Senior Project Proposal: Interactive Physically-Based Character Animation via Optimization

Nathan Marshak

Advisor: Aline Normoyle

University of Pennsylvania

ABSTRACT

Physically-based character animation frees artists from tedious keyframing work by producing biomechanically accurate motion that fits constraints set by artists and programmers. In current realtime applications, physically-based character animation is typically "passive" - the character itself has no goals and only responds to external forces. Previous papers have succeeded in "active" approaches where the character satisfies constraints set by the programmer. Most previous approaches, however, do not handle complex scenarios, such as a scene with two interacting characters. The goal of this project is to implement Witkin and Kass' paper on an optimization approach to physically-based character animation, and then to extend it to support an interactive demo featuring a complex, dynamic scene. Results will be evaluated based on the robustness of the demo, and possible comparison with video footage and/or motion capture data of humans.

Project Blog: http://na-sr-design.blogspot.com/

1. INTRODUCTION

For years, researchers have been able to animate characters with a physics-based approach. Furthermore, giving such characters high-level goals and constraints is considered a solved problem. However, satisfying constraints more sophisticated than retaining balance is difficult. Doing this in real-time while in a complex, dynamic environment (for instance, a scene with two interacting characters) remains an open problem.

Simulating multiple, interacting characters via a physics-based approach is highly desirable in the graphics community for two main reasons. First, if such an approach is robust enough, it frees content creators from relying on large amounts of motion-captured data and pre-scripted actions, since biomechanically-accurate motion is produced procedurally. Second, it enables a greater degree of interactivity in applications like games – given a complex scene and some user input, it may be possible in the future for animations to appear different every time the user plays the game.

This problem will be tackled by implementing "Spacetime Constraints" [WK88] as a plugin to a game engine. Once the implementation works in real time, a demo involving either multiple characters and/or a complex scene will be identified. This demo will then be implemented using a trial-and-error process.

This project makes the following contributions to physically-based character animation:

- Exploration of what constraints are necessary for multiple, interacting characters.
- Exploration of how these constraints should be changed dynamically.
- Exploration of similar constraints if characters are placed in other complex scenes, should this direction be pursued in addition to / instead of multiple interacting characters.
- An implementation that will most likely work as a plugin to the Unity game engine. Hence, there will be a tool that game developers and researchers could easily integrate into their own projects.

1.1 Design Goals

The target audience for this project is the research community, as well as the games industry. Academics benefit from research in a specific area that is yet to be widely explored. Game developers would benefit from having a tool that works with a popular game engine, since most game engines do not animate characters with the approach proposed in this project. Even without a novel research component, this would be useful since physically-based character animation in most games is limited to ragdolls.

Since this is a real-time project, its application to the animation industry is less obvious. Nonetheless, there has been interest in procedural techniques for crowd simulation and for making content creation less labor-intensive, so this project may have an application to movies.

1.2 Projects Proposed Features and Functionality

The project can be summarized with the following list of features:

- Realtime implementation of an animation system like the one on described in "Spacetime constraints" [WK88]
- An interactive demo that will use the aforementioned realtime implementation
- Characters in the demo must have sophisticated goals (more than just staying balanced or moving).
- There must be either two interacting characters in the demo, or the demo must present a dynamic environment that that facilitates sophisticated goals for the character.

2. RELATED WORK

In the seminal paper "Spacetime Constraints" [WK88], Witkin and Kass reproduce the "Luxo, Jr." jumping animation by modeling the lamp as a series of rigid bodies, setting constraints such as start point, end point, and hurdles, and running an optimization that minimizes power used. Such a method provides a way to achieve physically-based character animation with constraints that can be easily set and understood [SHP04].

Performing a space-time optimization over the human range of motion is an extremely difficult problem. Because of this, physical models of humans that work in a lower state space have been developed. Optimization in lower state spaces has been shown to be tractable [MdLH10], [SHP04].

An alternative to optimization that is less computationally intensive is a controller-based approach, such as the one used by Raibert and Hodgins to animate hopping characters. Forces and torques at joints are adjusted instantaneously to keep a character balanced and moving in a particular direction [RH91]. This technique has been extended to animate human walking in realtime [YLdP07].

Physically-based animation can be combined with motion-capture data in order to allow the user to interact with virtual objects [NWB10] or to allow a user to direct an onscreen character via a device like the Kinect [LZ11].

Simulation of two interacting characters has been done in the context of a fighting game – the characters can switch from motion-captured animations to a physically-based model in order to simulate a character struggling to maintain balance [LCC*07].

Physically-based character animation has been used to animate characters in interacting with a simple environment, such as with frictional forces exerted by rigid bodies, with the help of some user input. For instance, Kim and Pollard animate a starfish that lifts an obstacle out of the way (given some simple user input) [KP11]. This is done in real time. In contrast, Kwatra, Wojtan, et al. perform and offline simulation of character physics within a very complex environment – a fluid simulation [KWC*10].

There has been much research on making characters interact with themselves (e.g. to walk and keep balance), or research on applying physically-based methods to user control of characters. Relatively little research has been done on giving characters more sophisticated goals, such as interacting with each other in a situation like a swordfight or a stunt. The same applies for very complex, dynamic environments. Doing this interactively and in realtime is also a challenge that relatively few have attempted.

3. PROJECT PROPOSAL

3.1 Anticipated Approach

An implementation of "Spacetime Constraints" by Andrew Witkin and Michael Kass, will be written. The implementation will use the AMPL optimization solver (by Alla Safonova).

Then, in order to simulate human motion, the optimization may have to be reduced to a lower state space [MLH01], [SHP04], or combined with motion-captured data [NWB*10].

A demo scene conducive to experimentation with multiple characters / a complex environment will be chosen. This demo must be interactive, so it must at the very least have parameters that can be adjusted. Ideas for such a demo are a swordfight, two characters dribbling a ball, an acrobatic stunt, and game characters that respond to being shot and having limbs damaged.

Finally, characters will be given high-level constraints with respect to each other that the optimization will fulfill. These constraints must be adjusted dynamically with an algorithm that must be designed and implemented. This will be done within the context of one specific demo or test case. Experimentation and comparison to motion capture or video of humans will be required. The demo will probably be iterated upon several times before a satisfactory result is obtained.

3.2 Target Platforms

Unity will most likely be used as the game engine. The AMPL optimization solver will be used. The system will probably be written in C/C++, then bound to C# to work with Unity. This project is intended to run on a modern x86 PC.

3.3 Evaluation Criteria

A robust interactive demo in itself would distinguish this project from previous work.

The demo can be evaluated using two criteria. The first is framerate, which is easy to measure. The second is robustness - to what extent can parameters be adjusted interactively before the system breaks or becomes unstable? How many edge cases are there, and what are they? How much can the user "play" with the demo, and in what ways, before something breaks?

In addition, simulated human motion can be compared against video or motion capture data of real humans doing a similar task. Motion capture may make for a particularly direct comparison, since animation from either motion capture or this project could be used to animate the same skeleton.

4. RESEARCH TIMELINE

Short Term (Short term schedule suggested by Aline):

- **Week 0 (9/18/2012):** Read the "Spacetime Constraints" paper.
- **Week 1:** A moving particle in 2D
- **Week 2:** A worm (2-link chain) in 2D
- **Week 3,4:** Lamp in 2D
- **Week 5, 6:** Lamp in 3D

Project Milestone Report (Alpha Version)

- Animation of lamp jumping in 2D.
- Specifications written for the interactive demo, including what the demo scene is, what the user can interact with, and high-level software architecture of the demo (i.e. how the optimization will be tweaked).

Project Milestone Report (Beta Version)

- Animation of a human. Should support locomotion and reaching in some capacity. I.e. should be able to walk to part of a room, reach for an object, and touch it.

December 1 thru Beginning of Spring Semester

- The character should try to balance or otherwise respond to perturbations, i.e. like an external force applied on the character. Some work on this will be conducted over winter break.

Spring Semester

- The spring semester should be spent implementing the demo. Since the demo is not clearly defined yet, the schedule is more nebulous. A rough goal is 3 weeks for the alpha version, a month for the beta, and another month to finish up the final version. While working on the beta, real-world data will be collected to compare to the simulation. This data will be used as a reference all the way until the end.

Project Final Deliverables

- Interactive demo complete. The user should be able to tweak several parameters without breaking the system. The system should run at a "playable" framerate.
- Documentation, especially on how someone else could plug the animation system into their own Unity project.
- Report on how robust the demo is, what the parameters are and how far they can be tweaked.
- Comparison of demo with video, motion capture, or some sort of "ground truth" data.

.

Project Future Tasks

- Refactor of the code in the demo and in the animation system in order to make it a framework that has a clean interface for use with other applications.
- Create two more demos in order to determine if the approach taken with this project is robust enough to serve as a framework, rather than a simulation of one specific scene. If not, determine why, setting the scene for a future research project.

GANT Chart – see last page

References

[KP11] Junggon Kim; Pollard, N.S.; , "Direct Control of Simulated Nonhuman Characters," *Computer Graphics and Applications, IEEE* , vol.31, no.4, pp.56-65, July-Aug. 2011 doi: 10.1109/MCG.2011.58

[KWC*10] Kwatra, N.; Wojtan, C.; Carlson, M.; Essa, I.A.; Mucha, P.J.; Turk, G.; , "Fluid Simulation with Articulated Bodies," *Visualization and Computer Graphics, IEEE Transactions on* , vol.16, no.1, pp.70-80, Jan.-Feb. 2010.

[LCC*07] Huai-Che Lee; Chia-Ming Chang; Jui-Shiang Chao; Wei-Te Lin; , "Realistic Character Motion Response in Computer Fighting Game," Multimedia, 2007. ISM 2007. Ninth IEEE International Symposium on , vol., no., pp.169-175, 10-12 Dec. 2007.

[LZ11] C. Liu and V. Zordan, “Natural User Interface for Physics-Based Character Animation,” Motion in Games, pp. 1–14, 2011.

[MdLH10] Igor Mordatch, Martin de Lasa, and Aaron Hertzmann. 2010. Robust physics-based locomotion using low-dimensional planning. In *ACM SIGGRAPH 2010 papers* (SIGGRAPH '10), Hugues Hoppe (Ed.). ACM, New York, NY, USA, , Article 71 , 8 pages.

[NWB10] Nam Nguyen, Nkenge Wheatland, David Brown, Brian Parise, C. Karen Liu, and Victor Zordan. 2010. Performance capture with physical interaction. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (SCA '10). Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 189-195.

[RH91] Marc H. Raibert and Jessica K. Hodgins. 1991. Animation of dynamic legged locomotion. SIGGRAPH Comput. Graph. 25, 4 (July 1991), 349-358.

[SHP04] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. 2004. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. In *ACM SIGGRAPH 2004 Papers* (SIGGRAPH '04), Joe Marks (Ed.). ACM, New York, NY, USA, 514-521.

[WK88] Andrew Witkin and Michael Kass. 1988. Spacetime constraints. SIGGRAPH Comput. Graph. 22, 4 (June 1988), 159-168

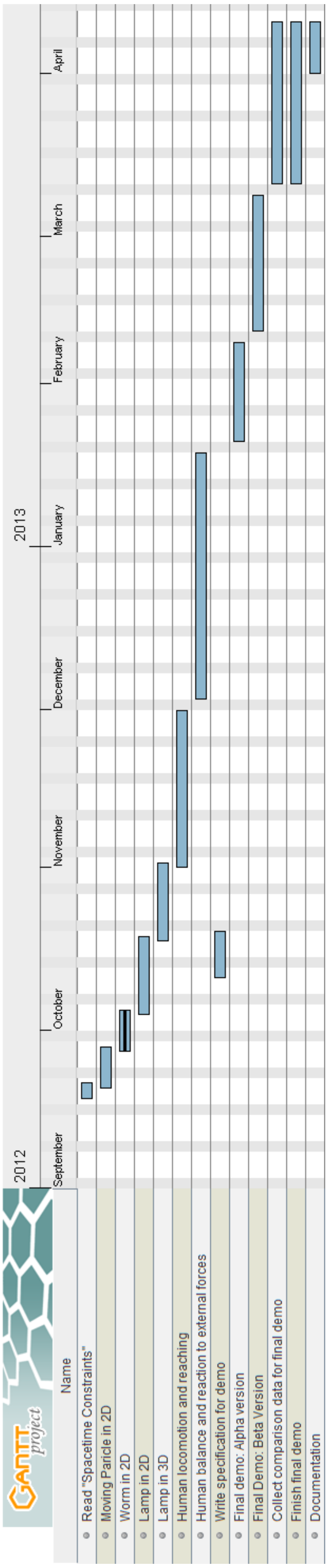
[YLdP07] KangKang Yin, Kevin Loken, and Michiel van de Panne. 2007. SIMBICON: simple biped locomotion control. In ACM SIGGRAPH 2007 papers (SIGGRAPH '07). ACM, New York, NY, USA, , Article 105 .

Acknowledgements

Aline Normoyle and Norman Badler for supervision.

Mubbasir Kapadia for ideas / suggestions on background reading.

Figure 1: *Gant Chart. Rough graphical approximation of the project plan.*



.