



### **INDIVIDUAL ASSIGNMENT**

#### **RESEARCH METHODS FOR COMPUTING AND TECHNOLOGY**

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

# Exploring Deep Learning Approaches for Real-Time Interactive Character Animation

Yi Heng Cheng

voxell.technologies@gmail.com

**Abstract—** Xxx

## **Index Terms**

Character Animation, Deep Learning, Neural Networks, Interactive, Real-time

## **1. Introduction**

## **2. Literature Review**

### **2.1. Interactive Character Animation**

Interactive character animation is made up of multiple underlying technologies. At its core, it consists of the presentation component and the state management component. The end goal of the presentation component is to provide a visual feedback of the animation towards the user. The presentation component heavily depends on the state management component which decides what pose to be displayed at a given point in time. Managing a state machine well can be a day and night difference between having a good or bad animation response for a given scenario.

#### **2.1.1. Mesh Skinning**

Skinning is the process of performing mesh deformation through the movement of several skeletons. It plays a major role in the presentation component.

Linear Blend Skinning (LBS) is a commonly used technique in character animation where each vertex of the character mesh is influenced by a weighted sum of the transformations of nearby bones (Lander, 1998). It is being used in AAA game engines like Unity3D and Unreal Engine. LBS is known for its fast and simple algorithm

that maps advantageously to the graphics hardware.

Spherical Blend Skinning (SBS) is another form of skinning technique that employs spherical interpolation to smoothly blend between bone transformations (Kavan & Žára, 2005). SBS aims to solve the “lost of volume” artifact that LBS brings despite its efficient algorithm.

To solve the computational and memory overhead that SBS brings, Kavan et al. (2007) propose Dual Quaternion Blending (DQB). DQB uses dual quaternions to represent both translation and rotation, allowing for more accurate and natural deformations of the character mesh. Unlike SBS, it does not require additional memory to cache rotation centers. The DQB technique is also extremely efficient. However, DBQ comes with a limitation, it only supports rigid transformation and is not suitable for scaling or shearing effects.

#### **2.1.2. Inverse Kinematics**

Inverse kinematics (IK) is widely used in video games and robotics to create realistic poses within a defined constraint. In short, the ultimate goal of IK is to determine an appropriate joint configuration that allow the end effectors to reach a target position (Aristidou et al., 2018).

One use case of IK is to perform animation retargeting to map movements between characters with different proportions (Molla et al., 2017). In the context of interactive applications like games, IK can also be used to perform secondary motions on top of

an already playing animation (Ruuskanen, 2018). For example, turning the head towards an interest point, or moving the hand towards a target position.

At its current state, there is a total of 4 main categories towards IK:

### 1. Analytical

Analytical IK solvers aim to determine all potential solutions based on mechanism lengths, initial posture, and rotation constraints. They often rely on assumptions to compute a single solution.

### 2. Numerical

Numerical methods often require a set of iterations to achieve a satisfactory approximation by minimizing a predefined cost function.

### 3. Data-Driven

Data-driven methods relies on large accurate animation databases. Most data-driven methods employs some kind of machine learning algorithms to learn from the dataset.

### 4. Hybrid

The hybrid method is simply a way of combining 2 or more different IK methods into a single solution.

#### 2.1.3. Physics Based Animation

Physics based animation offers a completely new solution for developers to prioritize physics accuracy over animation precision. It forces characters to obey the laws of physics like preventing collisions between collidable objects and interacting with external forces such as gravity, pressure, etc (Ye Hu, 2016).

Authoring physics based animation can be extremely hard. This is due to the unpredictability of the physical world. For example, a character might accidentally get hit by a physical object during runtime,

resulting in unexpected movements or behaviors that can disrupt the intended animation sequence.

A major limitation of physics based animation is the inability to precisely control the artistic intent for achieving specific visual effect. Additionally, ensuring computational efficiency while simulating complex physical interactions adds another layer of challenge to the authoring process.

#### 2.1.4. Animation Graph

Animation graphs are typically used to systematically change the playing animation clip based on the current state. In game engines like Unity3D, a state machine graph is used to managed the transition between these states. During the transition, a method known as animation blending is used to interpolate between 2 distinct animation clips.

The simplest and fastest form of animation blending is linear interpolation. However, using only linear interpolation results in unrealistic rigid transitions.

### 2.2. Deep Learning in Animation

#### 2.3. The Hybrid Approach

## 3. Methodology

### 3.1. Target user

### 3.2. Sampling method

### 3.3. Data collection method

## 4. 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.

## 5. References

- Aristidou, A., Lasenby, J., Chrysanthou, Y., & Shamir, A. (2018). Inverse kinematics techniques in computer graphics: A survey. *Computer Graphics Forum*, 6, 35–58.
- 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.
- Kavan, L., & Žára, J. (2005). Spherical blend skinning: a real-time deformation of articulated models. *Proceedings of the 2005 Symposium on Interactive 3d Graphics and Games*, 9–16.
- Kavan, L., Collins, S., Žára, J., & O'Sullivan, C. (2007). Skinning with dual quaternions. *Proceedings of the 2007 Symposium on Interactive 3d Graphics and Games*, 39–46.
- Lander, J. (1998). Skin them bones: Game programming for the web generation. *Game Developer Magazine*, 1, 10–18.
- 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.
- Molla, E., Debarba, H. G., & Boulic, R. (2017). Egocentric mapping of body surface constraints. *IEEE Transactions on Visualization and Computer Graphics*, 7, 2089–2102.
- 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., & Fratarcangeli, M. (2016). State of the art in skinning techniques for articulated deformable characters. *International Conference on Computer Graphics Theory and Applications*, 200–212.
- Ruuskanen, A. (2018). *Inverse Kinematics in Game Character Animation*.
- 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*.
- Ye Hu, L. (2016). Physics-based character animation. *Universitat Politècnica De Catalunya*.