Solving the challenging problem of synthesizing realistic human movement in real-time, that is fully controllable, and interactive, is a significantly important and interesting issue in video games.

Why physics-based character animations are a Must for computer games

Ben Kenwright+

GAME CONSOLES ARE REACHING NEW HEIGHTS IN computational power and memory capacity, while research advancements in graphics, physics, artificial intelligence, and networking have pushed computer games into boundaries of indescribable realism with breathtaking worlds, and epic action packed scenarios. Hence, it should come as no surprise to know that, the video game industry is now bigger than the film industry in terms of sales.

Passive physics-based motions, such as rigid bodies and clothes, are well understood and can be emulated in real-time using Newtonian principles. While active physics-based motions, such as those created by a human, that require intelligent controlling forces to achieve specific tasks, such as balanced stepping or picking up objects, is significantly more complex to emulate, and is an ongoing area of research in numerous fields (e.g., robotics, computer graphics, and biomechanics) [KDM11, YOV07,K12, H98].

Both passive and active movements need to obey the laws of physics to appear realistic. However, movements that are physically accurate are not always life-like and human looking. Humans possess a vast assortment of muscles that can be used in diverse and complex ways to produce diverse movements, such as, walking, running, and jumping. We must combine human factors with physical factors to produce natural-looking motions (i.e., dual-goals that include realism and control) [PW99, RGB96, WP95].

Animation Actions (Control)

Two main techniques exist for generating computer animations, data-driven key-framed methods, and procedural synthetic approaches. However, the crucial factor is "Control"! Key-framed solutions allow artists to run free with their imaginations to create bizarre, original, and diverse animations, either life-like or non life-like fantasy based. While these low-level artistry controls allow the animator to specifically position and orientate every single element of a character at every moment in time, it is a huge burden, time consuming, and can be difficult for even the most highly skilled animators to create motions that are fluid, life-like, and worthwhile.

This low-level control is highly important in computer-generated movies, however, in games the animator cannot control how a character will react and move at every moment within the scene.

In computer games, it is more desirable to focus on high-level motions, such as foot placement, direction or walk type, and leave the low-level problems, such as individual joint orientations and balancing information to the underlying system. This underlying system is what this paper focuses on.

The game developer/animator should be able to define a set of pre-defined actions, such as standing, walking, and running. We can create these fundamental motions using low-dimensional models. These basic motions can be mixed with rhythmic noise or key-framed data to produce more realism (e.g., specifying emotional content, such as happy, sad, or tired).

If the basic motions (e.g., standing and walking) are generated using procedural physics-based techniques, then if we add disabilities to the character,

Email: +bkenwright@xbdev.net
Title: Why Physics-Based Character Animations are a Must in Computer Games
Date: 04/09/2013
URL: www.xbdev.net

such as leg injuries (i.e., limiting knee strength or movement), the procedural algorithm can compensate to produce motions that accomplish the task (e.g., walking) while possessing the realistic limp that a human would exhibit with the injury. Adjusting specific details, such as leg-stiffness, leg-transition time, and arm positions, a whole variety of diverse and original movements can be created by the animator without needing to specify low-level details.

Generating a diverse range of complex actions, such as kicks, punches, back-flips, and baseball hits, can be difficult without key-framed data. A set of controllers or even a pre-canned library of diverse motions can be time consuming to create.

Synthesizing Character Motions

We as humans can perceive subtle unnatural visual artefacts in human motion effortlessly. An animator must understand the natural mechanics of human movement in an attempt to mimic and synthesize high quality natural animations. One particular challenge arises in the creation of fast moving, dynamic, 3D character actions. The animator must understand, physical principles, biomechanics, and human nature to create the most realistic solution.

We focus on generating fundamental animation motions from scratch that possess key characteristics that are inherent in all shared motions (e.g., balanced stepping). We can accomplish this, by exploiting research from robotics and biomechanics, which have long studied the principles of human movement in an attempt to emulate and identify physical issues.

Elementary motions, such as standing, walking, and running, create motions that are less realistic and posses a level of realism that is light. However, we must stress that these are "base" motions that can be combined with additional features through blended to produce a richer assortment of believability.

Data-driven approaches are popular due to the amount of low-level control that can be achieved. Motions are generated based on pre-canned libraries and are modified through motion transforms to accomplish specific tasks (e.g., inverse kinematics and physics-based adaptation). This is a popular and active approach by researchers in computer animation [HP97, LS99, BW95, G97, LGC94].

Low-Dimensional Approach

A low-dimensional model simplifies the character model down to its essential elements (e.g., foot position and centre-of-mass). Algorithms can then be created that can run in real-time, that are robust and straightforward to fit a specific motion (e.g., walking or jumping). Specific information include, body mass, pose/reach constraints, muscle strength, foot and placement constraints. The motions can be edited and customized by adjusting the specific parameters to produce the final animation sequences.

A low-dimensional model simplifies the model so we can focus on the original motion to produce animations in real-time that are robust and unambiguous. A low-dimensional model allows the developer to focus on high level intuitive movement which is easier to control and adapt (i.e., foot placement, timing, kinematic structure, and dynamic interaction).

A low-dimensional model has the added feature of encapsulating highly dynamic movements, such as jumping and flipping, and can be optimized and controlled much more easily due to the simplicity of the structure (i.e., fewer degrees of freedom and less ambiguity compared to a fully articulated character model).

The adaptive nature of the approach means that as situations arise, for example, stairs and sloping terrain, the character's stepping will automatically compensate. However, compared with purely inverse kinematic methods, a physics-based approach will ensure the model obeys the natural laws (i.e., balance and physical momentum). Furthermore, modifying transition and ground contact durations creates stepping motions with varying characteristics. The procedural model can automatically compensate for obstacles along the path, which is important in dynamic environments, such as games.

The character's physical qualities, such as limb dimensions and mass can be altered at run-time and the algorithm will automatically compensate and generate the final motions without complications or necessary input from an animator.

A low-dimensional model reduces the number of degrees-of-freedom (DOF). While it is the job of the animator/programmer to define how these small number of DOF are mapped back onto the highly articulated character with a much larger number of DOF, the extra DOF introduce ambiguity, that can be exploited to blend in additional motion characteristics (e.g., sad, happy, and curious).

Injured-walking and running sequences can be generated automatically (shortening a leg, making on foot lighter than another, or leg-trajectory durations different). Emulating different muscle properties to effect the final output motions.

The player in a game "feels" the motions are natural and under his control since they are generated based on physical interactive qualities (e.g., push the character he will take realistic life-like steps to remedy the disturbance and remain upright).

Certain objectives take priority in the motion, such as balance and foot placement constraints. Softer looking stepping movements can be created by modifying the floor-impact force, by reducing the leg-stiffness. While the low-dimensional procedural model for generating the underlying motion is minimalistic in its design, the computational cost of converting back towards the high-dimensional model can be expensive and thwart with difficulties due to ambiguities and mixing in various behavioural life-like characteristics while keeping the final motion physically correct and realistic (e.g., using weighted priority-based inverse kinematic techniques). Nevertheless, the final motions can be created in real-time, which is crucial for interactive gaming environments.

Conclusion

This paper has shown one solution based on algorithmic techniques for generating physically accurate biped character animations for interactive gaming environments. The powerful high-level control of this approach is appealing because it does not require a large library of key-framed data to create the fundamental animations (e.g., standing, walking, and running), which is necessary for any virtual avatar character to explore its world.

The problem of editing pre-canned motion libraries so they account for dynamics and engage their environment is challenging. While offline approaches for generating animations, such as in computer-generated movies, can allow an animator the time and low-level control to formulate realistic motions. In computer games, the situation and environment can be ever changing and unpredictable; hence, having the basic character motions generated by means of intelligent procedural physics-based techniques is highly beneficial.

Nevertheless, algorithmic approaches have a number of problems that need to be addressed. We need to ensure the final motions remain realistic and do not appear robotic and unnatural. The development of diverse actions and intelligent algorithms can be time consuming and difficult. We need to be aware of the challenges in retargeting, adapting, and mixing the different motions with varying priority, while preserving realism and user control over the character.

The eventual research in this area focuses on creating intelligent life-like motions that can be "plugged-in" to character solutions without requiring any key-framed data. The approach in this paper, gives the game developer a reusable base animation-library that can produce customizable, interactive, and controllable avatars.

Keywords

Animation, Procedural, Balanced, Motion, Key-Frames, Active, Passive, Dynamic, Low-Dimensional, Video Games, Real-Time

References

[BW95] Bruderlin, A. and Williams, L. Motion signal processing. In Proceedings of SIGGRAPH'95 (Los Angeles, Aug.). Addison Wesley, 1995, 97–104.

[G97] Gleicher, M. Motion editing with spacetime constraints. In Proceedings of the 1997 Symposium on Interactive 3D Graphics, M. Cohen and D. Zeltzer, Eds. (Apr.). ACM SIGGRAPH, 1997, 139–148.

[H98] Hodgins, J. Animating human motion. Scientif. Amer. 278, 3 (Mar. 1998), 64–69.

[HP97] Hodgins, J. and Pollard, N. Adapting simulated behaviors for new characters. In Proceedings of SIGGRAPH'97 (Los Angeles, Aug.). Addison Wesley, 1997, 153–162.

[KDM11] Kenwright, B., Davison, R., and Morgan, G. 2011. Dynamic balancing and walking for real-time 3d characters. In Proceedings of the 4th international conference on Motion in Games, Springer-Verlag, Berlin, Heidelberg, MIG'11, 63–73.

[K12] Kenwright, B. 2012. Responsive biped character stepping: When push comes to shove. In Proceedings of the 2012 International Conference on Cyberworlds, vol. 2012, 151–156.

[LS99] Lee, J. and Shin, S.-Y. A hierarchical approach to interactive motion editing for human-like figures. In Proceedings of SIGGRAPH'99 (Los Angeles, Calif., Aug.). Addison Wesley Longman, 1999, 39–48.

[LGC94] Liu, Z., Gortler, S., and Cohen, M. Hierarchical spacetime control. In Proceedings of SIGGRAPH'94 (Orlando, Fla., July). ACM Press, New York, 1994, 35–42.

[PW99] Popovi´c, Z. and Witkin, A. Physically based motion transformation. In Proceedings of SIGGRAPH'99 (Los Angeles, Aug.). Addison Wesley Longman, 1999, 11–20.

[RGB96] Rose, C., Guenter, B., Bodenheimer, B., and Cohen, M. Efficient generation of motion transitions using spacetime constraints. In Proceedings of SIGGRAPH'96 (New Orleans, Aug.). Addison Wesley, 1996, 147–154.

[WP95] Witkin, A. and Popovi´c, Z. Motion warping. In Proceedings of SIGGRAPH'95 (Los Angeles, Aug.). Addison Wesley, 1995, 105–108.

[YOV07] Yin, K., Loken, K., and Van de Panne, M. 2007. Simbicon: simple biped locomotion control. In ACM SIGGRAPH 2007 papers, ACM, New York, NY, USA, SIGGRAPH '07.