Deep Motion Transfer without Big Data

Byungjun Kwon
Creative AI Team, NCSOFT
South Korea
requiem4546@ncsoft.com

Moonwon Yu
Creative AI Team, NCSOFT
South Korea
moonwonyu@ncsoft.com

Hanyoung Jang*
Creative AI Team, NCSOFT
South Korea
ncdrjang@ncsoft.com

KyuHyun Cho
Project A2, NCSOFT
South Korea
keicoon@ncsoft.com

Hyundong Lee
Project A2, NCSOFT
South Korea
mansoul@ncsoft.com

Taesung Hahn
Project A2, NCSOFT
South Korea
spinel@ncsoft.com

Figure 1: Synthesized motions from the motions of being attacked/turning/running for a human and an idle motion for Krall.

ABSTRACT

This paper presents a novel motion transfer algorithm that copies content motion into a specific style character. The input consists of two motions. One is a content motion such as walking or running, and the other is movement style such as zombie or Krall. The algorithm automatically generates the synthesized motion such as walking zombie, walking Krall, running zombie, or running Krall. In order to obtain natural results, the method adopts the generative power of deep neural networks. Compared to previous neural approaches, the proposed algorithm shows better quality, runs extremely fast, does not require big data, and supports user-controllable style weights.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Motion processing**; **Procedural animation**;

KEYWORDS

Deep Neural Networks, Character Animation Synthesis

ACM Reference Format:
Byungjun Kwon, Moonwon Yu, Hanyoung Jang, KyuHyun Cho, Hyundong Lee, and Taesung Hahn. 2018. Deep Motion Transfer without Big Data. In *Proceedings of SIGGRAPH '18 Posters*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3230744.3230751

*Corresponding author

SIGGRAPH '18 Posters, August 12-16, 2018, Vancouver, BC, Canada

ACM ISBN 978-1-4503-5817-0/18/08.
https://doi.org/10.1145/3230744.3230751

1 INTRODUCTION

In modern games, especially massively multiplayer online role-playing games (MMORPGs), characters appear with different races, professions, genders, and more. Creating all the motions of each character is very hard work because motion capture and manual processes of reducing captured data to *motion sets* consisting of many motion clips, including locomotion for seamless animations and transitions in games, are very tedious and time-consuming.

Motion retargeting approaches are often adopted to reduce the cost. However, these approaches require a manual process for mapping between different characters, or depend on complex physics simulations. Recently, deep neural networks have shown generalizability and generative power in image processing. Gatys [Gatys et al. 2016] synthesized high-quality images from photographs and artworks (e.g., Starry Night by Vincent van Gogh). Holden [Holden et al. 2016] applied that algorithm to the motion data of 3D characters in order to transfer a motion's *content* (e.g., walking) to another character while preserving the character's own *style* (e.g., zombie, depressed, old man, or injured). This work demonstrates the possibility of high-quality motion transfer without manual mapping and physics simulations. However, it is impractical in game production because training deep neural networks requires a huge amount of motion capture data.

In this paper, we present a motion transfer algorithm that generates synthesized animations while imitating content motions and preserving styles. The algorithm requires only a few seconds of content and style motions, enabling game and movie companies

to use their own motion capture data. The proposed method supports a user-controllable style weight for each body part in order to produce the desired result. It is also sufficiently faster than previous approaches to perform interactively. In our experiments, the proposed method was able to transfer various content motions to many types of characters.

2 OUR APPROACH

The input to the algorithm consists of a content motion, a style motion, and style weights. Both motions have the same frame length, and each frame is defined using quaternions. In our experiments, 120 frames and 25 quaternions were used. Therefore, the input matrix for a given motion has dimensions of $(120 \times 25 \times 4)$. Quaternion does not fit well into the neural network framework because of ambiguity problem. To solve this problem, we replaced discontinuous quaternion q with $-q$, and ensured that each component value of the quaternion changes smoothly.

Each motion is passed to convolutional neural networks (*CNNs*), and then encoded to a *feature vector*. Our CNNs consist of 6 subparallel CNNs, and each of which corresponds to a part of the character's body. The output of the subparallel CNNs has dimensions of (120×1024). Unlike in previous approaches, our CNNs were initialized using uniform random values, and used as is. This is because random values have more generative power [He et al. 2016], which big data does not need.

Using the content and style feature vectors, content and style losses are computed in the feature space. Content loss is proportional to the Euclidean distance in the feature space. Style loss is computed using the Gram matrix [Gatys et al. 2016] and amplified by the style weights. Using these losses, the *target feature vector*, which minimizes the sum of content loss and weighted style loss, can be computed.

Finally, the synthesized motion is decoded from the target feature vector using the CNNs that we used for encoding. Because one-to-one correspondence between a motion and a feature vector is not guaranteed, the back-propagation algorithm is performed repeatedly until the feature vector of the synthesized motion becomes equal to the target feature vector, completing the decoding process [Gatys et al. 2016].

3 IMPLEMENTATION AND FUTURE WORK

We implemented our algorithm using Python and TensorFlow, and visualized it using Unreal Engine 4. All motion data was obtained using commercial motion capture systems, and about 10000 frames (6 minutes) were used for testing the algorithm. We believe that the proposed algorithm is the first neural motion transfer method that overcomes the lack of data problem, which causes visual artifacts in existing methods. To generate n synthesized frames, our method requires only $2n$ input frames. Moreover, we achieved a speed more than $10X$ faster than previous methods; Our method spent 2.5 seconds, while the recent method [Holden et al. 2016] took 25 seconds on a NVIDIA GTX 1080. Table 1 shows the comparison.

A limitation of this method is that the result can be physically incorrect (e.g., foot skating and self-collision). Currently, we are integrating an inverse kinematics method with our neural network algorithm to solve this. Another limitation is that the style tends to depend on the range of bone movement; however, some styles appear with animation timing (e.g., lame walkers). Therefore, a user-controlled parameter for timing should be considered to improve the algorithm.

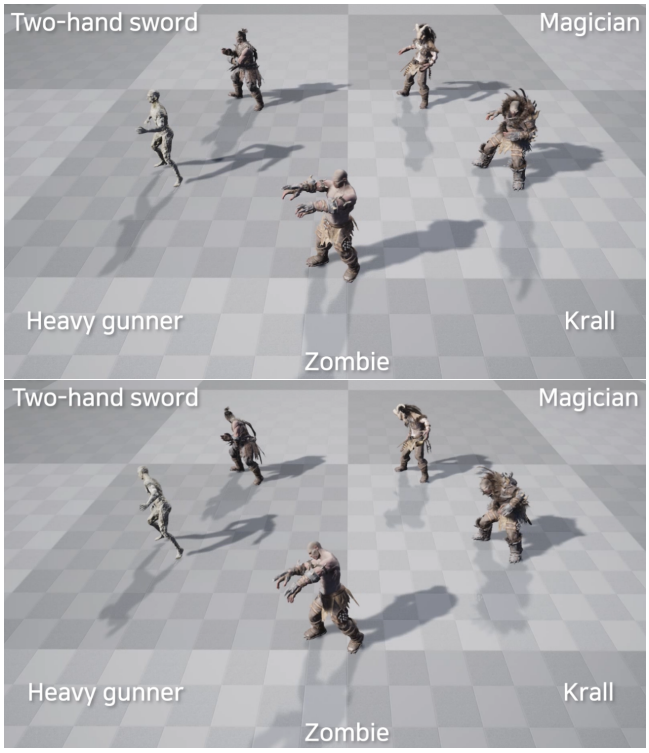

Figure 2: Transferring an attacked content motion to 5 different characters: top(rest), bottom(attacked).

Table 1: Complexity comparison

	Holden *et al.* '2016	Our method
Data size (# of frames)	Millions	Hundreds (240 per execution)
Networks training (times)	Hours	None
Motion transfer (times)	Tens of seconds (25 seconds)	A few seconds (2.5 seconds)

REFERENCES

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2414–2423.

Kun He, Yan Wang, and John E. Hopcroft. 2016. A Powerful Generative Model using Random Weights for the Deep Image Representation. In *Advances in Neural Information Processing Systems 29 (NIPS)*. 631–639.

Daniel Holden, Jun Saito, and Taku Komura. 2016. A Deep Learning Framework for Character Motion Synthesis and Editing. *ACM Trans. Graph.* 35 (2016), 138:1–138:11.