Presenting a Deep Motion Blending Approach for Simulating Natural Reach Motions

Felix Gaisbauer[1,2], Philipp Fröhlich[1], Jannes Lehwald[1], Philipp Agethen [1,2] and Enrico Rukzio[2]

[1]Daimler AG, Wilhelm-Runge-Str. 11, 89081 Ulm, Germany
[2]Ulm University, James-Franck-Ring, 89081 Ulm, Germany

Abstract

Motion blending and character animation systems are widely used in different domains such as gaming or simulation within production industries. Most of the established approaches are based on motion blending techniques. These approaches provide natural motions within common scenarios while inducing low computational costs. However, with increasing amount of influence parameters and constraints such as collision-avoidance, they increasingly fail or require a vast amount of time to meet these requirements. With ongoing progress in artificial intelligence and neural networks, recent works present deep learning based approaches for motion synthesis, which offer great potential for modeling natural motions, while considering heterogeneous influence factors. In this paper, we propose a novel deep blending approach to simulate non-cyclical natural reach motions based on an extension of phase functioned deep neural networks.

CCS Concepts

•***Computing methodologies*** → *Collision detection; Simulation types and techniques; Animation;*

1. Introduction

Nowadays, character animation systems are an essential aspect of different domains such as gaming or production industries. The predominant proportion of the applied approaches are based on motion blending techniques and use specialized motion controllers and blending trees. These methods provide natural motions while being computationally efficient. In general, these approaches can be subdivided into Barycentric-, K-Nearest-Neighbor- and Radial Basis Function-interpolation [FHKS12]. Moreover, inverse blending approaches and inverse kinematics are frequently utilized for solving various constraints. However, with increasing amount of parameters and restrictions, these systems increasingly tend to fail or require a vast amount of time to satisfy all desired constraints.

Due to the growing computational capabilities and ongoing development of software for machine learning, recently, deep learning approaches received significant attention in various domains, including character animation. Recent works present deep learning based approaches [HKS17] [LZX*17] which offer great potential for modeling natural motions. On the one side, naturalness can be ensured through data driven training, whereas on the other side various influence parameters can be considered by these approaches. We extend the recent work of holden et al. [HKS17], in which the authors introduced phase functioned neural networks to realize natural walking motion synthesis, to the domain of modeling reach motions. In this context, we present a deep neural network (DNN) which synthesizes human reach motions depending on heterogeneous influence parameters such as the person's height or velocity, thus providing a flexible and powerful alternative to common motion blending systems.

2. Deep Blending Approach

As shown in Figure 1, the basic idea of the proposed approach is to generate natural human reach motions based on a series of DNNs. Whereas traditional motion interpolation techniques may fail to incorporate various influence factors such as the person's height or velocity, these parameters can be directly considered by the developed system.

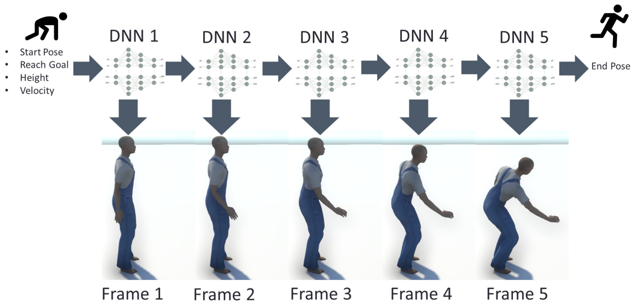

Figure 1: *Overview of the proposed approach. A series of DNNs is sequentially evaluated to compute natural reach motions.*

DOI: 10.2312/egp.20181010

2.1. Phase Functioned Neural Network

The proposed approach extends the concept of phase functioned neural networks, being first introduced by Holden et al. [HKS17]. The idea of phase functioned neural networks combines several distinct DNNs by using a so called phase function, whereas each DNN computes the next pose at a given phase. As shown by the authors, the concept has been successfully applied to the domain of walking while considering environmental constraints. Since the human walking behavior can be subdivided into cyclical walk-phases, the walking motions have been mapped to a phase in time. Each DNN estimates the pose of the avatar at the next phase considering the current avatar pose and constraints. Whereas, the proposed approach only has been applied for cyclical walk motions, we extend the basic idea to non-cyclical reach motions while considering input parameters like the person's height, velocity and reach goal.

2.2. Concept & Network Architecture

Since reach motions have defined start- and endpoints, analogously to the phases introduced in [HKS17], a motion can be subdivided into a discrete amount of static poses laying on the corresponding motion trajectory. Within this work, the motions are independently subdivided into 10 distinct frames (f_0 - f_9), equidistantly aligned in time. Each frame is represented by a trained DNN and its corresponding weights. The input of the respective DNN is a 100-dimensional vector. This vector represents the pose of the character (91 dimensions) at the respective time of f_i, containing the root position and all relative rotations of the character joints. Additionally, the height of the character, the current velocity, as well as the reach goal (position and rotation) are stored for each f_i. In general, the vector is used as input for the DNN at frame f_{i+1}. The output of the network at f_{i+1} is the character pose of frame f_{i+1}.

Based on these distinct neural networks, a motion can be generated by sequentially evaluating this chain (see Figure 1). Initially, the starting pose of the character, reach goal, person's height and velocity are set as input to the DNN at frame f_0. The output of the network is used as input for the subsequent network, whereas the reach goal, person's height and velocity are further added. This process is repeated until the last frame is reached. The results of this process are 10 static poses of the digital avatar. To generate a smooth motion out of this discrete poses, a spline interpolation is applied using cubic B-splines. Since the 10 utilized frames produced natural looking results with spline interpolation of the pose, the weights of the network were not further interpolated.

All DNNs are arranged with identical architectures using the Keras python package [Co15]. The networks have 100 input values, 2 hidden layers and 91 output values. The two hidden layers contain 800 and 600 neurons, while dropout with value of 0.5 is used between each layer. As activation functions for layer 1-3 Exponential Linear Unit (ELU) [CUH15] was chosen, whereas the last layer utilizes a linear activation function.

2.3. Training & Results

Each of the DNNs has been individually trained with respective input and output vectors. The data was normalized using the mean and standard deviation. As training data, 600 000 reach motions have been artificially generated using inverse kinematics and self collision avoidance, while using a Rapidly Exploring Random Tree algorithm for path planning. Each motion was subdivided into 10 equidistantly aligned frames. Thereby, the target position and rotation of the hand have been randomly sampled. The reaching velocity has been randomly sampled within [0.80; 2.00] m/s, whereas the person's height has been adjusted ranging from 1.50 m to 2.00 m in steps of 0.10 m. The network consisting of 10 distinct DNNs has been trained in total for 12 hours in Keras with tensorflow GPU back-end. For optimization, the stochastic gradient descent (SGD) optimizer has been used for 100 epochs (batch size 128).

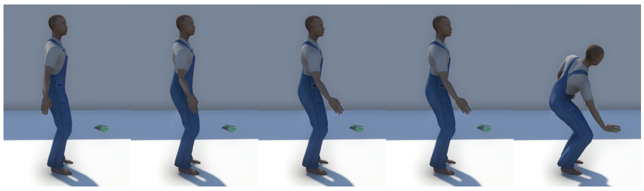

Figure 2: *Image series of a reach motion computed by the DNNs.*

After training, the network was able to generate natural looking motions based on varying influence parameters (see Figure 2). Even though being not explicitly trained, the network is able to handle heights in between the taught values and successfully interpolate the data. Moreover, the network is able to extrapolate natural looking motions in terms of velocity. Occasionally, at reach points near the border of the reaching space, the feet loose contact to the ground, which can be further post-processed using inverse kinematics. Overall the accuracy of the network has been evaluated using a validation set of 60000 motions. The final mean mean squared error of the overall output vector can be specified with 0.0002.

3. Conclusion & Outlook

This paper presents an approach to model reach motions with heterogeneous influence factors utilizing a set of deep neural networks. First validations show, that the approach produces natural motions while being able to interpolate between the trained values. In future work, the network will be trained with a large database of motion capture clips, whereas collision avoidance and object interaction will be further considered.

References

[Co15] CHOLLET F., OTHERS: *Keras*. GitHub, 2015. 2

[CUH15] CLEVERT D.-A., UNTERTHINER T., HOCHREITER S.: Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289* (2015). 2

[FHKS12] FENG A., HUANG Y., KALLMANN M., SHAPIRO A.: An analysis of motion blending techniques. In *International Conference on Motion in Games* (2012), Springer, pp. 232–243. 1

[HKS17] HOLDEN D., KOMURA T., SAITO J.: Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 42. 1, 2

[LZX*17] LI Z., ZHOU Y., XIAO S., HE C., LI H.: Auto-Conditioned LSTM Network for Extended Complex Human Motion Synthesis. *arXiv preprint arXiv:1707.05363* (2017). 1