# Ski Stunt Simulator: Experiments with Interactive Dynamics

Michiel van de Panne*
University of British Columbia

Cedric Lee
Relic Entertainment Inc.

(a) (b) (c) (d) (e)

Figure 1: Ski Stunt Simulator (a) mouse-based control over the pose of the skier. (b) Visualization of reaction forces and the center-of-mass trajectory (c) Front flip over a ravine. (d) Balancing on a stack of crates. (e) Santa goes skiing.

## Abstract

Physics-based animation of characters has been a disappointment in terms of its successful application to animation for film or games. We present Ski Stunt Simulator, which is an interactive physics-based simulation of a skier whose actions can be controlled using mouse gestures. This simulator was developed with the goal of capturing some of the characteristics of many real sports, which can be simultaneously difficult and rewarding to master and which leverage our ability to efficiently learn motor control tasks with practice. The simulator has applications as a game, as an educational tool for biomechanics, and eventually as a sports prototyping tool.

**Keywords:** Animation, Physics-based Simulation, Interactive Control

## 1 Introduction

Physics-based animation of characters has been a disappointment in terms of its successful application to animation for film or games. We present Ski Stunt Simulator, which is an interactive physics-based simulation of a skier whose actions can be controlled using mouse gestures. This simulator was developed with the goal of capturing some of the characteristics of many real sports, which can be simultaneously difficult and rewarding to master and which leverage our ability to efficiently learn motor control tasks with practice. The simulator has applications as a game, as an educational tool for biomechanics, and eventually as a sports prototyping tool.

## 2 Introduction

While physics-based animation has been successfully adopted for the animation of fluid and cloth, it use for character animation has met with much less success. This is in large part due to the difficulties involved in providing appropriate control over a character's motion. In many ways, this is a natural reflection of the difficulty of performing human motions, which are shaped by a multitude of constraints involving a complex combination of physics, musculature, planning, and control. The examples where physics-based character animation has been used in games typically involve the simulation of falling motions, these being situations that can be simulated with passive "rag-doll" physics or some limited form of control.

As shown in Figure 2, animation techniques can draw on algorithms, motion capture data, or the animator (a user) in order to create a new motion. The interactive control technique used in Ski Stunt Simulator is a mix of algorithmic control, namely the physics-based simulation, and animator control. Unlike a typical game, where a player provides typically might initiate a particular prescripted action such as a back flip by using a sequence of discrete button presses, our mouse-based interface provides for continuous control over the character's actions. The final motion is then the product of the user's control acting in concert with the underlying physics. While the use of an underlying physics-based simulation makes the job of controlling the character's motion more difficult, it also opens up a wider space of possible motions because it allows for novel motions that are precluded by the prescripted action vocabularies of most games.

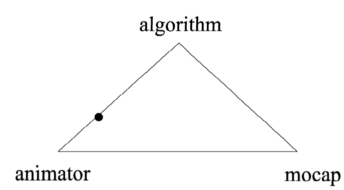


Figure 2: The animation design space. Animations can be created algorithmically using a model of motion, based upon motion capture data, designed directly by an animator, or some mix of these techniques.

Interactive techniques that allow for continous mappings of motion from a performer or animator to a character have a fairly long history, and includes both performance animation and computer puppetry[Sturman 1998]. The system presented here builds principally on the previous work on interactive control of dynamic simulations[Laszlo et al. 2000]. Several other notable results exist on the

*email: van@cs.ubc.ca

interactive control involving dynamical systems[Forsey and Wilhelms 1988; et al. 1993; Oore et al. 2002; Kokkevis et al. 1996; Troy and Vanderploeg 1995; Troy 1998], although these have typically been targetted specifically at animators rather than for use by a more general audience. More recent developments has seen the development of game middleware for physics-based simulation of articulated figures[MathEngine 2003; Havok 2003], although appropriate methods of control remain a largely unresolved issue.

# 3 Simulator

The simulation consists of a four-link planar articulated figure, with links representing the torso and head, the arms, the upper leg, and a single link representing the combination of the lower leg, foot, and ski. Each link has a mass and moment of inertia appropriate to a character of the given size. The simulator handles multiple active objects, typically consisting of the skier, the static environment, and a series of props that are placed in the environment. Props include a stack of crates, a breakable bridge, and other objects such as rocks, etc.

The skis of the skier (see Figure 1) detach when the computed binding forces become too large, thereby causing the bindings to release. Most of the active objects (props) are initialized to be 'sleeping', and only become active elements in the ongoing simulation when another active object makes contact with them. The objects also have criteria for returning to sleep once they come to rest, allowing them to be removed from the simulation loop. A fixed time-step method is used in combination with a penalty method. This was motivated by the need to simplify the mechanics of keeping the simulation running at a constant frame-rate. A particle system is invoked when then skier and other objects impact the ground model. The number of particles and their initial velocities are chosen to be proportional to the loss of momentum of the impacting object. The java version makes use of a somewhat reduced version of the simulation engine, the principal difference being the lack of multiple active objects – the skier is the only animated object.

# 4 Motion Control

The motion of the skier is controlled by specifying a desired pose for the character using the mouse position. The target joints angles for the shoulder, hip, and knee are computed as a linear function of the mouse $x$ and $y$ position. As illustrated in Figure 1, the vertical position of the mouse controls whether the skier is in a crouch or is extended. The horizontal position controls the forwards and backwards lean of the skier.

Given a target position for the joints of the skier, a torque is computed using a proportional derivative (PD) controller as $\tau = k_p(\theta_d - \theta) - k_v\dot{\theta}$. This implements a damped angular spring that forces each joint towards its target angle.

A key feature of the interface is that initial slow and deliberate movements become transformed into smooth gestures with sufficient practise. This natural 'chunking' of continuous actions is a particularly interesting aspect of the game because it involves learning a natural "motion alphabet" for the skier. Many of the mouse actions can be thought of in terms of a cursive script – for example, the input motion required to perform a backflip can be thought of as drawing a large 'C' at a particular speed.

While most users employ a mouse as the interface device, a tablet and stylus seems to allow for better performance, given the fine motor control that a stylus affords when combined with the extensive experience of people with writing. A joystick was found to be less satisfactory; the fixed neutral point makes them more suited to specifying a direction and a magnitude, whereas the interface for the game necessitates more precise positional control. One can imagine that it is more difficult to sketch the shape of a 'C' using a joystick than it is with a mouse or stylus. One last point of interest is that the accelerations implemented in the mouse drivers do not seem to significantly affect user performance.

# 5 Applications

## 5.1 Game

A key goal of the project was to develop a simulator that shares some of the features that make sports such as snowboarding and skateboarding attractive. First and foremost, many individual sports involve a significant creative element: "What can I do with this piece of equipment on this terrain?". Second, one should expect to become significantly better with practise. Ski Stunt Simulator affords both of these elements, which is evidenced in a variety of ways. The java version sees an average of 330,000 runs per day, or about 10,000,000 runs per month. More interestingly, the ability to record any given run and save it in a compact file format allows the ability to contribute towards a public 'stunt gallery', which has become a testament to the level at which experts are able to control the motions of the skier.

## 5.2 Educational Tool

The simulator allows for an interactive visualisation of the center of mass (COM), the COM trajectory, and the ground reaction forces, as shown in Figure 1(b). This lends itself to the teaching of concepts related to moments of inertia, angular momentum, ground reaction force patterns, the timing requirements for executing motions, and the study of motor learning effects. We hope to explore this further in the near future.

We have also experimented with interactive control of a platform diving simulation and the motion of a person on a swing, as shown in Figure 3.

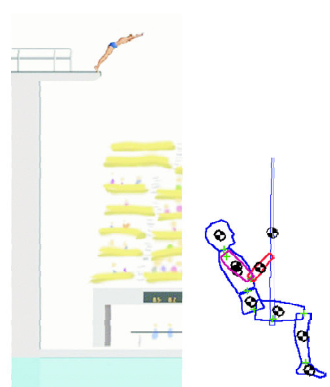

Figure 3: Two other figures we have experimented with: a platform diver and a person on a swing.

## 5.3 Sports Prototyping Tool

Simulators allow for experimentation with potentially new motions. For example, one might explore the possibility of performing a new type of jump or flip, and become familiar with the required motion patterns and timing requirements by working with an appropriate simulator. While the current version is clearly inadequate to be used in this fashion, we see significant future potential in this direction.

## 6 Conclusions

We have presented Ski Stunt Simulator, a particular application of physics-based character animation that provides a compelling game experience despite the simplicity of the current planar implementation. While the interface is intially challenging to learn, its flexibility supports the type of experimentation that results in a compelling game playing experience.

## References


ET AL., T. E. 1993. Interactive control of biomechanical animation. *The Visual Computer*, 459–465.

FORSEY, D., AND WILHELMS, J. 1988. Techniques for interactive manipulation of articulated bodies using dynamic analysis. In *Proceedings of Graphics Interface '88*, 8–15.

HAVOK. 2003. http://www.havok.com/.

KOKKEVIS, E., METAXAS, D., AND BADLER, N. 1996. User-controlled physics-based animation for articulated figures. In *Proceedings of Computer Animation '96*.

LASZLO, J., VAN DE PANNE, M., AND FIUME, E. L. 2000. Interactive control for physically-based animation. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH / Addison Wesley Longman, Computer Graphics Proceedings, Annual Conference Series, 201–208. ISBN 1-58113-208-5.

MATHENGINE. 2003. http://www.mathengine.com/.

OORE, S., TERZOPOULOS, D., AND HINTON, G. 2002. Local physical models for interactive character animation. *Computer Graphics Forum 21*, 3, 337–346. ISSN 1067-7055.

STURMAN, D. J. 1998. Computer puppetry. *IEEE Computer Graphics and Applications 18*, 1 (Jan-Feb), 38–45.

TROY, J., AND VANDERPLOEG, M. 1995. Interactive simulation and control of planar biped walking devices. In *Workshop on Simulation and Interaction in Virtual Environments*, 220–224.

TROY, J. 1998. Real-time dynamic balancing and walking control of a 7-link planar biped. In *Proceedings of ASME Design Engineering Technical Conferences*.