Implementation of Fast Frictional Dynamics for Rigid Bodies

Danny M. Kaufman Timothy Edmunds Dinesh K. Pai
Department of Computer Science, Rutgers University

1 Introduction

This sketch will present some important details that went into the implementation of our Fast Frictional Dynamics algorithm [Kaufman et al. 2005]. Our goal is the fast simulation of large sets of non-convex rigid bodies. By avoiding repeated pairwise comparisons between bodies, we are able to formulate a multi-body simulation algorithm with a complexity that is linear in the total number of contacts per time step.

We employ a velocity-level treatment of contact resolution. For each rigid body, we consider the mass, location, and velocity of multiple contacts, with multiple bodies, at the time of maximum compression of each contact. This allows each body to receive the appropriate amount of imparted momentum from the collisions in which it is involved. We use a Quadratic Program (QP) to tell us which contacts are the most active, and to obtain the effective normal impulse generated by them. Because this impulse combines the contributions of all active contacts, we are able to get good frictional behavior, at low cost, using a new Coulomb-like friction model that requires a single additional QP.

Our algorithm can be combined with a wide variety of broad- and narrow-phase collision detection systems. We have employed an off the shelf narrow-phase system (PQP [Gottschalk et al. 1996]) and implemented a simple spatial hash table for our broad-phase. More efficient collision detectors could be seamlessly swapped into our implementation.

We have implemented our simulation using fixed size time stepping (although there is no reason an adaptive stepping approach could not be used). At each step we advance the state of all rigid bodies in our system. Individual steps can be conceptually partitioned into the following five phases: An initial forward Euler half-step for all bodies, broad phase collision detection, contact point determination between colliding bodies, resolution of all contacts on all bodies, and a final forward Euler half-step for all bodies.

2 Description

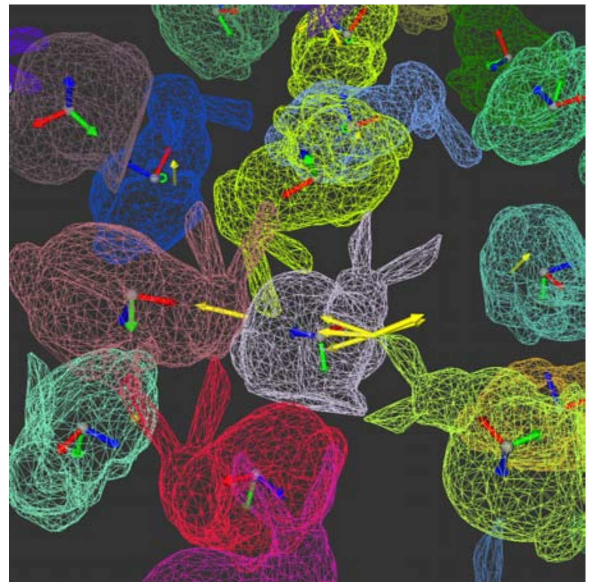
In this sketch we will explain some details of the approach we take to ensure that the phases described above can be performed in a memory-efficient fashion while also minimizing the necessity of performing computations that depend on more than one body's state. This includes the following techniques:

Contact Streaming. Contacts found in the contact point determination phase are processed sequentially in a single pass. This ensures that the entire set of contacts never needs to be explicitly constructed.

Active Contact Sets. We place a large (but fixed) upper bound on the number of contacts that can affect a body in a given time step. This allows us to avoid dynamic memory allocation.

We will also outline some of the design decisions we make in our implementation that allow us to efficiently approximate our underlying physical model, including:

Point Culling. When choosing the points to represent the contact between two interpenetrating, non-conforming bodies, we use points from the space curve formed by the intersection of the two bodies' meshes (rather than determining which vertices on one body's mesh are inside the other).



Surrogate Constraint Method (SCM). We perform two separable, convex QPs for each contacting body during each iteration. We approximate these QPs using the SCM algorithm [Yang and Murty 1992]. SCM moves towards our desired solution using a conservative step, with guaranteed convergence. SCM also works well with our contact streaming.

Additionally, we will discuss simple techniques that reduce the computational cost of contact resolution and explicit Euler stepping, such as:

Maximum Compression Velocity Computation. In order to maintain momentum conservation we need to find the maximum compression velocity (v_{mc}) for each contact/interpenetration detected. We are able to do this efficiently because v_{mc} can be calculated without needing to determine the actual instant at which maximum compression occurs. All values required to find v_{mc} are computed in local reference frames. This allows us to maintain our streaming contact model, and reduces the number of frame transformations (matrix multiplies) we need to perform for each contact.

Body Frame Dynamics. All dynamics computations are performed in body frame coordinates. In this frame we are able use an axis aligned, diagonalized inertia matrix. This reduces the cost of the weighted dot products (of the form $x^T M y$) we perform throughout our implementation. This also simplifies the explicit Euler steps we take during each iteration.

References

GOTTSCHALK, S., LIN, M. C., AND MANOCHA, D. 1996. OBB-tree: A hierarchical structure for rapid interference detection. In *Proceedings of SIGGRAPH 96*.

KAUFMAN, D. M., EDMUNDS, T., AND PAI, D. K. 2005. Fast frictional dynamics for rigid bodies. *ACM Transactions on Graphics (SIGGRAPH 05)*.

YANG, K., AND MURTY, K. G. 1992. New iterative methods for linear inequalities. *Journal of Optimization Theory and Applications 72*, 1, 163–185.