



INDIVIDUAL ASSIGNMENT

RESEARCH METHODS FOR COMPUTING AND TECHNOLOGY

Student Name	Cheng Yi Heng
TP Number	TP058994
Intake Code	APU2F2305CGD
Module Code	CT09832RMCT
Lecturer Name	Assoc. Prof. Ts. Dr. Tan Chin Ike
Hand Out Date	7 th November 2023
Hand In Date	19 th January 2024

Exploring Deep Learning Approaches for Real-Time Interactive Character Animation

Yi Heng Cheng

voxell.technologies@gmail.com

Abstract— Deep learning, in particular, neural networks has proven to be capable of solving a wide variety of complex tasks. Multiple research on deep learning based approach for character animation has been proposed to utilize the enormous learning capabilities of neural networks for generating dynamic animation for real-time interactive applications like games. This research explores a wide variety of deep learning approaches towards interactive character animation. A total of 3 different methods (motion matching based, reinforcement learning, and pose generation) was investigated based on their strengths, limitations, and novel contributions. A comparison were also performed based on their evaluation results. Throughout the research, we found that the use of deep learning can drastically improve the quality of runtime animation in dynamic settings. Neural networks are capable of adapting to unseen data, filling in the gaps where needed, and even react to the physical environment accurately.

Index Terms

Character Interactions, Locomotion, Neural Networks, Motion Matching

1. Introduction

Interactive character animations are typically carried out through skeletal motions of an articulated figure. This is achieved using a technique called skinning, which deforms the surface of the character based on bone transformations, particularly, the position, orientation, and sometimes the scale of the bones (Rumman & Fratarcangeli,

2016). Animation software like Blender allow animators to author animations using keyframes. Each keyframe stores a snapshot of a character pose which consists of multiple bone transforms. When an animation is being played during runtime, the software interpolates between these keyframes, producing a fluid motion.

Relying solely on manual animation authoring can be extremely inefficient. Motion capture was widely used during the process of animation authoring. Motion capture generates animations by tracking and recording moving objects in the physical space (Menolotto et al., 2020). The raw data from motion capture will then be cleaned and refined by animators before it is being used in production. In addition, inverse kinematics can also be used to generate runtime animation layering such as orienting the head towards an interest point or positioning the hands on an object correctly (Rose III et al., 2001).

Runtime usage of animations is normally done using some form of state machine. Developers were tasked to manually map a diverse set of animations to their target states. These states are then connected usually in the form of graph, allowing the character to react accordingly to different scenarios that may happen during runtime by traversing the graph. This method does not adapt well to development changes, adding a new state might result in significant modifications on the existing state graph, making it hard to maintain. To address this problem, motion matching was proposed by Büttner & Clavet (2015), which opens the possibility of using

unstructured animation data. This method performs a search from a large database to find an animation sequence that best fit the current context, namely, the current pose and the current trajectory of the character (Clavet, 2016).

In recent years, deep learning techniques such as the Transformer neural network has proven to be capable of learning large unstructured data and construct pattern to perform complex human level tasks such as language translation (Vaswani et al., 2017). Deep learning, which is a subset of machine learning, enables developers to overcome the traditional approach of having to solve everything manually. Instead, it learns from the data provided in a goal-centric manner, which maps nicely to the animation problem. In this paper, we will examine the techniques available and provide some insights based on the discoveries.

2. Problem Statement

Creating decent looking character animation system for complex behaviors and scenarios often require very large and manually authored animation sequences (Harvey et al., 2020). Motion capture can be used to generate animations and motion matching can be a great candidate for replacing the traditional data-driven graph traversal method.

However, these new techniques come with a cost. The use of motion capture introduces the need to clean up animations. Due to the heavy reliance of the animation dataset, motion matching is unable to perform seamless transitions if the data is lacking. This results in the use of large animation database, filling up the memory during runtime and a generally more computationally expensive animation lookup (Holden et al., 2020). Additionally, of all the techniques above, in between transitions of different animation sequences still relies heavily on simple blending. This

means that if no sample data are available on how 2 different animation states blends, the resulting transition might look unnatural. Finally, these animation systems lack the capability of reacting to the physical world accurately such as losing balance when tripping over an obstacle (Peng et al., 2018; 2022).

3. Research Aims

The aim of this research is to explore the potential of deep learning techniques for producing fluid character animation that can react naturally to dynamic and unpredictable factors such as terrain changes and user interactions.

4. Research Objectives

1. Evaluate the strengths and weaknesses of different deep learning approaches in character animation.
2. Comparison of different deep learning techniques for character animation.
3. Exploring ways for incorporating these techniques into real-time application development.

5. Research Questions

1. How does deep learning contribute to the enhancement of character animation in interactive environments?
2. What are the types of deep learning techniques for character animation?
3. What is the impact of deep learning in real-time interactive character animation industry?

6. Research Significance

Fluid and natural looking animation is crucial for the immersion of real-time interactive applications. Performing such task can be very tedious and costly. Recent findings show that deep learning offers a way to streamline these processes by learning from large unstructured animation dataset

and “compresses” them for limited runtime inferencing budget (Holden et al., 2020; Peng et al., 2018; Peng et al., 2022; Starke et al., 2020; Starke et al., 2022). Because of the learning nature of neural networks, it can also generate animation for unseen circumstances, solving the memory issue that motion matching poses (Holden et al., 2020). Overcoming these issues will allow the interactive industry to be one step ahead in achieving a more immersive experience.

7. Conclusion

Real-time interactive character animation contributes largely into the immersion of interactive applications such as games. Constructing such system that can react realistically and naturally to dynamic environments is extremely challenging without incorporating machine learning components. Thanks to the adoption of neural networks and the abundance of accelerated computing of GPUs in recent years, the animation industry has been able to benefit from it by harnessing the enormous learning capability of neural networks to revolutionize the interactive application industry.

References

- Büttner, M., & Clavet, S. (2015). Motion matching-the road to next gen animation. *Proc. Of Nucl. Ai*, 2015, 2.
- Clavet, S. (2016). Motion matching and the road to next-gen animation. *Proc. Of GDC*, 4, 9.
- Harvey, F. G., Yurick, M., Nowrouzezahrai, D., & Pal, C. (2020). Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 4, 60–1.
- Holden, D., Kanoun, O., Perepichka, M., & Popa, T. (2020). Learned motion matching. *ACM Transactions on Graphics (TOG)*, 4, 53–1.
- Menolotto, M., Komaris, D.-S., Tedesco, S., O’Flynn, B., & Walsh, M. (2020). Motion capture technology in industrial applications: A systematic review. *Sensors*, 19, 5687.
- Peng, X. B., Abbeel, P., Levine, S., & Panne, M. Van de. (2018). Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 4, 1–14.
- Peng, X. B., Guo, Y., Halper, L., Levine, S., & Fidler, S. (2022). Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions on Graphics (TOG)*, 4, 1–17.
- Rose III, C. F., Sloan, P.-P. J., & Cohen, M. F. (2001). Artist-directed inverse-kinematics using radial basis function interpolation. *Computer Graphics Forum*, 3, 239–250.
- Rumman, N. A., & Fratacangeli, M. (2016). State of the art in skinning techniques for articulated deformable characters. *International Conference on Computer Graphics Theory and Applications*, 200–212.
- Starke, S., Mason, I., & Komura, T. (2022). DeepPhase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 4, 1–13.
- Starke, S., Zhao, Y., Komura, T., & Zaman, K. (2020). Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG)*, 4, 54–1.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.