Mass-Spring Systems

Lorenzo Marsicano & Athos van Kralingen

The paper(s)

3rd Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2006)
C. Mendoza, I. Navazo (Editors)

Position Based Dynamics

Matthias Müller Bruno Heidelberger Marcus Hennix John Ratcliff

AGEIA

Abstract

The most popular approaches for the simulation of dynamic systems in computer graphics are force based. Internal and external forces are accumulated from which accelerations are computed based on Newton's second law of motion. A time integration method is then used to update the velocities and finally the positions of the object. A few simulation methods (most rigid body simulators) use impulse based dynamics and directly manipulate velocities. In this paper we present an approach which omits the velocity layer as well and immediately works on the positions. The main advantage of a position based approach is its controllability. Overshooting problems of explicit integration schemes in force based systems can be avoided. In addition, collision constraints can be handled easily and penetrations can be resolved completely by projecting points to valid locations. We have used the approach to build a real time cloth simulator which is part of a physics software library for games. This application demonstrates the strengths and benefits of the method.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object ModelingPhysically Based Modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and RealismAnimation and Virtual Reality

1. Introduction

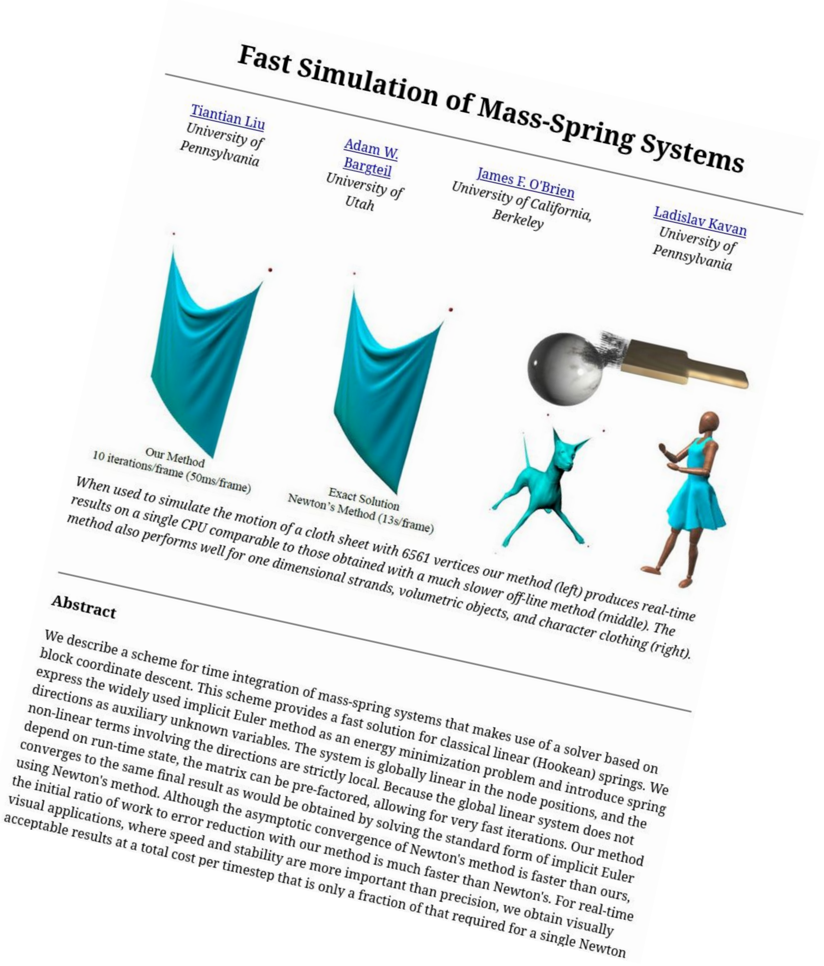
Research in the field of physically based animation in computer graphics is concerned with finding new methods for the simulation of physical phenomena such as the dynamics of rigid bodies, deformable objects or fluid flow. In contrast to computational sciences where the main focus is on accuracy, the main issues here are stability, robustness and speed while the results should remain visually plausible. Therefore, existing methods from computational sciences can not be adopted one to one. In fact, the main justification for doing research on physically based simulation in computer graphics is to come up with specialized methods, tailored to the particular needs in the field. The method we present falls into this category.

The traditional approach to simulating dynamic objects has been to work with forces. At the beginning of each time step, internal and external forces are accumulated. Examples of internal forces are elastic forces in deformable objects or viscosity and pressure forces in fluids. Gravity and collision forces are examples of external forces. Newton's second law of motion relates forces to accelerations via the mass. So using the density or lumped masses of vertices, the forces are transformed into accelerations. Any time integration scheme can then be used to first compute the velocities from the accelerations and then the positions from the velocities. Some approaches use impulses instead of forces to control the animation. Because impulses directly change velocities, one level of integration can be skipped.

In computer graphics and especially in computer games it is often desirable to have direct control over positions of objects or vertices of a mesh. The user might want to attach a vertex to a kinematic object or make sure the vertex always stays outside a colliding object. The method we propose here works directly on positions which makes such manipulations easy. In addition, with the position based approach it is possible to control the integration directly thereby avoiding overshooting and energy gain problems in connection with explicit integration. So the main features and advantages of position based dynamics are

- Position based simulation gives control over explicit integration and removes the typical instability problems.

© The Eurographics Association 2006.



Fast Simulation of Mass-Spring Systems

Tiantian Liu, University of Pennsylvania
Adam W. Bargteil, University of Utah
James F. O'Brien, University of California, Berkeley
Ladislav Kavan, University of Pennsylvania

When used to simulate the motion of a cloth sheet with 6561 vertices our method (left) produces real-time results on a single CPU comparable to those obtained with a much slower off-line method (middle). The method also performs well for one dimensional strands, volumetric objects, and character clothing (right).

Abstract

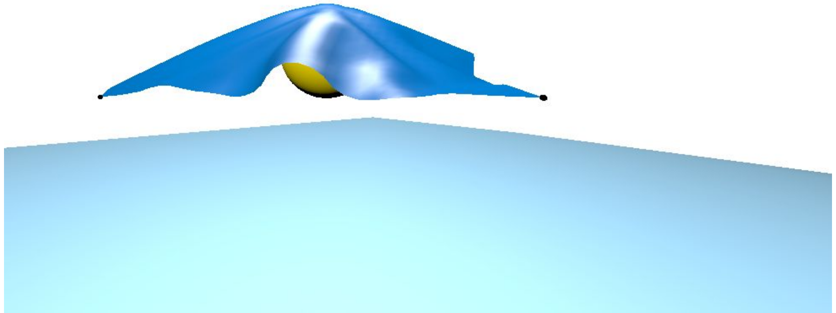
We describe a scheme for time integration of mass-spring systems that makes use of a solver based on block coordinate descent. This scheme provides a fast solution for classical linear (Hookean) springs. We express the widely used implicit Euler method as an energy minimization problem and introduce spring directions as auxiliary unknown variables. The system is globally linear in the node positions, and the non-linear terms involving the directions are strictly local. Because the global linear system does not depend on run-time state, the matrix can be pre-factored, allowing for very fast iterations. Our method converges to the same final result as would be obtained by solving the standard form of implicit Euler using Newton's method. Although the asymptotic convergence of Newton's method is faster than ours, the initial ratio of work to error reduction with our method is much faster than Newton's. For real-time visual applications, where speed and stability are more important than precision, we obtain visually acceptable results at a total cost per timestep that is only a fraction of that required for a single Newton

Tooling

- Raylib
- RayGUI (not recommended)
- Eigen
- YAML-cpp

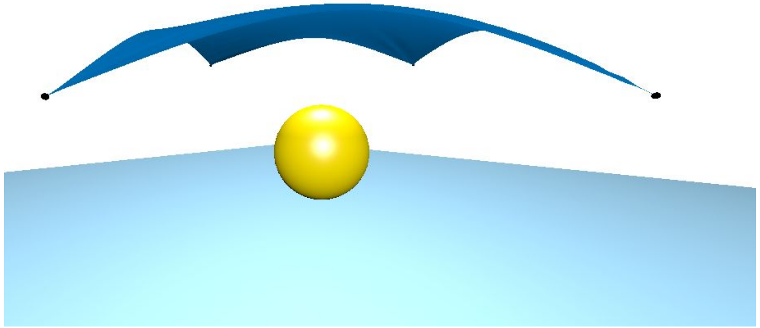
Implementation

- Integration using gradient descent
- Dampening
- Interpenetration resolution

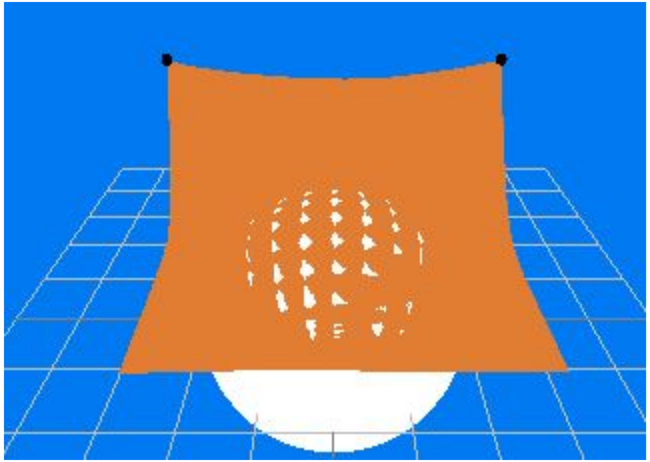


External Forces

- Wind
- Collision responses



Visual Corrections



Demo

Questions?