Vol.2019-CG-176 No.23
Vol.2019-DCC-23 No.23
Vol.2019-CVIM-219 No.23
2019/11/8

MotionGAN: 関節パラメータの敵対的学習による動作スタイル生成

渡辺 祐貴[1,a)]　中澤 篤志[1,b)]　幸村 琢[2,c)]

概要：人間の動作には必ず固有のスタイルが存在し，キャラクタアニメーションの生成や person identification など，動作に関するタスクにおいて重要な要素である．そのため，動作スタイルに関して様々な研究が行われてきたが，そういった従来の研究ではを離散的なクラスとして識別・生成を行うことが多い．ところが，スタイルは年齢や心理状態といった連続的な要因に基づいて現れるため，semantic な連続空間として扱うのが妥当である．そこで，本研究ではそのような動作スタイルに関する連続空間を，深層学習と adversarial learning を用いて獲得する．具体的には，Adaptive Instance Normalization によるスタイル制御を導入した新しい動作生成ネットワークを提案，stylized された動作の生成を学習することで，中間層において目的のスタイル空間を獲得する．これにより，様々なスタイルの動作を生成するだけでなく，連続空間上での補間や加減算による新規スタイルの生成を行うことが出来る．また，スタイルを保持しながら任意の入力 trajectory に従うような動作を生成することで，controllable な動作生成を実現するとともに，明示的に動作のコンテンツとスタイルを分離して学習することが可能となる．また，従来の多くの研究では学習に用いるモーションデータに対して foot contact や phase のアノテーションを行い，データ間で厳密なアライメントを取る必要があったが，本手法ではそういったデータの前処理を必要としないため，データ作成のコストを大幅に削減できる．

1. はじめに

動作スタイルとは人間の心理状態や属性などに起因して動作に現れる特徴であり，動作の理解において非常に重要な要素である．アニメーション制作においてキャラクタを特徴付けることはもちろん，スタイルの分析により個人の同定が可能であり (person identification), また感情のような内部状態の推定にも利用できる (emotion estimation). そのため，動作スタイルに関する研究は様々に行われてきたが，その多くはスタイルを happy，angry といった感情ラベルのように離散的に扱うにとどまっている．しかし，感情に限らず身体的特徴，年齢など様々な要素に起因する動作スタイルの性質を捉えるには，そのようなクラス分割では不十分である．また，人間は初見の動作であっても，年齢や属性，心理状態などをある程度把握することが出来るため，動作スタイルには何らかの認知的 parameter space が存在するはずである．そこで本研究では，そのような動作スタイルに関する連続空間を，深層学習を用いて実際の動作データ (モーションキャプチャデータ) から bottom-up に獲得することを目的とする．本論文では，adversarial learning を導入した学習フレームワークと動作生成に適したネットワーク構造を提案，中間層において動作のコンテンツに依存しない連続スタイル空間が獲得できることを確認した．またこの連続空間を利用して，スタイルの類似関係の分析や，空間上での加減算によるスタイル間の連続補間・新規スタイルの生成を行うことが可能となる．さらに，深層学習を動作生成に適用した従来手法の多くは，データの前処理として厳密なタイミングの alignment や foot contact のアノテーションが必要であった．しかし提案手法ではそういった処理を要さないため，様々なモーションデータを学習に利用することが可能であり，また学習データを用意するコストも大幅に削減される．

[1] 京都大学
Kyoto Uniersity
[2] エジンバラ大学
The University of Edinburgh
[a)] watanabe@ii.ist.i.kyoto-u.ac.jp
[b)] nakazawa.atsushi@i.kyoto-u.ac.jp
[c)] tkomura@ed.ac.uk

2. 関連研究

2.1 Learning for motion

機械学習の動作生成への応用については，多くの研究が行われてきた．Brand ら [1] は隠れマルコフモデルを用いて動作とスタイルの状態空間モデルを学習，stylized された動作の生成を行った．Lee ら [13] は Motion Field とい

うデータ構造を提案，Example-base で動作の多様体を構成し，ユーザがその上での生成のコントロールを行えるシステムを構築した．ユーザの入力に対して最も適した動作を出力するために，マルコフ決定過程における強化学習を行っている．

Grochow[6] らは Gaussian Process Latent Variable Model (GPLVM) を動作生成に適用し，キーフレーム補間や欠損データ補完などの様々なタスクにおいて従来の Inverse Kinematics(IK) と置き換わる手法を提案した．この手法では IK を元のポーズと GPLVM のモデルパラメータに対する出力ポーズの尤度の最大化として定式化，モデルパラメータが学習データのスタイルを反映するとしてスタイルに基づいた IK を提案している．

2.2 Deep Learning for motion

近年では深層学習における研究の急速な進歩に伴い，動作生成にも深層学習を取り入れた手法が数多く提案されている．

Holden ら [9] は，Convolutional Neural Network(CNN) でオートエンコーダを構成，モーションキャプチャデータを用いた動作の多様体学習を行うことで，ある動作に対して類似した多様な動作を生成することを可能にした．

Ghosh ら [4] は Recurrent Neural Network(RNN) を利用した次フレーム予測手法を提案，過去フレームとの長・短期依存に基づく先の動作の予測生成を可能にした．一般に RNN による動作生成は自己回帰を繰り返すことによって出力が不安定になったり逆に収束してしまうという問題があるが，この手法では事前学習したオートエンコーダを利用してノイズの除去を行った．また Li ら [14] はこの問題に対して，auto-conditioned LSTM (acLSTM) というネットワークを提案，歩行や走行動作の安定した生成に加え，ダンスやアクロバットのような複雑な動作の生成を実現した．

Holden らの手法 [8] では，Phase-Functioned Neural Network(PFNN) という動作生成に適したネットワーク構造を提案．動作の位相 (Phase) という重要な性質を明示的にネットワークの構造に反映した．具体的にはネットワークの重みを動作の位相に応じて動的にコントロールすることで，ユーザのコマンドや地形に応じた歩行・走行動作生成を実現した．Zhang ら [17] は 4 足動物の歩行・走行動作のような複数のモードが存在する動作の学習において，それらが平均化されてしまう問題に対して，PFNN[8] にモードをコントロールするパラメータを導入したネットワーク構造を提案した．

Peng ら [16] では，動作生成に深層強化学習を応用している．高レベル (低周波) の動作と低レベル (高周波) の動作の学習を別ユニットに分離することで，決められた道筋をたどる・サッカーのドリブルをするなどの高レベルなタスク設定に対してパラメータを直接学習することに成功した

2.3 Style Learning

スタイルの研究は画像分野でも盛んに行われており，特に近年では CNN をスタイル変換・生成に適用する手法が数多く提案されている．

Gatys ら [3] は CNN の中間特徴のグラム行列がスタイルを表現すると仮定し，これを入力画像と目的スタイルで近づけることでスタイル変換を実現した．具体的には，入力画像と目的のスタイル画像を学習済み CNN(VGG) に入力，中間特徴のグラム行列の L2 ノルムを目的関数として，入力画像のピクセル値を直接最適化する．Holden[7] らはこのアイデアを動作生成に導入，VGG の代わりに動作データを潜在変数空間にマッピングするネットワークを学習し，グラム行列を近づけることでスタイル変換を行う．これらの手法は参考データ (目的スタイルのデータ) として任意のデータを設定することが出来る一方で，毎変換ごとに最適化を行う必要があり計算コストが高い．

Dumoulin らの手法 [2] ではスタイルを CNN の中間特徴の平均及び分散でをコントロール出来るとした．CNN に設けた Instance Normalization(IN) における平均・分散をスタイルパラメータとして各スタイルごとに学習し，変換の際にはそれを用いた正規化を適用するだけでスタイル変換を行うことができる．

Huang ら [10] は，IN の平均・分散を各スタイルごとに学習するのではなく，参考画像から直接算出する手法 Adaptive Instance Normalization(AdaIN) を提案，一度の学習で任意スタイル変換を可能にした．Karras ら [11] は AdaIN を Generative Adversarial Networks (GAN) のフレームワークに導入，Latent Transform というサブネットワークを用いて潜在変数 z で AdaIN のパラメータをコントロールする構造を提案し．従来の GAN より質の高い画像の生成を実現した．また，画像から潜在変数 z へのマッピングを学習することで，任意の参考画像を用いた Stylized 生成を行うことが出来る．

3. 提案手法

本章では，提案手法およびデータに関する処理について述べる．提案手法の概要図を図 1 に示す．

3.1 モーションデータの前処理

3D モーションデータの表現方法としては，各関節の 3D Euler 角，quarternion，exponential map など様々な方法が広く用いられているが，本研究では 3 次元座標による表現を採用する。具体的には，各フレームにおいてグローバ

Vol.2019-CG-176 No.23
Vol.2019-DCC-23 No.23
Vol.2019-CVIM-219 No.23
2019/11/8

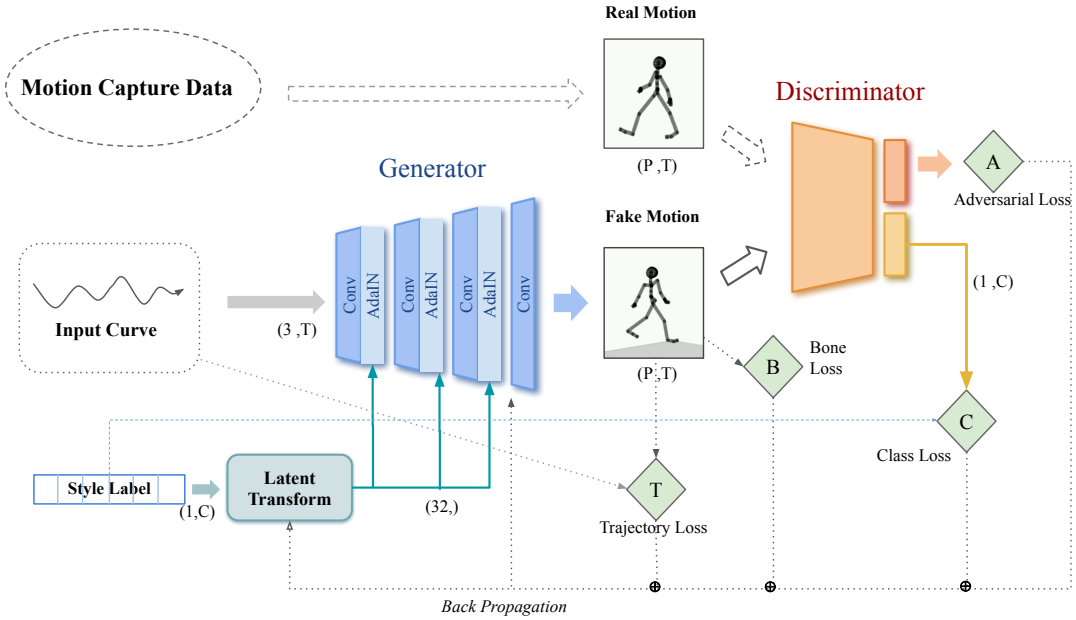

図 1 提案手法の概要図．ネットワークとしては，2 つの CNN(Generator, Discriminator) に加えてサブネットワーク Latent Transform がある．Generator は input curve とスタイルラベルを入力として Fake Motion を生成する．このとき，スタイルラベルは Latent Transform に入力され，Generator の各 AdaIN への入力パラメータに変換される．Discriminator にはこの fake sample あるいはモーションキャプチャデータセットからサンプリングした Real Motion を入力する．Discriminator は 2 つのヘッドを持ち，一方 (図中赤) はサンプルの学習データセットに対する尤度を，他方は各クラス (スタイル) に対する尤度を出力する．図中の各菱形は 3.3.1 節に示した Generator の損失関数を構成する各項を表す．

ル座標系における root position およびそれを原点とする各 joint の相対座標によって，動作を指定する．また，多くの従来手法においては，前処理として foot contact や phase のアノテーションや，異なるスタイルの動作間でのタイミングのアライメントを行う必要があるが，提案手法ではそのような作業は行わない．

3.1.1 モーションのデータ構造

本研究ではモーションのデータとして各関節の 3 次元座標を用いる．関節 p の時刻 t における座標を $({x_p}^t, {y_p}^t, {z_p}^t)$ とすると，時刻を列方向として全関節について行方向に並べることで，行列として動作を表現することができる．この行列は，関節数を P，フレーム長を T とすると，(P × 3) 行 T 列の行列となる．また，正規化のため roo t joint の座標 $({x_0}^t, {y_0}^t, {z_0}^t)$ を原点とした相対座標 $({x'_p}^t, {y'_p}^t, {z'_p}^t) = ({x_p}^t - {x_0}^t, {y_p}^t - {y_0}^t, {z_p}^t - {z_0}^t)$ を求めて動作のパラメータとし，root position の時系列 $({x_0}^t, {y_0}^t, {z_0}^t)(t = 0, 1, 2, ...)$ を trajectory として用いる．したがって，動作および trajectory は次式のように表現される．

$$motion = \begin{bmatrix} {x'_0}^0 & {x'_0}^1 & \dots & {x'_0}^T \\ {y'_0}^0 & {y'_0}^1 & & {y'_0}^T \\ {z'_0}^0 & {z'_0}^1 & & {z'_0}^T \\ {x'_1}^0 & {x'_1}^1 & \ddots & {x'_1}^T \\ \vdots & \vdots & & \vdots \\ {z'_P}^0 & {z'_P}^1 & \dots & {z'_P}^T \end{bmatrix} \quad (1)$$

$$trajectory = \begin{bmatrix} {x_0}^0 & {x_0}^1 & \dots & {x_0}^T \\ {y_0}^0 & {y_0}^1 & \dots & {y_0}^T \\ {z_0}^0 & {z_0}^1 & \dots & {z_0}^T \end{bmatrix} \quad (2)$$

.

3.1.2 input curve の生成

提案手法では，生成を自在にコントロール可能にするために，キャラクタの Trajectory をネットワークに入力する．この trajectory とは root position の軌道を指すが，学習に用いるモーションキャプチャデータに記録された root position は，厳密なタイミングや phase に起因する微妙な振動など多くの情報を保持しており，想定しうるユーザの入力とはかけ離れているうえ，生成の汎化性を損なう原因となる．また，そこで，モーションキャプチャデータの正確な trajectory から，より情報量が少なく抽象的な曲線を

Vol.2019-CG-176 No.23
Vol.2019-DCC-23 No.23
Vol.2019-CVIM-219 No.23
2019/11/8

抽出する必要がある．以後，この曲線を input curve と呼び，図 2 および以下の手順によって作成する．

1. y 座標 (高さ) を定数にする．
2. 一定フレーム間隔でコントロールポイントを設定，cubic spline 補間を行う．
3. ネットワークの入力フレーム数に合わせて，等間隔で座標をサンプリングする．

手順 1 では，trajectory を 2D 平面に射影する．これは，ユーザ入力としてキャンパスに描写するような直感的なインターフェースを想定しているためである．手順 2 では，trajectory から先述のような高周波の振動を除去するために，spline 補間を行っている．最後に，タイミングの情報を取り除くために，時間ではなく距離に基づいた等間隔のサンプリングを行う．このようにして入力として用いる抽象的な曲線を得る．逆に言えば，モデルは各 joint の動きだけでなく，root position についてここで取り除かれた情報 (高さやタイミング, 高周波成分) を補い生成するように学習する．

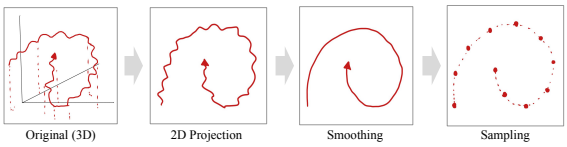

図 2　Input curve の生成手順．モーションキャプチャデータに記録された 3D trajectory から 2D の曲線を抽出する．

3.2　ネットワーク

提案手法のネットワークは，Generator および Discriminator の 2 つの CNN および Generator のサブネットワーク Latent Transform から構成される．各ネットワークの構造を図 3 に示す．

3.2.1　Generator

Generator は 14 層の 2D 畳み込み層とそれに続く AdaIN ユニットから成る Encoder-Decoder 構造をしており，UNet と同様の skip connection を含む．またこれに加えて，スタイルの潜在空間の獲得および AdaIN のパラメータ制御のために，全結合層のみから成るサブネットワーク Latent Transform を設けている．　Encoder-Decoder 部分の入力は先述の input curve であり，2x フレーム数の形をしている．encoder 部分ではこれに kernel size 2x5 の畳み込みを行って 1 次元に圧縮したのち，隔層で時間方向の downsampling を行っていく．一方，decoder 部分では隔層で時間方向だけでなく関節方向にも upsampling を行っていき，最終的に 81x フレーム数という形でモーションデータを出力する．skip connection では，encoder の中間層の feature map(1 次元) を，対応するの Decoder の中間層の feature map の高さに合うように複製し，チャネル方向に結合する．さらに，decoder の最後には root trajectory を出力するための畳み込み層を設けている．　また，スタイル生成を行うため，各畳み込み層の後には AdaIN ユニットを入れている．各 AdaIN ユニットでは，パラメータ β および γ に基づいて以下のように feature map をチャネルごとに正規化する．

$$IN(x) = \gamma_{lk}\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta_{lk} \tag{3}$$

$beta_{lk}$，γ_{lk} は第 l 層目の各 AdaIN ユニットの第 k チャネルの正規化パラメータである．提案手法では，スタイルラベルを Latent Transform により潜在空間に埋め込んだ共通の潜在変数 w を入力として，各 AdaIN ユニットに含まれる 1x1 畳み込み層による変換を通して求められる．

3.2.2　Discriminator

Discriminator は 4 層の 2 次元畳み込み層と 2 層の全結合層から成る．各畳み込み層では feature map に対し 2 次元の downsampling を行うとともに，Spectral Normalization[15] を適用する．2 層の全結合層はともに最後の畳み込み層の出力 Feature を入力とする並列構造となっており，1 つは入力 $\mathbf{x}$ の真のサンプル集合 $\mathbf{R}$ に対する尤度 $P(\mathbf{R}|\mathbf{x})$ を出力し，もう 1 つは各スタイル c_k に対する尤度 $P(c_k|\mathbf{x})$ を出力する．

3.3　学習

Generator の Encoder-Decoder 部分は Full Convolution Neural Network(FCNN) なので入力長は任意であるが，学習の際は固定長の motion clip と input curve のペアを real sample として用いる．

3.3.1　損失関数

Generator の損失関数は下式で表される Adversarial Loss, Classification Loss, Curve Loss, Bone Loss の 4 項の重み付き和である．実験では、$\lambda_{Adv} = 1.0$, $\lambda_{Cls}= 5.0$, $\lambda_{Curve}= 0.05$, $\lambda_{Bone}0.1$ とした。

$$\begin{aligned} L_G = \quad & \lambda_{Adv} * L_{Adv} + \lambda_{Class} * L_{Class} \\ & + \lambda_{Curve} * L_{Curve} + \lambda_{Bone} * L_{Bone} \end{aligned}$$

$$L_{Adv} = -\mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}(\boldsymbol{x})}[\log D(G(\boldsymbol{x}))] \tag{4}$$

Adversarial Loss には Modified GAN Loss[5] を用いる．G, D はそれぞれ Generator，Discriminator が表現する関数．$\boldsymbol{x}$ は input curve.

$$L_{Class} = \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}(\boldsymbol{x}),C}[\log P(C|G(\boldsymbol{x}))] \tag{5}$$

Classification Loss としては，fake sample について Dis-

Vol.2019-CG-176 No.23
Vol.2019-DCC-23 No.23
Vol.2019-CVIM-219 No.23
2019/11/8

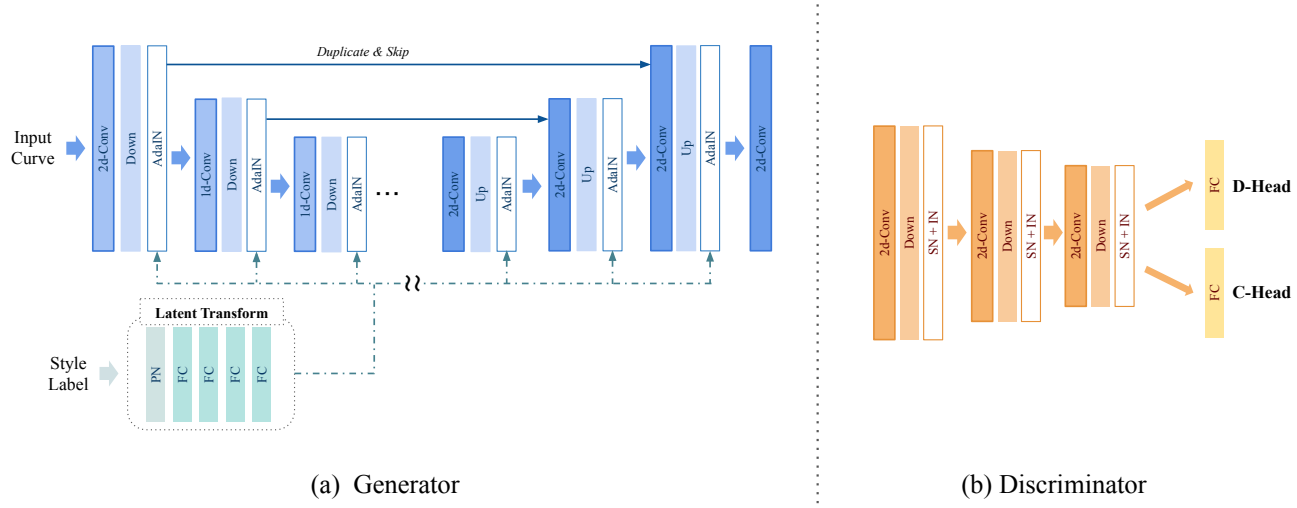

図 3 Generator および Discriminator のネットワーク構造.

criminator が出力したスタイル尤度の Binary Cross Entropy をとる．C は各スタイルに対応するクラスを表す．

$$L_{Curve} = \sum_{n=0}^{4} \left(\left(\boldsymbol{s}_0 + \sum_{t=0}^{T_k} \boldsymbol{v}_t \right) - \boldsymbol{s}_{T_k} \right) \tag{6}$$

Curve Loss は，Generator が出力する root position が入力の input curve に沿うように制約する項であり，それぞれから固定数 (実験では 5 つ) の点をサンプリングし，ユークリッド距離上の二乗誤差を取る．$\mathbf{v}_t$，$\mathbf{s}_t$ はフレーム t における root position の速度と座標，T_k(k=0,1,2,3,4) はサンプリングする際のフレーム番号を表す．

$$L_{Bone} = \sum_{t} \sum_{(i,j)\in\{S\}} (\|\boldsymbol{m}_i^t - \boldsymbol{m}_j^t\| - b_{ij}\|) \tag{7}$$

Bone Loss は各関節の距離が伸び縮みしないように制約する項である．出力されたモーションの 2 次元配列において，あらかじめスケルトンで親子関係にある関節に対応する行 i,j の組み合わせを全て求めておき，2 点の距離とスケルトンにおけるボーンの長さの二乗誤差を計算，全関節および全フレームで総和を取る．$\boldsymbol{m}_i^t$ は生成データのフレーム t における関節 i の座標，b_{ij} はスケルトンにおいて i 番目の関節と j 番目の関節をつなぐボーンの長さ，S はスケルトンに含まれるボーンの集合を表す．

Discriminator の損失関数は Adversarial Loss, Classification Loss の 2 項の和である。Adversarial Loss には Modified GAN Loss を、Classification Loss は Real Sample について各クラス尤度の BCE を用いる。

$$L_D = L_{Adv} + L_{Class} \tag{8}$$

$$L_{Adv} = \mathbb{E}_{\boldsymbol{m}\sim p_{data}} \log D(\boldsymbol{m}) + \mathbb{E}_{\boldsymbol{x}\ p_{\boldsymbol{x}}(\boldsymbol{x})} 1 - \log D(G(\boldsymbol{x})) \tag{9}$$

$$L_{Class} = \mathbb{E}_{\mathbf{m}\ \mathbf{p_{data}},\mathbf{C}} \log P(C|\mathbf{m}) \tag{10}$$

3.3.2 Optimization

Optimizer には Adam[12] を使用、β_1=0.5、β_2=0.999 とした。学習率は G が 0.0002、D が 0.0001 とした。各 iteration につき G・D ともに 1 度ずつ update し、200,000 iteration で学習を打ち切った。

3.3.3 Arbitrary Paring of Style and Trajectory

Generator が fake sample を生成する際にはモーションキャプチャデータから抽出された input curve に加えて，スタイルラベルを入力するが，このとき元データのスタイルに限定せず全スタイルからランダムに選択したものを用いる．これにより，学習時の入力の input curve とスタイルの組み合わせを格段に増やすことが出来，任意の input curve に対して正しく stylized されたモーションを生成することができる。

3.3.4 FPS の Random Argumentation

本研究ではスタイル空間の学習のために，同一の input curve から様々なスタイルのモーションを生成するというタスクでモデルを学習する．この際，スタイルによって動きのスピードが異なるため、同一の input curve に対しても異なる時間のモーションの出力が必要である．ところが，CNN では入出力の長さの対応は固定であるため，同一の入力に対して出力フレーム長を直接変化させることは難しい．そこで，出力において FPS に幅を持たせることで固定のフレーム長でも異なる時間長のモーションを表現できるようにする．具体的には，学習時に実データの FPS をランダムに変化させて入力することで，出力の FPS に曖昧性を許容するような Discriminator が学習され，Generator は input curve とスタイルにそれぞれ矛盾しないような FPS での出力を学習することができる．

Vol.2019-CG-176 No.23
Vol.2019-DCC-23 No.23
Vol.2019-CVIM-219 No.23
2019/11/8

4. 実験

提案手法の性能を評価するため，モーションキャプチャデータを用いてモデルの学習・動作生成を行った．また，学習により獲得した連続スタイル空間について，連続補間生成や主成分分析による可視化など様々な分析を行い，その妥当性や意味を確認した．

4.1 データセット

学習用のモーションキャプチャデータセットとしてはCMU Motion Capture Data に含まれる styled Walk(29 style) を使用した．これらは 2 人のモーションアクターがそれぞれ 14,15 種類のスタイルの歩行モーションを演じて収録したデータである．データセットに含まれるスタイルの詳細や各データの総フレーム数については表 () に記載した．また，テストに用いる input curve は，CMU データの Locomotion カテゴリに含まれる別のアクターの歩行動作データから抽出したものを用いた．これらのモーションキャプチャデータはすべて bvh フォーマットに準じており，root joint を含め計 28 関節のフレームごとの角度が 3 次元 Euler 角で記述されている．実験ではそれらを 3 節の方法で 3 次元相対座標に変換して用いる．いずれのデータも 120FPS で収録されているが，実験では計算資源が限られている都合より，15FPS にダウンサンプルして学習・生成を行った．

4.2 生成結果

Test データから抽出した input curve からの生成結果のいくつかを図??に示す．またスタイルとしては，”Elated”, ”GanglyTeen”, ”Joy”, ”Sneaky”, ”Depressed”の 5 つについてそれぞれ生成した．いずれの生成結果でも入力したスタイルに特徴づけられた歩行動作が生成できている．

4.3 スタイルの連続空間

提案手法では，input curve からの任意スタイル動作生成というタスクについて学習を行うことで，同時に Latent Transform による離散スタイルラベルから連続潜在変数 **w** へのマッピングを学習できる．この連続潜在変数 **w** は Generator の AdaIN レイヤのコントロールパラメータであるから，**w** はスタイルの連続空間を構成する．実際にこの空間に主成分分析をかけて 2 次元に次元圧縮し可視化したものを図 5 に示す．また，**w** 空間上において各スタイルラベルの embed のユークリッド距離を比較し，各スタイルについて距離の近いもの上位 3 つを表 1 に示した．主成分分析で圧縮された空間において，Sad-Shy-Depressed, Elated-Lavish-Joy-GracefulLady, Sexy-SexyLady, Sneaky-Scared といった類似したスタイルが概ね近い値に分布していることがわかる．また表 1 の結果から，元の **w** 空間においても

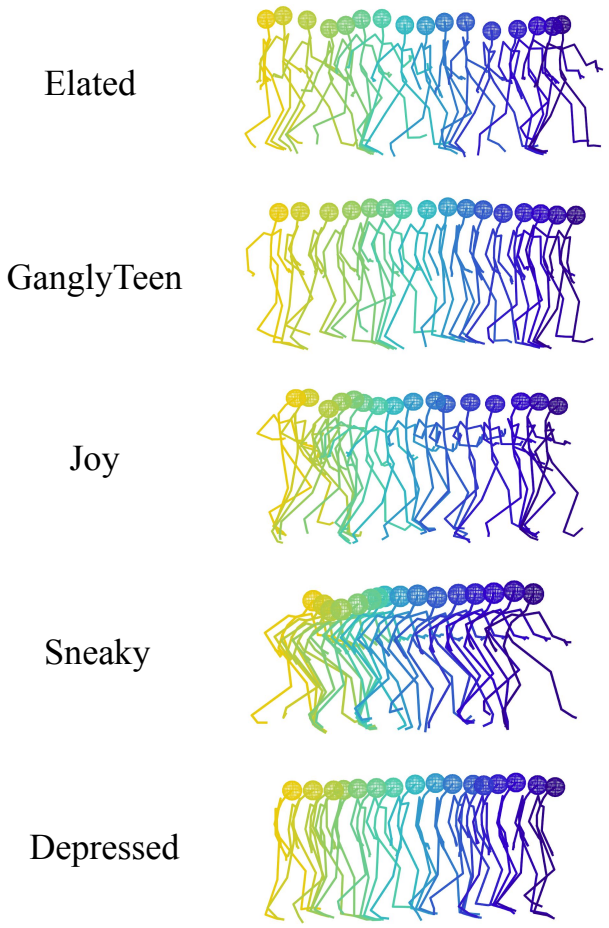

図 4 異なるスタイルでの歩行動作の生成結果．

類似したスタイル同士が近い場所に Embed されていることが確認できる．なお，スタイルのラベルはあくまで主観的な名前付けであり，必ずしもその言語的な類似性が動作の類似性と対応しないことは注意されたい．例えばスタイルラベル”OldMan”と”Elderlyman”はラベルの意味はほぼ同義であるが，別々のモーションアクターが演じているため，姿勢や動作の速度に比較的大きな差異があり，スタイルとしての類似度はそれほど高くない．

特筆すべき点として，今回使用したデータセットでは必ずしも動作の content(ここでは歩行経路やタイミング)は共通でない．例えばスタイルラベル Sexy に属する動作データとスタイルラベル SexyLady に属する動作データは全く異なる経路での歩行データから成る (モーションアクターも異なる)．このようなデータでは，動作データ同士の類似度を直接学習するだけではスタイル空間を適切に学習出来ず，動作のスタイルとコンテンツを分離するような仕組みが必要となる．提案手法では input curve(コンテンツに対応) からの生成というタスク設定や，AdaIN による明示的なパラメータの分離を行うことで，そのような分離を実現しコンテンツに依存しないスタイルの連続空間を獲得

Vol.2019-CG-176 No.23
Vol.2019-DCC-23 No.23
Vol.2019-CVIM-219 No.23
2019/11/8

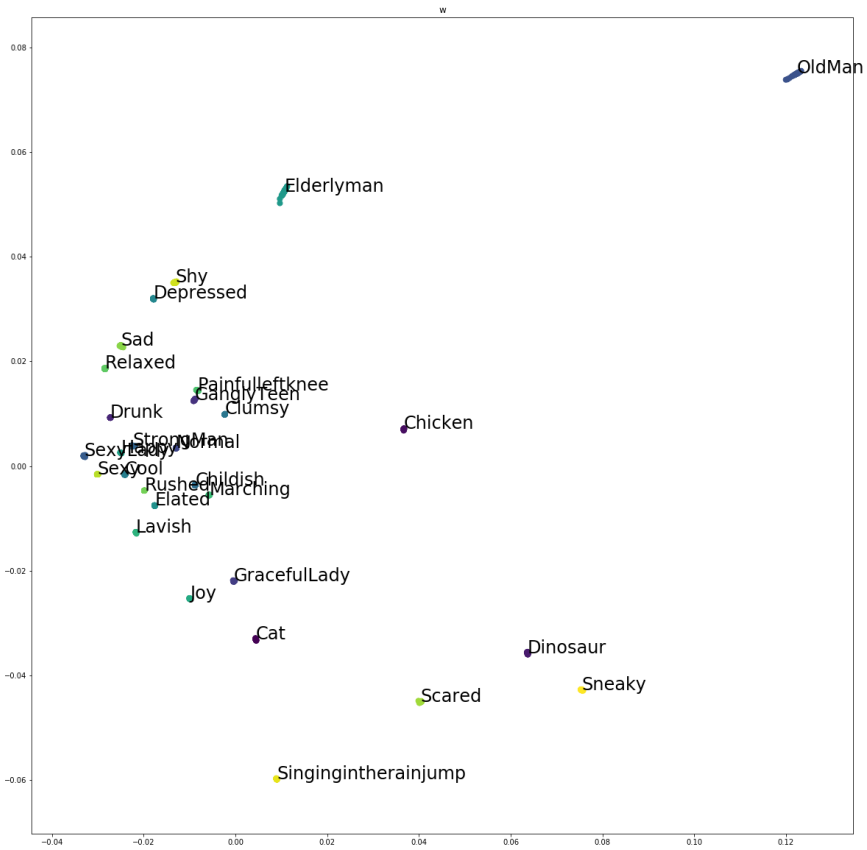

図 5 Latent Transform ユニットの出力であり，Generator の各 AdaIN レイヤのコントロールパラメータである潜在変数 $\mathbf{w}$ の空間について，主成分分析により 2 次元への圧縮を行ったもの．プロットされている各点はそれぞれ各スタイルラベルの embed を表す．

することが出来る．

4.4 スタイル空間における演算

学習データではスタイルは離散ラベルとしてのみ存在するが，提案手法で学習したスタイル空間 w は連続であるため，その上でのベクトル同士の加算・減算を行うことが出来る．具体的には，ある 2 つの異なるスタイルの embed の内分をとることで，スタイルの間を連続補間するような動作が生成出来たり，スタイル同士の加算減算により新しいスタイルを生み出すことができる．例として，2 つのスタイル”Depressed”と”Happy”の間の連続補間生成と，3 つのスタイル”OldMan”,”Normal”,”Marching”を用いて”OldMan”-”Normal”＋”Marching”という演算を行って生成した動作を図 6,7 に示す．

図 6 の動作は，最上段と最下段がそれぞれ”Depressed”，”Happy”をスタイルとしてそのまま入力した生成動作である．そして，その間は上から順に 1:4, 2:3, 3:2, 4:1 という比率で内分を取り生成した動作である．これらの中間動作では，両端のそれぞれのスタイルの特徴が内分比率通りに現れている上で，自然な動作になるよう足を出すタイミングやスピードが調整されており，両端の生成動作を直接混ぜ合わせただけのものとははっきりと異なることが確認できる．

図 7 には，”OldMan”, ”Normal”, ”Marching”の 3 つの異なるスタイルの Embed$\boldsymbol{w}_{OldMan}, \boldsymbol{w}_{Normal}, \boldsymbol{w}_{Marching}$ を加算減算した $\boldsymbol{w}_{OldMan} - \boldsymbol{w}_{Normal} + \boldsymbol{w}_{Marching}$ を入力して生成した動作と，元となる 3 スタイルを入力した動作をそれぞれ示した．演算により”OldMan”に”Marching”の特徴が加わった新しいスタイルが生成できていることがわかる．

5. Discussion&Conclusion

本研究では，スタイルを考慮したキャラクタ動作生成に適したネットワークおよび学習のフレームワークを提案し

Vol.2019-CG-176 No.23
Vol.2019-DCC-23 No.23
Vol.2019-CVIM-219 No.23
2019/11/8

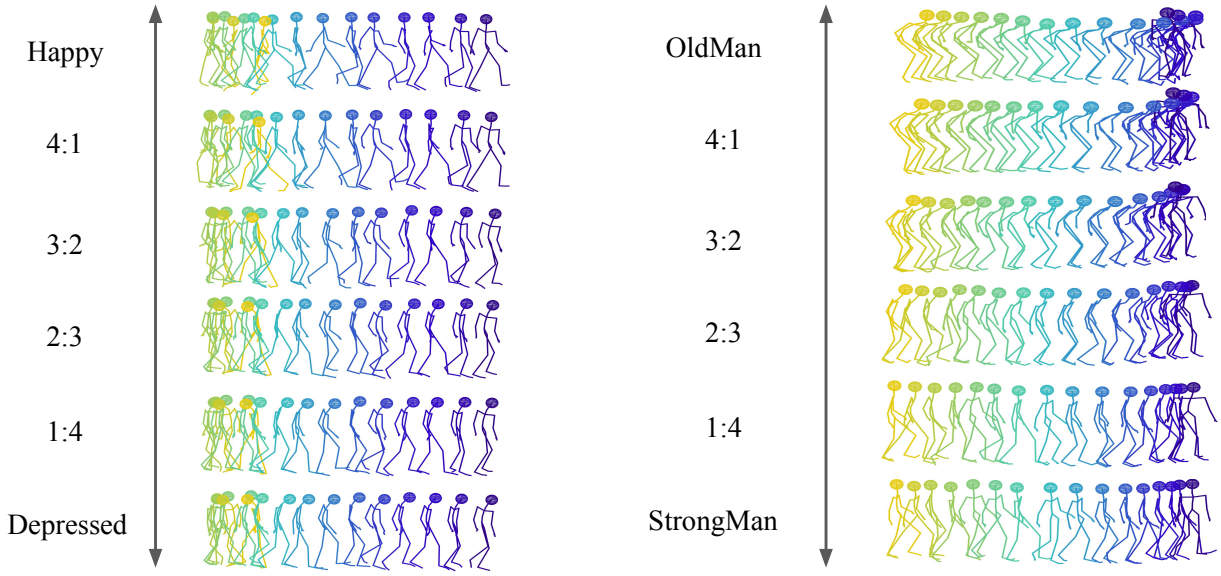

図 6 スタイル連続空間における 2 つのスタイル間の補間生成.

Base Style	First	Second	Third
Cat	Normal	Rushed	Elated
Chicken	GanglyTeen	Shy	Elderlyman
Childish	Cool	Sad	GanglyTeen
Clumsy	Drunk	Elated	GanglyTeen
Cool	Sad	Relaxed	SexyLady
Depressed	Sad	Relaxed	Cool
Dinosaur	Scared	Sneaky	Marching
Drunk	Clumsy	Relaxed	Sad
Elated	Happy	GanglyTeen	Cool
Elderlyman	Depressed	Sad	GanglyTeen
GanglyTeen	Elated	Cool	Sad
GracefulLady	Normal	GanglyTeen	Elated
Happy	Elated	Sad	Relaxed
Joy	Elated	Rushed	Cool
Lavish	Sexy	Cool	Marching
Marching	Lavish	GracefulLady	Elated
Normal	Relaxed	Sad	Happy
OldMan	Elderlyman	Dinosaur	Sneaky
Painfulleftknee	Rushed	Sad	GanglyTeen
Relaxed	Sad	SexyLady	Cool
Rushed	Sexy	Painfulleftknee	Normal
Sad	Depressed	Relaxed	Cool
Scared	Childish	Singingintherainjump	Dinosaur
Sexy	SexyLady	Cool	Relaxed
SexyLady	Relaxed	Cool	Sexy
Singingintherainjump	GracefulLady	Scared	Cool
Shy	Sad	Depressed	Relaxed
StrongMan	Sad	Cool	GanglyTeen
Sneaky	Scared	Dinosaur	Cat

表 1 潜在変数 **w** の空間における各スタイルラベルの embed について，距離の近いスタイルラベル上位 3 つを示す．最も左の列に基準となるスタイルラベルを，その右に近い順に 3 つのスタイルラベルを記載した．

た．提案手法では，従来手法のようなデータに関する厳密な制約や前処理を要さず．質の高い動作の生成が可能である．加えて提案手法では，中間出力としてコンテンツと分離したスタイルの連続空間を獲得，スタイル間の関係性の分析や，embed を用いた演算による新規スタイルの生成を行うことが出来る．

本研究では歩行動作という限られた動作のみでの生成を行ったのみであり，より充実した style 動作データセットを作成することで提案手法の汎化性能を確認する必要がある．また，キャラクタ動作生成にスタイルを関連付けた研究は数多く存在するものの，スタイルそのもの定義・分析は未だ十分になされておらず，今後本研究の発展としてスタイルに関してより詳細な分析を行っていく．

謝辞 本研究は JST CREST JPMJCR17A5 の支援を受けたものである．本研究で使用した CMU Graphics Lab Motion Capture Database は NSF EIA-0196217 の出資により作成され，mocap.cs.cmu.edu で提供されている．

参考文献

[1] Matthew Brand and Aaron Hertzmann. Style machines. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 183–192. ACM Press/Addison-Wesley Publishing Co., 2000.

[2] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

[3] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

[4] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*, pp. 458–466. IEEE, 2017.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville,

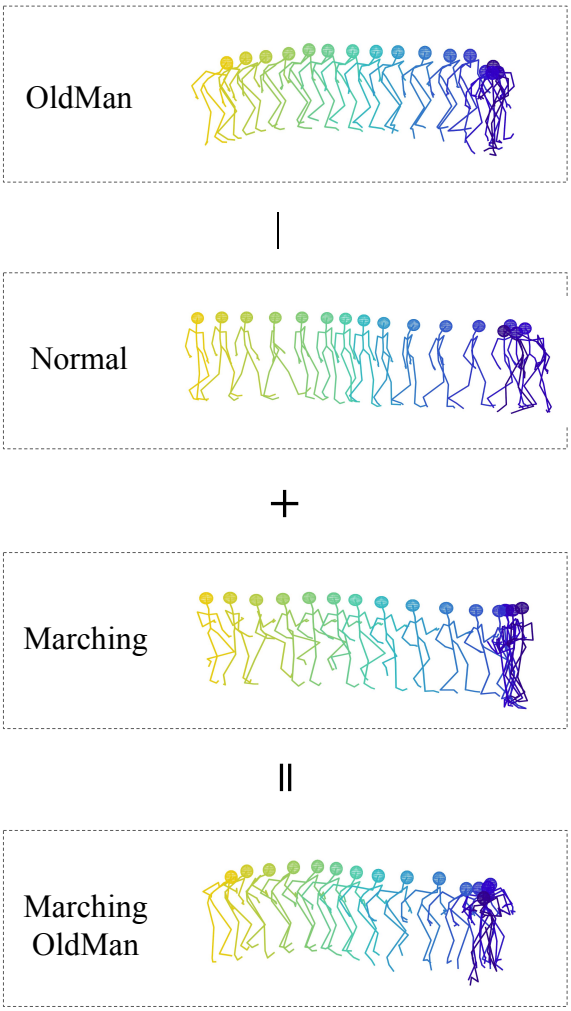

図 7　スタイル連続空間での加算減算による新規スタイル生成.

and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[6] Keith Grochow, Steven L Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. In *ACM transactions on graphics (TOG)*, Vol. 23, pp. 522–531. ACM, 2004.

[7] Daniel Holden, Ikhsanul Habibie, Ikuo Kusajima, and Taku Komura. Fast neural style transfer for motion data. *IEEE computer graphics and applications*, Vol. 37, No. 4, pp. 42–49, 2017.

[8] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, Vol. 36, No. 4, p. 42, 2017.

[9] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, p. 18. ACM, 2015.

[10] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510, 2017.

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Yongjoon Lee, Kevin Wampler, Gilbert Bernstein, Jovan Popović, and Zoran Popović. Motion fields for interactive character locomotion. In *ACM Transactions on Graphics (TOG)*, Vol. 29, p. 138. ACM, 2010.

[14] Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, and Hao Li. Auto-conditioned lstm network for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363*, Vol. 3, , 2017.

[15] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[16] Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, Vol. 36, No. 4, p. 41, 2017.

[17] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)*, Vol. 37, No. 4, p. 145, 2018.