Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

SPECIAL ISSUE PAPER

A Variational U-Net for Motion Retargeting

Seong Uk Kim[1] | Hanyoung Jang[2] | Jongmin Kim*[1]

[1]Computer Science and Engineering, Kangwon National University, Chuncheon, Republic of Korea
[2] Motion AI Team, Game AI Lab., NCSOFT, Seongnam, Republic of Korea

Correspondence
*Jongmin Kim, Computer Science and Engineering, Kangwon National University.
Email: jongmin.kim@kangwon.ac.kr

Summary

Motion retargeting is the process of copying motion from one character (source) to another (target) when the source and target body sizes and proportions (of arms, legs, torso, and so on) are different. The problem of automatic motion retargeting has been studied for several decades; however, the motion quality obtained with the application of current approaches is on occasion unrealistic. This is because previous methods, which are mainly based on numerical optimization, generally do not incorporate prior knowledge of the details and nuances of human movements. To address these issues, we present a novel human motion retargeting system using a deep learning framework with large-scale motion data to produce high-quality retargeted human motion. We establish a deep-learning-based motion retargeting system using a variational deep autoencoder combining the deep convolutional inverse graphics network (DC-IGN) and the U-Net. The DC-IGN is utilized for disentangling the motion of each body part, while the U-Net is employed to preserve details of the original motion. We conduct several experiments to validate the proposed motion retargeting system, and find that ours achieves better accuracy along with reduced computational burden when compared with the conventional motion retargeting approach and other neural network architectures.

KEYWORDS:
human motion, motion retargeting, deep learning

1 | INTRODUCTION

Over the past several decades, the concept of motion capture has received considerable attention, and it is now widely used in several industries including the gaming and visual effects (VFX) industries. Although the current state-of-art motion capture systems are sufficiently accurate in recording human motion, the process of modifying the obtained motion data is highly desirable for suitable motion data fitting in different scenarios for maintaining the appearance of realism. For instance, after motion data are acquired, a user may want to adapt the walking motion of a small child to that of a tall adult walking. Consequently, motion retargeting, which is the process of motion adjustment involving the copying of motion from one character (source) to another (target) when the source and targeting body sizes and proportions (arms, legs, torso, and so on) are different, becomes essential.

One of the simplest methods to retarget human motion is the manual modification of the joint angles one at a time; however, it is difficult and tedious to apply this procedure for all joints across all the motion sequences. Further, when the user attempts to manually retarget one pose to a new one for each frame, discontinuities of the joint trajectories along the timeline can occur because the character pose in a certain frame may be different from the one in the previous or next frame. Thus, joint positions that should be maintained are sometimes wrongly changed, which leads to undesirable problems such as the character's feet not

contacting the ground or the hand not being sufficiently close to an object that the character is expected to grasp. To overcome these issues, different *Inverse Kinematics* (IK)[1, 2] approaches have been proposed by many researchers as automatic and efficient solutions. In particular, numerical IK iteratively updates a set of joint angular parameters through nonlinear optimization with certain spatial constraints to be retained until the end-effector positions reach the target ones or the errors arising from the optimization are sufficiently reduced; however, the quality of the resulting motion with numerical IK is on occasion unrealistic. This is because current IK methods, which are mainly based on numerical optimization, generally do not incorporate prior knowledge of the details and nuances of actual human movements. Further, the methods are usually not sufficiently fast for retargeting many different human motions simultaneously. To remedy this problem, data-driven approaches[3, 4] have been proposed to ensure that the resulting human motion is more realistic. The main idea is to train pre-recorded motion data and establish a prediction model to statistically connect from the low-dimensional end-effectors to high-dimensional character poses. Many previous methods mostly rely on a relatively small data set and require off-line or on-line training processes. In such cases, the resulting character poses are highly dependent on the training data; thus, motion jerkiness can occur if the desired motions are very different from those of the training dataset.

Against this backdrop, based on deep learning frameworks that can efficiently handle large amounts of motion data[5–7], here, we propose a novel deep autoencoder that combines the deep convolutional inverse graphics network (DC-IGN)[8] and U-Net[9] shown in Figure 1 to efficiently retarget human motion. Our approach handles various types of motions to be retargeted and is sufficiently fast to be used as a real-time application. The DC-IGN disentangles the bone representation pertaining to arms, legs, torso, and other parts while the U-Net preserves the motion details owing to the skip connections between the encoder and decoder. We train a deep autoencoder to directly map the source motion to the target motion. Our network is fed with two inputs: source motion and a set of bone length ratios for the limbs (right arms/legs, left arms/legs, and torso) of a character. Once the neural network is trained, the proposed system is able to generate various sizes of retargeted motions by adjusting only the bone length ratio values. To completely preserve the bone length, our method also performs *kinematic optimization* in the prediction phase. In particular, we first project the predicted motion back into the latent space and subsequently optimize the intermediate latent variables to ensure that the character meets a set of bone length constraints. This optimization in latent space in the prediction phase yields a faithful resulting motion with no visible or noticeable artifacts.

The key contributions of our work are as follows:

- We propose a novel deep autoencoder for human motion retargeting with providing convenient user interfaces, and multi-characters with several spatial relationships are automatically retargeted in a faster manner.
- Our system offers easy control of motion retargeting and simultaneously transfers its original style into a different one and also can automatically fix the corrupted source motion that is inevitably obtained after capturing human motion when using the low-cost motion capture system (e.g., Inertial Motion Units (IMUs), Accelerometers, and *Microsoft Kinect*).

2 | RELATED WORK

The demand for motion retargeting has recently witnessed an exponential increase in terms of reusing motion data. Motion retargeting has been extensively studied for many decades. In this context, Gleicher[1] proposed an approach for adapting human motion to create characters with different body proportions and formulated this problem as a constrained optimization with space-time constraints. Lee and Shin[2] suggested a hierarchical motion retargeting framework satisfying certain constraints by employing a multilevel B-spline representation. When compared with the previous method[1], this method has the advantage of speed as the computationally expensive space-time optimization process is not considered. Choi and Ko[10] proposed an online retargeting approach based on the per-frame IK while considering motion similarity. Meanwhile, most recent motion retargeting methods are mainly based on the IK approach that updates several Jacobian matrices at every single iteration. In this regard, Buss[11] organized several different types of numerical IK solvers using such Jacobian matrices. The Jacobian transpose method, pseudo-inverse method, singular value decomposition (SVD), and damped least-squares (DLS) method are discussed. More information in this context can be found in the survey work of Aristidou *et al.*[12] for comprehensively understanding IK studies. Although current motion retargeting methods, which are mainly based on numerical IK, generate acceptable human motions, the retargeted motions sometimes appear unnatural due to the lack of consideration of actual human movement, thereby burdening the user with a secondary task. To address this problem, we build a deep-learning-based human motion retargeting system to accurately and sufficiently cover the space of the natural human motion manifold.

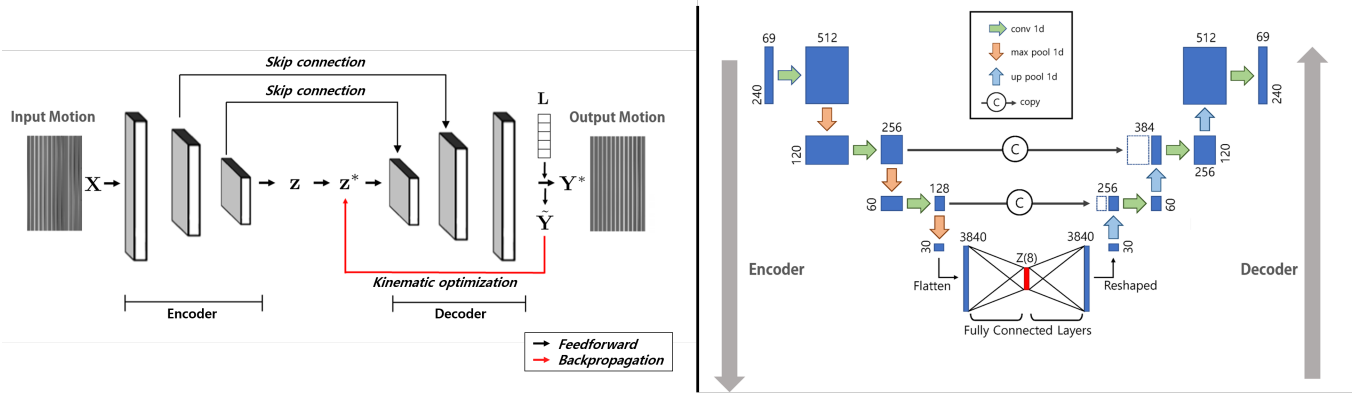


FIGURE 1 The architecture of proposed model for human motion retargeting (first column). The image in the second column shows the proposed network in detail. The input motion data are first entered into the encoder. The motion is embedded into the hidden layer by passing through each convolutional layer. Subsequently, it is converted into output motion through the decoder. Our network architecture consists of the deep convolutional inverse graphics network (DC-IGN) and the U-Net for disentangling each body part and efficiently conveying the details of motion, respectively.

Recently, deep neural networks that is capable of dealing with a large amount of motion data have frequently been employed in human motion research. In this context, Taylor *et al.* [13, 14] have employed conditional restricted Boltzmann machines (cRBM) for synthesizing the motion by inferring the pose at the next frame. Fragkiadaki *et al.* [5] proposed the encoder-recurrent-decoder (ERD) network that applies a long short-term memory (LSTM) model in the latent space for predicting the next pose of the body. Their model could handle motion capture data across multiple subjects and activity domains and synthesize novel motions. Du *et al.* [15] constructed a hierarchical recurrent neural networks (RNN) to achieve a state-of-the-art recognition accuracy. A deep learning-based motion synthesis system [6] composed of a set of convolutional neural networks (CNN) is able to map from a set of specified user inputs (e.g., positional constraints and trajectory constraints) to realistic human movement. Holden *et al.* [16] also proposed a phase-functioned neural network, which interpolates the weights of four feed-forward networks to make a single character interactively navigate on uneven terrain without foot skating. Tang *et al.* [17] introduced a RNN-based motion prediction system by analyzing the observed motion sequences. Cui *et al.* [7] proposed a bi-directional RNN with an attention mechanism to accurately infer the missing joints. In contrast to ours, unsupervised motion retargeting systems have been proposed by [18, 19]. Compared with the previous works, our system consisting of the DC-IGN and U-Net provides user-friendly control parameters that enable the user to efficiently and smoothly adjust the size and transfer style of the character to be retargeted at the same time. We note that multi-character motion retargeting using deep learning frameworks has not thus far been extensively studied. In this context, our method is different from previous ones pertaining to both the aspects of the problem and the technical approach.

3 | DATA PREPROCESSING

We use the CMU motion capture database for training our neural network [20]. Similar to the case of the motion data pre-processing proposed by [6], each character consists of $j = 21$ joints, and each motion clip to be used for the training neural network has $n = 240$ poses. Each pose is described as a set of joint positions that are relative to the root position. We also take into account the root rotational velocity and translational velocity along the X- and Z-axes as the input training features. As a result, the total number of dimensions of the input training features for a single training data set is 16,560 (69×240). We also annotate the contact state per pose about the left and right foot, and these are used to enforce foot constraints during the kinematic optimization process.

To establish the training data, we begin with the two following steps: normalizing the entire training data (motion data) for $\mathbf{X}$ and retargeting each motion to several motions with different bone lengths for $\mathbf{Y}$. In the first step, we retarget all motion data to normalize them. To do so, we find a scaling factor, which is generally the ratio between the height of the reference and non-normalized T-poses; however, it is at times inaccurate when the lengths of the arms are significantly different. To address this issue, we compute the optimal scaling factor $\alpha \in \mathbb{R}$ between the reference and non-normalized T-poses using the Procrustes method (see [21] for more details). First, we compute the optimal rigid rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ and translation $\mathbf{t} \in \mathbb{R}^3$ to transform

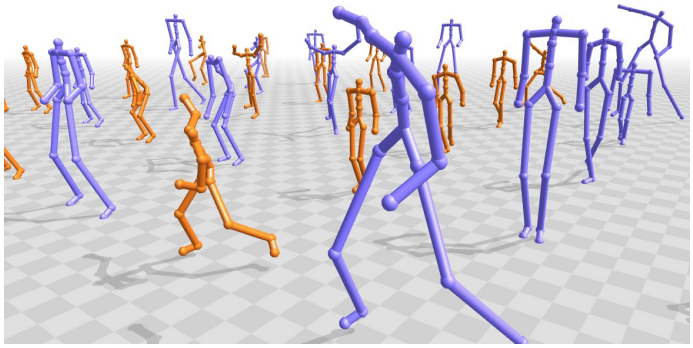


FIGURE 2 Retargeting of normalized source motions. In this example, our system retargets the normalized source motions (orange) into dozens of different types and sizes of output motions (blue) while preserving bone lengths.

the non-normalized data to the reference T-pose by minimizing $\sum_{j=1} w_j \|(\mathbf{R}\mathbf{p}_j + \mathbf{t}) - \mathbf{q}_j\|_2^2$, where $\{\mathbf{p}_j\}$ and $\{\mathbf{q}_j\}$ denote a set of joint positions of the non-normalized and reference T-poses, respectively, and w_j the weight value for the j-th joint. Next, we obtain the optimal scaling factor α by minimizing $\sum_{j=1} w_j \|\alpha\mathbf{R}(\mathbf{p}_j + \mathbf{t}) - \mathbf{q}_j\|_2^2$ with respect to α. We apply this process to each pose and subsequently solve the IK[11] equations to generate the normalized poses. In the second step, we manually resize the bone lengths based on the user-defined ratio in terms of the legs, arms, and spine across all the normalized motions and solve the IK once again. In our experiments, the bone length ratio ranges from 0.8 to 1.8 relative to the original one over uniform intervals of 0.5; therefore, 27 ($3 \times 3 \times 3$) retargeted motion clips are generated from a single motion clip because the bone lengths for the left/right arms or left/right legs should be the exactly the same. We augment the retargeting motions instead of only using the original ones because the latter are insufficient to train the deep autoencoder. We use $\sim$ 10K of the retargeted motion data for training the proposed neural network.

4 | OUR APPROACH

Figure 1 shows our deep autoencoder, which is basically constructed upon the DC-IGN which is associated with variational autoencoder. The input of the deep autoencoder is composed of joint positions $\mathbf{X} \in \mathbb{R}^{240\times69}$ whereas the output comprises the retargeted joint positions $\mathbf{Y} \in \mathbb{R}^{240\times69}$. Our deep autoencoder consists of two networks: the *encoder* that encodes source motion $\mathbf{X}$ to a latent vector $\mathbf{z} = \texttt{enc}(\mathbf{X})$ and the *decoder* that decodes it into target motion $\mathbf{Y} = \texttt{dec}(\mathbf{z})$. The bone length ratios $\mathbf{L} \in \mathbb{R}^5$ are input into the network with $\mathbf{z} \in \mathbb{R}^8$ representing a set of disentangled latent variables $z_i \in \mathbf{z}$ such as those of the arm (z_1), leg (z_2), torso (z_3), and additional parameters ($z_4, \ldots, z_8$). Here, we note that our approach automatically factorizes the latent variables after training the network. The encoder has three convolutional layers followed by max pooling, and the decoder similarly consists of four convolutional layers of upsampling. Each convolutional layer except the final layer is followed by a ReLU activation function. The U-Net is also combined with the DC-IGN as its skip connections allow the autoencoder to convey the details of the input directly to the output.

The variational distribution $\mathbf{q}(\mathbf{z}|\mathbf{X})$ is related to a prior distribution of the latent variables, and a centered multivariate Gaussian with identity covariance $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is chosen for our network. We train the network to minimize the errors measured by the Kullback-Leibler divergence, prediction, smoothness, and regularization losses. The total loss function is expressed as $-\log \mathbf{p}(\mathbf{X}|\mathbf{z}) + KL(\mathbf{q}(\mathbf{z}|\mathbf{X})||\mathbf{p}(\mathbf{z})) + \phi_p(\hat{\mathbf{Y}}, \mathbf{Y}) + \lambda_s \sum_t \phi_s(\hat{\mathbf{Y}}_t, \hat{\mathbf{Y}}_{t-1}) + \lambda_r \|\theta\|_1$, where λ_s and λ_r represent user-defined variables for the smoothness and regularization, $\mathbf{Y}_t$ denotes the pose at frame t ($1 < t \leq 240$) and $\hat{\mathbf{Y}}$ is the predicted motion. Here, $\phi(\cdot)$ denotes the $\mathcal{L}_2$ norm distance between the predicted and training motions, and θ that is to be regularized with the $\mathcal{L}_1$ norm forms the network parameter. In our experiments, we set $\lambda_s = 0.2$ and $\lambda_r = 0.01$. Our deep autoencoder disentangles representation by selectively optimizing a single latent variable while all others that do not correspond to it are clamped. For training the network, we organize the training data into mini-batches corresponding to changes in only a single variable (arm (z_1), leg (z_2), torso (z_3)). We establish the mini-batches such that the selected single variable is changed while all other variables are fixed. The total number of dimensions of each mini-batch is ($3 \times 240 \times 69$) because all the motion data are resized to 0.8, 1.3, and 1.8

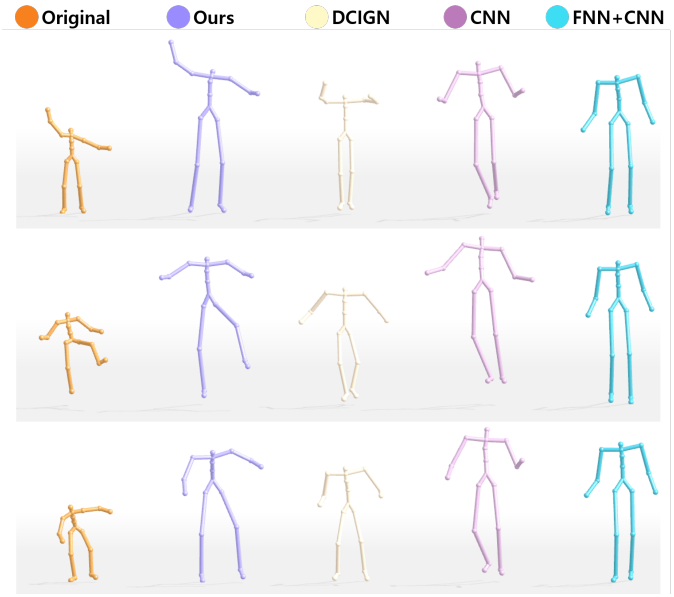


FIGURE 3 Screenshots of original and retargeted motions obtained from our approach and other networks. The columns correspond to motion generated by our method, deep convolutional inverse graphics network (DC-IGN), convolutional neural network (CNN), and feed-forward neural network (FNN)+CNN in that order. We show qualitative performance on three different networks as described in Figure 4.

times the original ones for each body part (arm, leg, and torso). The encoder takes the mini-batches as input while the decoder can generate the retargeted motion via the latent variable and the skip connection (see[8] for more details).

An initial $\tilde{\mathbf{Y}}$ is obtained by $\mathbf{X}$ (see Figure 1). In the prediction phase, we iteratively update $\mathbf{z}$ to strictly preserve the bone lengths and find an optimal latent vector $\mathbf{z}^*$ by minimizing $\|\Pi(\texttt{dec}(\mathbf{z})) - \Pi(\mathbf{Y})\|_2^2$. Here, $\Pi(\cdot)$ computes the bone lengths between adjacent joints to retain the rigidity of the skeleton. Specifically, the network parameters θ are fixed, and the adjusted latent variables $\mathbf{z}$ are obtained by performing backpropagation with the bone length error function. After the optimization process, we obtain the newly updated latent variable $\mathbf{z}^*$, and the retargeted motion $\mathbf{Y}^* = \texttt{dec}(\mathbf{z}^*)$ is computed by passing it through the decoder.

5 | EXPERIMENTAL RESULTS

Figure 2 and 5 (**b**) show multiple retargeted motions generated for random bone length ratios ranging from 0.8 to 1.8. Our method produces realistic retargeted motions even for dynamic motions such as jumping and running. This result demonstrates that our network can smoothly navigate the manifold of character motions corresponding to the user input of the bone length ratios. Figure 5 (**a**) shows that our network can also denoise and retarget the character motion; in the figure, the source motion (left)

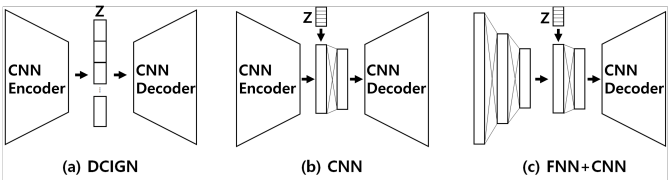
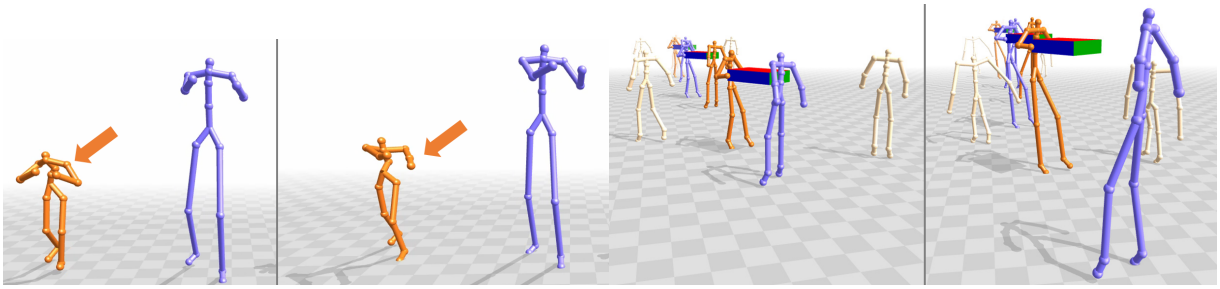


FIGURE 4 Comparison of three different types of neural networks. (a) Deep convolutional inverse graphics network (DC-IGN) structure with three convolutional layers without skip connections, (b) convolutional neural network (CNN) with convolutional layers, and (c) feed-forward neural network (FNN) + CNN with convolutional and fully connected layers. The networks in (b) and (c) use concatenated hidden vectors in the middle of network to adjust the bone lengths.



(a) (b)

FIGURE 5 (**a**): Results of motion retargeting and denosing. In this example, the results obtained using our method (blue) show that ours has the ability to automatically fix a corrupted source motion (orange) and retarget and denoise it, while preserving kinematic constraints. (**b**): In the input motion, multiple characters deliver the boxes in relay (left). The proposed deep autoencoder retargets the multi-character motion (right).

contains noise. Our method further offers the inherent advantage of correcting undesirable source motions and transforming them into natural ones.

Our results indicate that our system efficiently retargets human motion based on our well-designed neural network. In the next phase of the study, we evaluated our synthesized motion results using different networks trained on the subset of the CMU motion capture database. To validate our neural network, we used the same motion dataset to train four different neural networks with different bone length ratios, which forms the main parameter of the network (see Figure 3 and 4). We trained all networks using a Nvidia 1080Ti GPU and training our full network takes about 6 hours. For a fair comparison, we performed the same latent variable optimization step as described in (Section. 4) for all network structures.

In the case of DC-IGN, we attempted to make the network learn disentangled representations of independent skeleton body parts. We quantitatively compared the degree of disentanglements learnt by different numbers of hidden variable $\mathbf{z}$; however, the resulting motions were highly dependent on the number of hidden variables. Results not retaining bone length ratios were obtained when the number of hidden variables was more than a 100 dimensions. Further, when we reduced the number of hidden variables, the motion detail was lost. The errors of the validation motion data from the DC-IGN were relatively higher than those of other neural networks; DC-IGN hardly preserved the user-specified bone length ratio during kinematic optimization in the prediction phase. On the other hand, it usually maintained the details and nuances of the original motion to be retargeted.

For the CNN, we concatenated the hidden variable $\mathbf{z}$ with the vector that is flattened from two-dimensional matrix (30×256) of last layer of the encoder of CNN in the middle of the network. Subsequently, the data pass through a dense layer with dimension = 7685. When using the CNN for retargeting the motion, the errors were relatively small, and it could successfully retain the bone length ratios during kinematic optimization. However, the motion detail was degraded.

In the case of FNN+CNN, the input (240×69) was flattened and sent to the dense layers. The dimensions of each dense layer were 1000, 500, and 100, and the user input was concatenated with $\mathbf{z}$ to obtain $\mathbf{z}^* \in \mathbb{R}^{105}$. Next, the number of dimensions was converted into 7680 and reshaped as (30×256). The output of this layer was sent to the decoder, which is similar to the one in the CNN. From the results, we found that the FNN+CNN exhibits relatively small training errors; however, the output is likely to be overfitted as the number of weight values to be trained for the network is relatively large. Further, the detail of motion was lost when we attempted to retarget the dynamic motion instead of static motion. The bone length ratio was efficiently retained during kinematic optimization.

In general, the character having longer legs tends to walk with larger step width and its upper body occasionally bends forward to the ground. However, the numerical optimization-based motion retargeting methods hardly express those motion style and the resulting motion is likely to follow the style of the source motion even for lengthening the legs, torso, and arms. To remedy this problem, we replace the target motion data with new ones to simultaneously achieve motion style transfer. For establishing new target motions, the incremental angles are added to the torso joints of the target motion and those angle values that gradually decrease from the hip joint to the neck are determined to be proportional to the length of the legs and then we retrained our network with new motion data. If the user increases the leg length ratio parameters, the stylistic retargeted motion whose upper body bend forward to the ground is then continuously produced (see Figure 6 (B)).

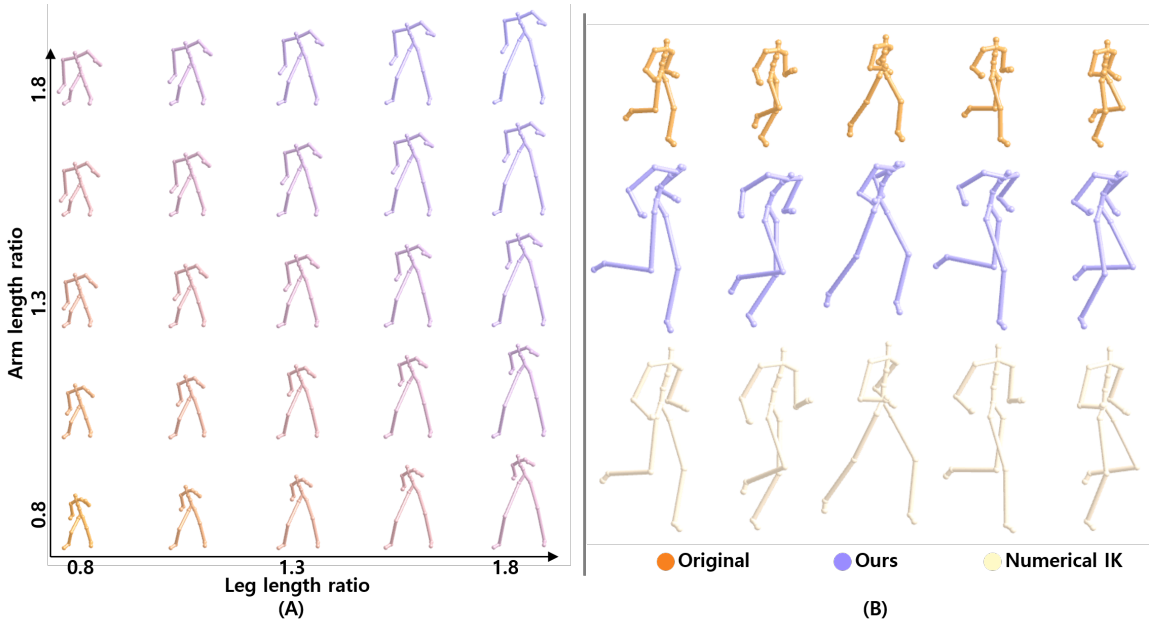


FIGURE 6 Manipulation of bone length variable. The results show the capability of our proposed network to retarget motion data onto a new character. The latent variable arm (z_1) and leg (z_2) change from 0.8 to 1.8. The latent variables for the body parts are successfully disentangled (A). When compared to the conventional IK-based retargeting method[11], ours produces the stylistic motion, and its upper body joints are realistically adjustable in terms of the bone length ratio (B).

Our network combines the DC-IGN and U-Net, and we add skip connections to the network without using the first convolution layer to prevent its negative influence on the prediction capability resulting from the outcome of the first layer. Similar to the DC-IGN, the number of hidden variables affects the quality of the output motion. In our experiments, the optimal number of hidden variables was eight. Figure 7 shows several examples taken from the CMU motion dataset. We note that our implementation outperforms other neural networks in terms of walking, running, dancing, and playing soccer motions ranging from static to dynamic. In contrast to the other networks that work well only for the training data, our network has the capability to retarget motion even for new motion data that do not exist in the training dataset.

6 | CONCLUSION

We have presented a novel motion retargeting system using a variational deep autoencoder combining the DC-IGN and the U-Net for disentangling motion of each body part and preserving details of the original motion. Learning a disentangled representation is an important problem in artificial intelligence, and computer graphics/vision communities. Our system provides the user to smoothly and continuously retarget the human motion based on adjusting the bone length ratios (see Figure 6 (A)). We conducted various cases of experiments for different networks and verified how accurate ours is. To the extent of our knowledge, multi-character motion retargeting using deep learning frameworks have not been previously explored. Our approach is highly robust compared with other motion retargeting frameworks, and it is also able to fix the occasional corrupted motion. We built a versatile motion retargeting framework that automatically enables motion style transfer. Ours offers user-friendly interfaces to produce more realistic human motion, and can be further developed for the scanned 3D face and body data in the near future.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (NRF-2019R1F1A1063467). It was also supported by NCSOFT.

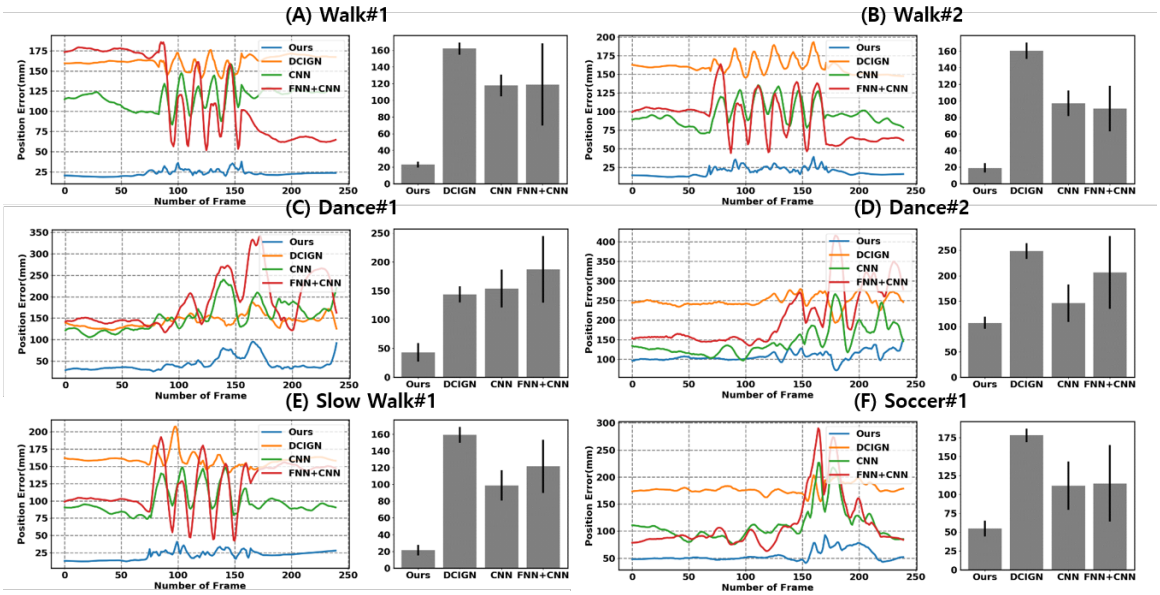


FIGURE 7 Quantitative performances of three different neural networks relative to ours, as described in Figure 3. Our network reliably predicts the retargeted motion from the test motion data set. Various types of motions including running, walking, dancing, etc., are generated, and our network exhibits the lowest prediction error. All errors are measured as the difference of joint positions using the root mean square (RMS) values.

References

1. Gleicher M. Retargetting motion to new characters. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques. ACM; 1998. p. 33–42.

2. Lee J, and Shin SY. A hierarchical approach to interactive motion editing for human-like figures. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques. ACM; 1999. p. 39–48.

3. Grochow K, Martin SL, Hertzmann A, and Popović Z. Style-based Inverse Kinematics. ACM Trans Graph. 2004 Aug;**23**(3):522–531.

4. Huang J, Wang Q, Fratarcangeli M, Yan K, and Pelachaud C. Multi-Variate Gaussian-Based Inverse Kinematics. In: Computer Graphics Forum. vol. 36. Wiley Online Library; 2017. p. 418–428.

5. Fragkiadaki K, Levine S, Felsen P, and Malik J. Recurrent Network Models for Human Dynamics. In: The IEEE International Conference on Computer Vision (ICCV); 2015. .

6. Holden D, Saito J, and Komura T. A Deep Learning Framework for Character Motion Synthesis and Editing. ACM Trans Graph. 2016 Jul;**35**(4):138:1–138:11.

7. Cui Q, Sun H, Li Y, and Kong Y. A deep bi-directional attention network for human motion recovery. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. AAAI Press; 2019. p. 701–707.

8. Kulkarni TD, Whitney WF, Kohli P, and Tenenbaum J. Deep convolutional inverse graphics network. In: Advances in neural information processing systems; 2015. p. 2539–2547.

9. Long J, Shelhamer E, and Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 3431–3440.

10. Choi KJ, and Ko HS. Online motion retargetting. The Journal of Visualization and Computer Animation. 2000;**11**(5):223–235.

11. Buss SR. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. IEEE Journal of Robotics and Automation. 2004;**17**(1-19):16.

12. Aristidou A, Lasenby J, Chrysanthou Y, and Shamir A; Wiley Online Library. Inverse Kinematics Techniques in Computer Graphics: A Survey. Computer Graphics Forum. 2018;**37**(6):35–58.

13. Taylor GW, Hinton GE, and Roweis ST. Modeling Human Motion Using Binary Latent Variables. In: Schölkopf B, Platt JC, and Hoffman T, editors. Advances in Neural Information Processing Systems 19. MIT Press; 2007. p. 1345–1352.

14. Taylor GW, and Hinton GE. Factored Conditional Restricted Boltzmann Machines for Modeling Motion Style. In: Proceedings of the 26th Annual International Conference on Machine Learning. ICML '09. New York, NY, USA: ACM; 2009. p. 1025–1032.

15. Du Y, Wang W, and Wang L. Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015. .

16. Holden D, Komura T, and Saito J. Phase-functioned neural networks for character control. ACM Transactions on Graphics (TOG). 2017;**36**(4):42.

17. Tang Y, Ma L, Liu W, and Zheng WS. Long-Term Human Motion Prediction by Modeling Motion Context and Enhancing Motion Dynamic. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. AAAI Press; 2018. p. 935–941.

18. Villegas R, Yang J, Ceylan D, and Lee H. Neural kinematic networks for unsupervised motion retargetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 8639–8648.

19. Lim J, Chang HJ, and Choi JY. PMnet: Learning of Disentangled Pose and Movement for Unsupervised Motion Retargeting. In: British Machine Vision Conference (BMVC); 2019. .

20. CMU. Carnegie-Mellon Motion Capture Database; 2013. http://mocap.cs.cmu.edu/.

21. Sorkine O. Least-squares rigid motion using svd. Technical notes. 2009;**120**(3):52.