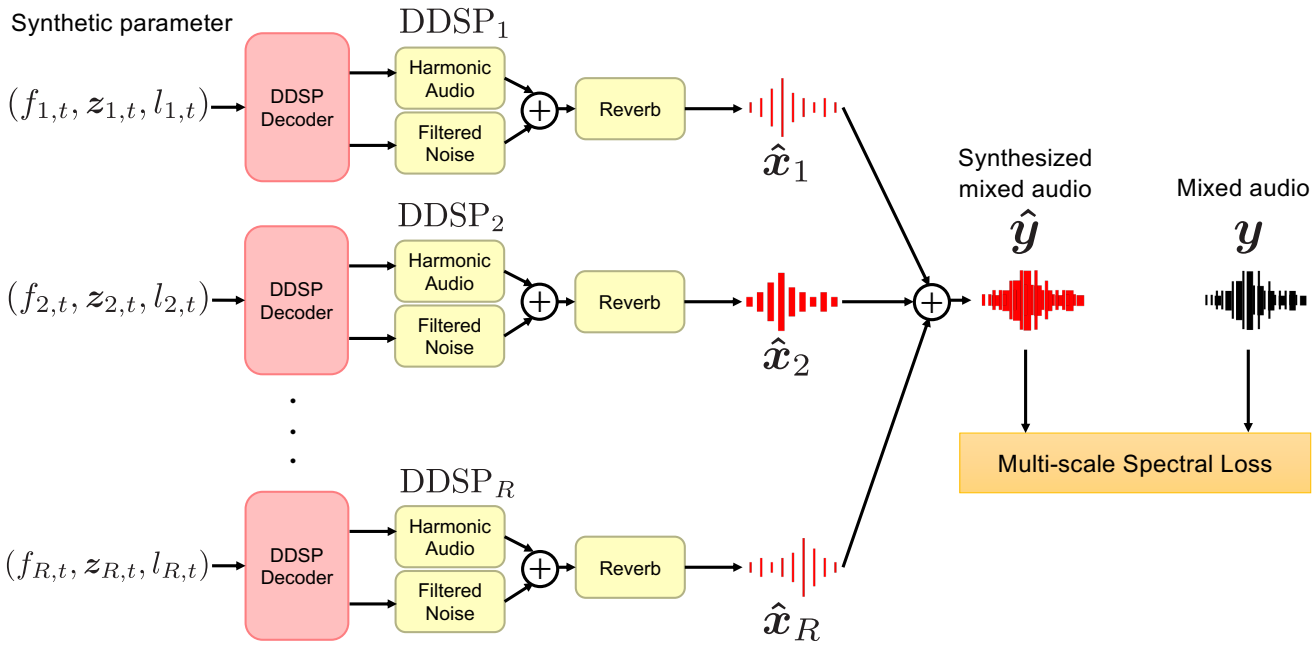

図 2: 混合楽器音に含まれる音源数が R のときの混合 DDSP モデル

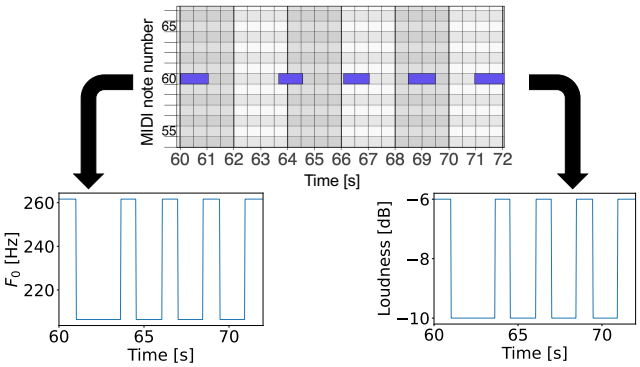

図 3: MIDI データを用いた合成パラメータの F_0 とラウドネスの初期値の例

とき，$f_{r,t}$ の初期値は次式で与えられる．

$$f_{r,t} = \begin{cases} 440 \times 2^{(p_{r,t}-69)/12} & (p_{r,t} \geq 0) \\ 440 \times 2^{(p_r^{(\mathrm{sil})}-69)/12} & (p_{r,t} = -1) \end{cases} \tag{7}$$

ここで，$p^{(\mathrm{sil})}$ は当該楽曲の音源 r の音符がある時間区間の MIDI ノートナンバーの平均である．図 3 の例では $p_r^{(\mathrm{sil})} = 53.7$ である．ラウドネス $l_{r,t}$ は，音符がある時間区間には $l^{(\mathrm{high})}$，ない時間区間には $l^{(\mathrm{low})}$ で初期化する．図 3 の例では，$l^{(\mathrm{high})} = -6$，$l^{(\mathrm{low})} = -10$ とした．

4. 実験的評価

4.1 実験条件

提案法の有効性を確認するため，楽譜情報を用いた混合楽器音からの合成パラメータ推定実験を行った．テストデータセットとして PHENICX-Anechoic データセット [15,19] を用いた．このデータセットは 4 曲の交響曲に対する各楽器の実演奏と，実演奏とアライメントの取れた楽譜情報が

表 1: 実験に用いた楽曲

Label	Musical piece
bruckner	Symphony no. 8, II movement by A. Bruckner
mahler	Symphony no. 1, IV movement by G. Mahler
mozart	An aria of Donna Elvira from the opera Don Giovanni by W. A. Mozart

表 2: テストデータの楽器構成

Label	Instruments	Total dur. [s]
Fl./Db.	flute, double bass	84
Fl./Vc./Va.	flute, violin cello, viola	120
Fl./Vc./Cl./Bn.	flute, violin cello, clarinet, bassoon	216
Fl./Fl./Fl.	flute, flute, flute	84

MIDI データとして配布されている．本実験では，4 曲のうち表 1 に示す mozart，mahler，bruckner の 3 曲を使用した．テストデータは，表 2 に示した 4 通りの楽器構成で作成した．Fl./Db.，Fl./Vc./Va.，Fl./Vc./Cl./Bn. はそれぞれ bruckner，mahler，mozart に含まれる対象楽器音を混合し作成した．Fl./Fl./Fl. は bruckner，mahler，mozart に含まれるフルートの楽器音を冒頭から混合し作成した．各テストデータは 16 kHz にダウンサンプリングした後，冒頭から 12 s 毎のセグメントに分割しそれらのセグメントに対して独立に各手法を適用した．分割後に 12 s に満たないセグメントは用いなかった．

本実験では以下の 3 つの手法を比較した．ただし，DDSP は 4.2 節に示す手順で事前に学習したものを共通して用い，合成パラメータ $f_{r,t}, z_{r,t}, l_{r,t}$ のフレーム間隔は 32 ms（$T = 375$ に対応）とした．

SS+DDSP: 文献 [16] で提案された楽譜情報を援用した音源分離手法を適用した後，各楽器の分離音に対して CREPE，DDSP のエンコーダ，ラウドネスの計算手法を用いて F_0，音色特徴量，ラウドネスを算出する．音源分離手法のパラメータは文献 [16] の実験と同一とし，実装として `https://github.com/AntonioJMM/OISS_Minus-One.github.io` を用いた．

SS+Proposed: SS+DDSP を用いて得られた F_0，ラウドネスを初期値として，混合 DDSP モデルをフィッティングする．$\boldsymbol{z}_{r,t}$ は標準正規分布からの乱数で初期化した．フィッティングでは Adam を用いて最適化した．反復数は 5000 とし，1000 ステップまでは学習率を 0.1，2000 ステップまでは 0.01，それ以降は 0.001 とした．マルチスケールスペクトルロスでは，フレーム長は $8, 16, 32, 64, 128, 256$ ms，フレームシフトはそれぞれ $2, 4, 8, 16, 32, 64$ ms とし，STFT には Hann 窓を用いた．

Proposed: 3.3 節で提案した楽譜情報を援用した F_0，ラウドネスの初期化法を用いて，混合 DDSP モデルをフィッティングする．F_0 とラウドネスの初期化は，3.3 節で提案した方法を用い，$\boldsymbol{z}_{r,t}$ は標準正規分布からの乱数で初期化した．ラウドネスの初期化に用いる値は，実験的に $l^{(\mathrm{high})} = -6$，$l^{(\mathrm{low})} = -10$ とした．フィッティングに用いる最適化手法や学習率のスケジュールは SS+Proposed と同一とした．

F_0 に対する評価指標として，正解楽器音に CREPE を適用して得られた対数 F_0 と各手法により推定された対数 F_0 の平均絶対値誤差を用いた．単位は cent とした．提案法によって推定された F_0 は負値になることもあったため，本実験では対数 F_0 の計算の前に，$f_{r,t}$ を 10^{-7} Hz でフロアリングした．また，文献 [10] の評価指標に倣い，CREPE によって得られる F_0 の推定信頼度が正解楽器音に関して 0.85 以上の個所のみを用いた．

ラウドネスに関する評価指標として，正解楽器音から得られたラウドネスと各手法のラウドネスの推定値の平均絶対値誤差を用いた．音色の評価指標として，正解楽器音と各手法で推定した合成パラメータから合成した音響信号の MFCC の平均絶対値誤差を用いた．MFCC は，フレーム長を 128 ms，フレームシフトを 32 ms として，20 から 8000 Hz に対応する 128 チャネルのメルフィルタバンクを用いて計算した．MFCC の次元は 30 とした．

4.2 DDSP の事前学習

DDSP は，University of Rochester Multimodal Music Performance（URMP）データセット [20] を用いて事前に学習した．URMP データセットは 44 曲のクラシック楽曲で構成されており，そのうち学習データとして 35 曲を用いた．学習データの構成を表 3 に示す．学習データの楽曲を 12 s 毎に分割して各セグメントとした．分割は各楽曲先頭から順に処理し，12 s に満たないセグメントは後ろに零詰めを行い 12 s の長さにした．DDSP の DNN の構造は，文献 [10] と同一のものを用いた．コスト関数としてマルチスケールスペクトログラムを用い，学習率を 0.001 とした Adam で 3000 エポック学習した．マルチスケールスペクトログラムはフィッティングの際と同一のフレーム長，フレームシフト，時間窓を用いた．

表 3: DDSP の学習データの構成

Instrument	# of musical pieces	Total dur. [s]
violin	26	2237
flute	17	1638
trumpet	19	1662
saxophone	10	754
violin cello	8	873
viola	10	865
clarinet	8	833
trombone	8	737
tuba	4	420
oboe	5	519
bassoon	3	215
horn	5	572
double bass	2	198

4.3 結果

表 4 に，全手法の合成パラメータ推定性能を示す．SS+DDSP は Proposed に比べ F_0 とラウドネスに関してともに平均絶対値誤差が大きかった．この原因の調査のため著者の 1 人が分離音を聴取してみたところ，対象楽器が強調されているものの分離音に他の楽器音が部分的に含まれていた．特に，Fl./Fl./Fl. では，他のパートの音が互いのパートに部分的に割り込むような分離結果となっていた．また，SS+DDSP で得られた合成パラメータを用いて DDSP を駆動すると，複数の楽器で音色が部分的にシンセサイザで生成したような機械的な音色となった．したがって，単純に音源分離手法と DDSP を組み合わせるだけでは必ずしも十分な精度で合成パラメータを推定できない．

一方，楽譜情報を用いて初期化した提案法である Proposed は，SS+DDSP や SS+Proposed に比べ高精度に合成パラメータを推定できた．比較的安定して F_0 とラウドネスの平均絶対値誤差も小さく，特に同一楽器の混合音である Fl./Fl./Fl. でも，F_0 に関しては Fl./Vc./Cl./Bn. と同程度，ラウドネスに関しては Fl./Db. と同程度の精度で推定できた．MFCC に関しては，Fl./Vc./Cl./Bn. 以外では提案法を用いた場合の平均絶対値誤差が小さかった．これらの結果は，提案法は混合楽器音からある程度高精度に合成パラメータを推定できることを示している．また，Proposed と SS+Proposed を比較すると，F_0 とラウドネスに関しては SS+Proposed の平均絶対値誤差が大きかっ

表 4: 混合楽器音に対する各手法の推定性能

Instruments	Method	F_0 [cent]	Loudness [dB]	MFCC
Fl./Db.	SS+DDSP	1158	2.44	3.34
	SS+Proposed	1168	2.15	2.67
	Proposed	**86.7**	**1.09**	**2.53**
Fl./Vc./Va.	SS+DDSP	1387	2.71	3.33
	SS+Proposed	1383	3.01	**3.01**
	Proposed	**298**	**1.64**	3.02
Fl./Vc./Cl./Bn.	SS+DDSP	682	3.11	2.69
	SS+Proposed	526	3.56	**2.57**
	Proposed	**93.8**	**1.59**	2.87
Fl./Fl./Fl.	SS+DDSP	116	3.42	2.22
	SS+Proposed	119	4.05	**2.01**
	Proposed	**50.6**	**1.17**	2.04

た．両者の差は初期化のみであるため，この結果から提案法において初期値の設計は重要であり，楽譜情報を用いた設計が有効であることを確認できた．

Proposed において，Vc. や Va. の含まれる混合音では合成パラメータの平均絶対値誤差が比較的大きかったが，DDSP での合成音を聴取したところ，正解音から得た合成パラメータを用いて合成した音であっても実楽器音と乖離があった．そのため，推定精度の低下は提案法ではなく学習した DDSP が原因である可能性がある．この問題は，DDSP の学習方法や学習データなどを適切に変更することにより低減しうる．

5. まとめ

本稿では，混合楽器音から各楽器の DDSP の合成パラメータを得るため，DDSP のデコーダ以降の部分を楽器音生成モデルとみなしそれらを混合し混合楽器音を表現する混合 DDSP モデルを提案した．提案法では，事前学習した DDSP を用いて，混合 DDSP モデルと混合楽器音の間のマルチスケールスペクトルロスを最小化するように勾配降下法を用いて合成パラメータを反復更新する．また，合成パラメータのうち F_0 とラウドネスに関して，楽譜情報を援用した初期化法を提案した．合成パラメータ推定実験により，音源分離手法を適用した後 DDSP を適用するだけでは合成パラメータを十分な精度で推定できないことを確認し，提案法の方が高精度に合成パラメータを推定できることを示した．

参考文献

[1] K. Yoshii, M. Goto, K. Komatani, T. Ogata and H. G. Okuno: Drumix: An audio player with real-time drum-part rearrangement functions for active music listening, *The Journal of Information Processing Society of Japan*, Vol. 2, No. 2, pp. 601–611 (2007).

[2] T. Nakamura, H. Kameoka, K. Yoshii and M. Goto: Timbre replacement of harmonic and drum components for music audio signals, *in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 7470–7474 (2014).

[3] 澤田 隼，深山 覚，後藤 真孝，平田 圭二：TransDrums：ドラムのフィルインとドラムパターン遷移確率に着目した 2 曲間のドラムパターン対応付け手法，情報処理学会論文誌，Vol. 61, No. 5, pp. 768–776 (2020).

[4] K. Itoyama, M. Goto, K. Komatani, T. Ogata and H. G. Okuno: Integration and adaptation of harmonic and inharmonic models for separating polyphonic musical signals, *in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. I–57–I–60 (2007).

[5] N. Ono, K. Miyamoto, H. Kameoka and S. Sagayama: A real-time equalizer of harmonic and percussive components in music signals, *in Proceedings of the International Conference on Music Information Retrieval*, pp. 139–144 (2008).

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

[7] C. Donahue, J. McAuley and M. Puckette: Adversarial audio synthesis, *in Proceedings of International Conference on Learning Representations* (2019).

[8] B. Hayes, C. Saitis and G. Fazekas: Neural waveshaping synthesis, *in Proceedings of International Society for Music Information Retrieval (2019) [arXiv: 1802.04208]* (2021).

[9] M. Michelashvili and L. Wolf: Hierarchical timbre–painting and articulation generation, *in Proceedings of the International Society for Music Information Retrieval Conference*, pp. 916–922 (2020).

[10] J. Engel, L. Hantrakul, C. Gu and A. Roberts: DDSP: Differentiable digital signal processing, *in Proceedings of International Conference on Learning Representations* (2020).

[11] F.-R. Stöter, A. Liutkus and N. Ito: The 2018 signal separation evaluation campaign, *in Proceedings of International Conference on Latent Variable Analysis and Signal Separation*, pp. 293–305 (2018).

[12] K. Shibata, E. Nakamura and K. Yoshii: Non-local musical statistics as guides for audio-to-score piano transcription, *Information Sciences*, Vol. 566, pp. 262–280

(2021).

[13] Y.-N. Hung, G. Wichern and J. Le Roux: Transcription is all you need: Learning to separate musical mixtures with score as supervision, *in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 46–50 (2021).

[14] M. Miron, J. Janer and E Gómez: Monaural score-informed source separation for classical music using convolutional neural networks, *in Proceedings of the International Society for Music Information Retrieval Conference*, pp. 55–62 (2017).

[15] M. Miron, J. J. Carabias-Orti, J. J. Bosch, E. Gómez and J Janer: Score-informed source separation for multichannel orchestral recordings, *Journal of Electrical and Computer Engineering*, Vol. 2016 (2016).

[16] A. J. Munoz-Montoro, J. J. Carabias-Orti, P. Vera-Candeas, F. J. Canadas-Quesada and N. Ruiz-Reyes: Online/offline score informed music signal decomposition: application to minus one, *EURASIP Journal on Audio, Speech, and Music Processing*, Vol. 2019, No. 23 (2019).

[17] J. W. Kim, J. Salamon, P. Li and J. P. Bello: Crepe: A convolutional representation for pitch estimation, *in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 161–165 (2018).

[18] V. Narayanaswamy, J. J. Thiagarajan, R. Anirudh and A. Spanias: Unsupervised audio source separation using generative priors, *in Proceedings of INTERSPEECH*, pp. 2657–2661 (2020).

[19] J. Pätynen, V. Pulkki and T. Lokki: Anechoic recording system for symphony orchestra, *Acta Acustica united with Acustica*, Vol. 94, No. 6, pp. 856–865 (2008).

[20] B. Li, X. Liu, K. Dinesh, Z. Duan and G. Sharma: Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications, *IEEE Transactions on Multimedia*, Vol. 21, No. 2, pp. 522–535 (2019).