I-023

RNN を用いたデータ駆動型モーションリターゲット
Data-driven Motion Retargeting Using Recurrent Neural Networks

出口　風人[1)]　遠藤　結城[1)]　栗山　繁[1)]
Kazato Deguchi　Yuki Endo　Shigeru Kuriyama

1　研究背景

CG キャラクタアニメーションの制作技術として，人のモーションキャプチャデータを体形の異なるキャラクタに転写するモーションリターゲットが知られている．しかし，古典的な最適化ベースの手法はモーションに応じた手動制約を必要とし [1]，より高品質なモーションデータを生成するにはアニメータによる熟練した手作業も要求される．

本稿では，深層学習によってリターゲット前後の変換をニューラルネットワークに学習させることで，高品質なアニメーションの自動生成を目指す．近年 encoder-decoder モデル [2] を用いた手法が提案されているものの，足の滑りや地面との接地といった問題があり，実データに対する性能は未だ十分とは言えない．そこでこのモデルをベースに，損失関数において末端の関節とルートに重み付けをすることでリターゲットの精度向上を目指す．最後に，実データを用いた評価実験によりその効果を定性的および定量的に検証する．

2　手法

本手法では Recurrent Neural Network (RNN) と Forward Kinematics (FK) を用いたモーションリターゲット手法である Neural kinematic networks [2] をベースとする．損失関数の計算において関節ごとに重みを設定することで，末端の関節等のモーションの見映えに大きく影響する重要な関節を相対的に強く学習させる．

2.1　基礎となるモデル構造

学習モデルは，図 1 に示すように各関節の変動を学習する RNN encoder-decoder 層とモーションのローカル座標を導出する FK 層に分かれている．変換対象の入力モーション $x_{1:T} \in \mathbb{R}^{(3J+4)\times T}$ は，T フレーム分の各関節のローカル座標 $p_{1:T} \in \mathbb{R}^{3J\times T}$ およびルートの速度と回転 $v_{1:T} \in \mathbb{R}^{4\times T}$ から構成される．J はモーションデータの関節数である．RNN はエンコード層とデコード層に分かれており，入力モーションは次式のように処理される．

$$
\begin{aligned}
h_t^{enc} &= RNN^{enc}(x_t, h_{t-1}^{enc}; W^{enc}) \\
h_t^{dec} &= RNN^{dec}(\hat{x}_{t-1}, h_t^{enc}, \bar{s}, h_{t-1}^{dec}; W^{dec}) \\
\hat{q}_t &= \frac{W^{pT} h_t^{dec}}{||W^{pT} h_t^{dec}||} \\
\hat{p}_t = FK(\hat{q}_t, \bar{s}), &\quad \hat{v}_t = W^{vT} h_t^{dec}, \quad \hat{x}_t = [\hat{p}_t, \hat{v}_t]
\end{aligned}
$$

ここで，$W^{enc}, W^{dec}, W^v \in \mathbb{R}^{d\times 4}$ および $W^p \in \mathbb{R}^{d\times 4J}$ は学習パラメータである．

エンコード層には入力として x_t を与え，出力される潜在変数 h_t^{enc} と骨格情報 $\bar{s} \in \mathbb{R}^{3J}$ をデコード層に与える．それに加え，1 フレーム前のデコード層より出力される h_{t-1}^{dec} も入力として与えることで，潜在変数 h_t^{dec} が得られる．これに線形変換を適用することで，四元数 $\hat{q}_t \in \mathbb{R}^{4J}$ とリターゲットされたルートの速度と回転 $\hat{v}_t$ が導出される．そして，$\hat{q}_t$ と $\bar{s}$ を入力に FK を用いることで，各関節のローカル座標 $\hat{p}_t$ が導出され，出力モーション $\hat{x}_t$ が得られる．

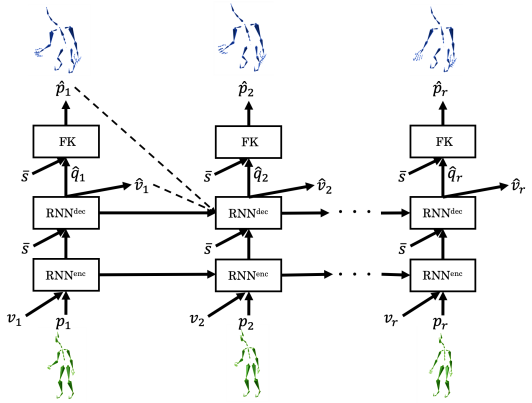

図 1　Neural kinematic networks の構造．ローカル座標 p_t とルートの速度と回転 v_t をエンコード層に入力する．その出力と骨格 $\bar{s}$ をデコード層に入力することで，四元数 $\hat{q}_t$，リターゲットされたルートの速度と回転 $\hat{v}_t$ が導出される．$\hat{q}_t$ と $\bar{s}$ を入力とし，FK によりリターゲットされたローカル座標 $\hat{p}_t$ が導出される．

2.2　提案損失関数

先行手法 [2] では Cycle consistency loss，Adversarial loss，Twist loss，Smoothing loss の 4 種の損失関数を用いて次式の通り学習する．

$$Loss = L_C + L_A + \lambda L_T + \omega L_S$$

Cycle consistency loss L_C は骨格 A のモーション x^A と，骨格 B のモーション $\hat{x}^B$ を再び骨格 A へとリターゲットした $\hat{x}^A$ より導出される損失関数であり，リターゲットの再現性を学習する．Adversarial loss L_A は骨格 A のモーション x^A に対して，変換された骨格 B のモーション $\hat{x}^B$ が本物らしいモーションであるかを評価する．Twist loss L_T は関節における捻れの異常を防ぎ，Smoothing loss L_S はルート移動の滑らかさを維持する．

本手法では，損失関数においてローカル座標を扱っている Cycle consistency loss と Adversarial loss に対して重み付けをするために，p_t と同形の行列のマスク $M_{j,t} \in \mathbb{R}^{3J\times T}$ を導入する．FK によりルートからの誤差が蓄積していくため，骨格における末端がもっとも正解データと誤差が大きくなると考えられる．そこで骨格の四肢の末端と頭頂部を重点的に学習するように重み付けすることにした．さらに全体の動きに大きく影響するルートにも重み付けする．学習時に n 倍重視したい関節の集合を J_E，それ以外の関節の集合を J_N とすると，マスク $M_{j,t}$ は次式のように表せる．

$$
M_{j,t} = \begin{cases} \frac{n(|J_E|+|J_N|)}{n|J_E|+|J_N|} & (j \in J_E) \\ \frac{|J_E|+|J_N|}{n|J_E|+|J_N|} & (otherwise) \end{cases}
$$

1)　豊橋技術科学大学, Toyohashi University of Technology

ここで，j は関節の ID, $|J_E|$ および $|J_N|$ は集合の要素数を表す．各フレームにおける $M_{j,t}$ の総和は関節数と同じ値になるようにした．

マスク M を用いた損失関数は次の通りである．

$$L_C = ||M \circ p^A_{1:T} - M \circ \hat{p}^A_{1:T}||_2^2 + ||v^A_{1:T} - \hat{v}^A_{1:T}||_2^2$$

$$L_A = \begin{cases} ||M \circ p^A_{1:T} - M \circ \hat{p}^B_{1:T}||_2^2 + ||v^A_{1:T} - \hat{v}^B_{1:T}||_2^2 & (A = B) \\ \log r^A + \beta \log(1 - r^B) & (otherwise) \end{cases}$$

ここで，$\circ$ は要素積を表し，r^A は真のデータによる Discriminator の出力，r^B は偽のデータによる Discriminator の出力を表す．本手法では既存手法と同様に 20%の確率で入力と生成モーションの骨格を同じに設定して学習するため，Adversarial loss では骨格 A と骨格 B が同一の場合，これらの二乗誤差を計算する．

3 実験

提案手法により学習したモデルを用いて実データをリターゲットし，生成データと正解データとの誤差により先行手法 [2] とリターゲットの精度を比較した．関節の重み付けにおける比率は，$n = 2, 4, 8$ で検証した．

3.1 データセット

データセットには Mixamo[2] に公開されている 3D ヒューマンモーションデータのうち X bot，Warrok W Kurniawan を使用した．これらのモーションデータは 65 個の関節からなるが，本手法では主要関節である 22 個の関節を学習に用いる．トレーニングデータには 659 種類のモーションの各々を 60 フレームごとに分割した 1247 個のデータを学習に用いた．ただし，端数のフレームについては，直前のフレームを結合することで 60 フレームにした．テストデータには 387 種類のモーションの各々を 120 フレームごとに分割した 775 個のデータを用いた．テストデータの検証には入力として X bot の骨格を有したモーションデータを使用し，Warrok W Kurniawan の骨格へのリターゲットを各条件で生成し，正解データと比較した．

3.2 実験環境

実験環境は 10 コア 20 スレッド CPU と単一の GTX 1080 を使用しており，バッチサイズ 16，学習回数 50000 ステップで検証した．RNN 部には GRU を使用し，512 個のユニット，2 層のレイヤーからなる．学習率は 0.0001 でドロップアウト率は 0.1 とした結果，学習時間は 12 時間程度を要した．損失関数の計算においては $\lambda = 10.0$，$\omega = 0.01$ とした．最適化には Adam を用いた．

3.3 結果と考察

それぞれの生成結果のモーションデータにおける各関節の座標の値を用いて正解データと比較し，リターゲットの精度を評価する．表 1 は 775 個の各テストデータの正解データとの平均二乗誤差 (MSE) の平均と標準誤差 (SE) をローカル座標と，グローバル座標それぞれで導出したものである．各条件における MSE を比較すると，local MSE，global MSE 共にもっとも MSE が小さくなったのは $n = 2$ の時であり，$n = 4, 8$ の時は重み付けなしの場合より MSE が大きくなった．これは比率が大き過ぎたことで，他の関節を十分に学習できなかったためと考えられる．

2) https://www.mixamo.com

表 1 正解データとの MSE による評価．± の後の数字は標準誤差を表す．

条件	local MSE	global MSE
重み付けなし	62.442 ± 4.433	122.638 ± 10.219
$n = 2$	60.195 ± 3.423	120.446 ± 9.147
$n = 4$	66.827 ± 5.803	132.239 ± 11.600
$n = 8$	71.014 ± 5.687	140.593 ± 12.359

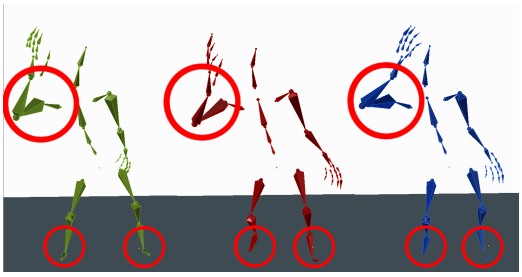

図 2 正解データと生成データの比較．正解データ (緑)，重み付けなし (赤)，$n = 2$ の場合 (青)．丸は違いが顕著な部分を表す．

重み付けなしの場合と比べ，$n = 2$ の場合に local MSE が小さいということはリターゲットによる関節角度の再現精度がより高いと言える．

各条件での global MSE の値が絶対的に大きくなっている原因としてはルートの影響が考えられる．例えば，ルート変動が大きい歩行や倒れこむような動作に対しては，ワールド座標系においては大きな誤差が生じる．

図 2 は重み付けなしの場合と，$n = 2$ でそれぞれ学習したモデルを用いてリターゲットした生成結果とその正解データの一例である．各生成データと正解データとを比較すると，全体的な関節角度は再現できたが，個々の関節を比較すると十分な質でない．重み付けなしの場合と $n = 2$ の場合を比較すると，$n = 2$ の場合の方が右腕，胴部分は正解データに近いが，他の部位は不十分であり，厳密な重みを設定することで精度が向上すると考えられる．足の接地については，生成データは共に足が地面をすり抜けており，正解データのような接地を再現できていない．ゆえに，損失関数において足の接地を考慮する手法を新たに導入する必要がある．

4 まとめと今後の課題

関節に重み付けをすることで汎用的なリターゲットの精度向上は確認できたが，個々のリターゲットの精度として比較すると未だ不十分であり，より厳密な重み付け方法の検討が必要である．特にルートの位置が大きく変動するモーションにおいては，関節角度の再現性は取れていたが，ルート位置が正解データと大きく異なってしまう現象が確認された．そのため，ワールド座標におけるルート位置の推定は別の学習手法を用いることが考えられる．また，リターゲットにより全体的な雰囲気の再現はできていても，足の接地や滑りといった幾何学的制約に対する誤差の問題は解決しておらず，入力モーションの動きとしての自然さを保持しながら，所望のリターゲットの精度を向上させる必要がある．

参考文献

[1] S. Tak and H.-S. Ko. A physically-based motion retargeting filter. In *ACM Transactions on Graphics (TOG)*, p. 24(1):98–117, 2005.

[2] Villegas, et al. Neural kinematic networks for unsupervised motion retargetting. In *CVPR*, June 2018.