Physics-based character animation for Virtual Reality

Joan Llobera *
Artanim Foundation

Caecilia Charbonnier †
Artanim Foundation

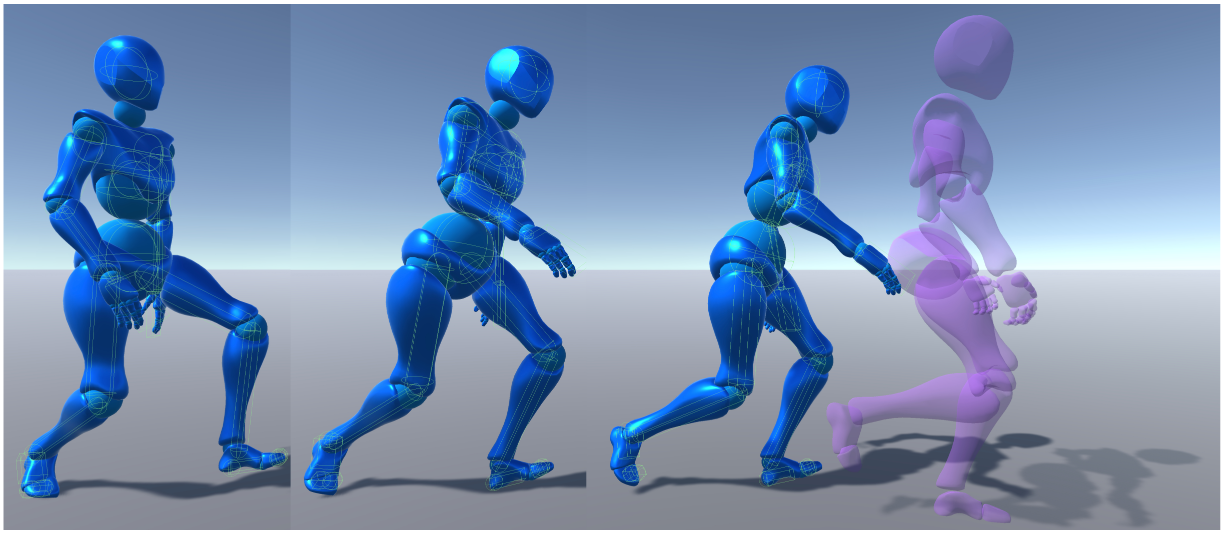

Figure 1: A rigged character mapped to a ragdoll that imitates a reference animation.

Abstract

Virtual Reality (VR) users tend to engage with interactive characters with gestures, social cues and their full body. However, it is still difficult to create VR characters that react appropriately to these cues. Recent progress in physics-based character animation suggests these techniques may offer a way forward. We have developed Marathon, an open source project that enables the use of physics-based interactive characters for VR, as well as support materials to facilitate using it.

1 Motivation

Virtual Reality (VR) experiences tend to make people feel as if they were in the place depicted by the scene rendered [16], and this feeling makes them engage with virtual characters with gestures, social cues and their full body, opposed to pressing buttons in game pads or joysticks as it is characteristic of video games. This offers the potential to use VR as a new medium where interactive characters and users collaborate to unfold a story within an interactive experience [7, 10]. However, the tools available to create interactive virtual characters are generally based on segmenting animation sequences and defining transitions among them in a state machine. These state machines are available as animation engines in what are *de facto* standards of video game production (Unity3D and Unreal Engine). A relatively recent alternative that has seen adoption by the video game industry is the use of *Motion Matching* [2]. This gives more flexibility as to where the transitions can occur, and generates smoother animations. However, these animation systems are best suited for button-based binary input, characteristic of video games. Using these techniques it is very difficult to generate interactive behaviour that responds to continuous data streams such as the position of VR users, where they look at, or the speed at which they move. There are no readily available tools to create a VR character that respects the implicit rules of interpersonal distances, or that will take into account the users orientation and position within a virtual space to synthesise their movements in a way that is easy to see by the VR users. Even less creating tasks that require close coordination between the user and a character, such as moving a table together, or building a tower together. As a result, VR experiences often avoid introducing virtual characters or, when they do, they offer limited or no interaction between users and autonomous characters (see Fig. 2).[1]

A possible strategy to address these challenges is to turn towards physics-based character animation. Here, we review the most relevant contributions in this field, outline our recent efforts to make these accessible to the VR community through an open source project and discuss the main challenges still standing to use physics-based animation for interactive VR characters.

2 Previous Work

Physics-based animation has improved considerably in the last five years, building on top of the improvement in deep reinforcement learning. Peng's Deep Mimic [13] demonstrated that physics-based characters could synthesise movement of the same quality than a reference animation. The authors also showed that introducing a random parameter position at the training phase allowed synthesising motion that integrated a particular value of that parameter. For example, kick an object in an arbitrary position, or throw a ball to a particular destination. The work presented by Wang et al. [6] showed that the method could be extended to integrate several parameters, and crucially to synthesise movements with different duration (for example, synthesising jumps of different heights from a single reference animation). This opens the way to create physics-based characters capable of rich interaction with their environment.

*e-mail: joan.llobera@artanim.ch
†e-mail: caecilia.charbonnier@artanim.ch

[1] https://artanim.ch/project/mayflower-vr/

Figure 2: Snapshot of the Artanim VR production Aboard The Mayflower, which uses virtual characters.

Another limitation of Deep Mimic is that each movement has to be synthesised with a separate policy, which limits the extent to which this approach can scale. To address this challenge distillation, techniques have been used to train a more general network that learns to imitate a large set of specialised controllers [11, 12]. Other strategies involve mixture models to combine different policies [14], different training strategies to generate a richer variety of movements within one policy [15], or a divide and conquer approach based on separating large corpora of motion data in clusters to train separate policies [19].

An alternative strategy to address this challenge has been to combine a kinematic controller with a physics-based controller. Bergamin et al. [1] combined motion matching and a physics controller, and Wang et al. [17] further extended this approach proposing a universal physics-based animation layer, trained with a very large database of humanoid animations and different cinematic controllers. Combing those, the kinematic layer integrates the interactive input of the user, and the physics controller integrates the result within the physical simulation of the scene.

A different challenge of adopting these techniques in production is the long time it requires to train each policy. This can be circumvented with a universal physics-based controller, where the physics controller does not need to be retrained for each new behaviour, but then interaction is limited to what is possible with a kinematic controller. Recent developments in differential physics engines show these results can be radically sped up [3, 18]. However, this requires physics engines completely different from *PhysX*, the one used both by default in Unity3D and Unreal. Holden et al. [4] have recently shown that it is possible to learn a forward model of the world [5] to act as a proxy of a differentiable physics engine. This work suggests a way forward to train physics-based controllers orders of magnitude faster, within popular game engines such as Unreal and Unity3D.

3 The Marathon Project

A significant challenge to try these techniques in VR is that contributors either use proprietary game engines and do not release the code, they either use physics-oriented simulation engines such as DART[2] or Bullet [3] or they create their own physics engine [3, 18].

The Marathon project is an open source solution that offers implementations of physics-based character animation contributions in the popular Unity3D game engine. It is freely available online.[4] It allows the use of rigged characters and can easily be integrated in VR experiences. To simplify user adoption we have also implemented the procedural creation of rag dolls ready to train, starting only from a rigged character and some animations [8], and developed a 1 hour video course based on this tool, for researchers wanting to get started in physics-based animation [9]. Our current efforts are focused on improving the quality of the animations generated and generating physics controllers capable of richer interactive behaviour, to then test its impact within VR experiences.

[2]https://dartsim.github.io/

[3]https://github.com/bulletphysics/

[4]https://joanllobera.github.io/marathon-envs/

References

[1] K. Bergamin, S. Clavet, D. Holden, and J. R. Forbes. Drecon: data-driven responsive control of physics-based characters. *ACM Transactions On Graphics (TOG)*, 38(6):1–11, 2019.

[2] S. Clavet. Motion matching and the road to next-gen animation. In *Proc. of GDC*, 2016.

[3] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem. Brax – a differentiable physics engine for large scale rigid body simulation, 2021.

[4] L. Fussell, K. Bergamin, and D. Holden. Supertrack: Motion tracking for physically simulated characters using supervised learning. *ACM Trans. Graph.*, 40(6), dec 2021. doi: 10.1145/3478513.3480527

[5] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[6] S. Lee, S. Lee, Y. Lee, and J. Lee. Learning a family of motor skills from a single motion clip. *ACM Trans. Graph.*, 40(4), jul 2021. doi: 10.1145/3450626.3459774

[7] J. Llobera, K. J. Blom, and M. Slater. Telling stories within immersive virtual environments. *Leonardo*, 46(5):471–476, 2013. doi: 10.1162/LEON_a_00643

[8] J. Llobera, J. Booth, and C. Charbonnier.

[9] J. Llobera, J. Booth, and C. Charbonnier. New techniques in interactive character animation. In *ACM SIGGRAPH 2021 Courses*, SIGGRAPH '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3450508.3464604

[10] J. Llobera and C. Charbonnier. Interactive characters for virtual reality stories. In *ACM International Conference on Interactive Media Experiences*, IMX '21, p. 322–325. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3452918.3465494

[11] J. Merel, L. Hasenclever, A. Galashov, A. Ahuja, V. Pham, G. Wayne, Y. W. Teh, and N. Heess. Neural probabilistic motor primitives for humanoid control. In *nternational Conference on Learning Representations*, 2019.

[12] J. Merel, S. Tunyasuvunakool, A. Ahuja, Y. Tassa, L. Hasenclever, V. Pham, T. Erez, G. Wayne, and N. Heess. Catch & carry: reusable neural controllers for vision-guided whole-body tasks. *ACM Transactions on Graphics (TOG)*, 39(4):39–1, 2020.

[13] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018.

[14] X. B. Peng, M. Chang, G. Zhang, P. Abbeel, and S. Levine. Mcp: Learning composable hierarchical control with multiplicative compositional policies. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 2019.

[15] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Trans. Graph.*, 40(4), July 2021. doi: 10.1145/3450626.3459670

[16] M. Slater. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3549–3557, 2009. doi: 10.1098/rstb.2009.0138

[17] T. Wang, Y. Guo, M. Shugrina, and S. Fidler. Unicon: Universal neural controller for physics-based character motion, 2020.

[18] K. Werling, D. Omens, J. Lee, I. Exarchos, and C. K. Liu. Fast and feature-complete differentiable physics engine for articulated rigid bodies with contact constraints. In D. A. Shell, M. Toussaint, and M. A. Hsieh, eds., *Robotics: Science and Systems XVII, Virtual Event, July 12-16, 2021*, 2021. doi: 10.15607/RSS.2021.XVII.034

[19] J. Won, D. Gopinath, and J. Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.*, 39(4), jul 2020. doi: 10.1145/3386569.3392381