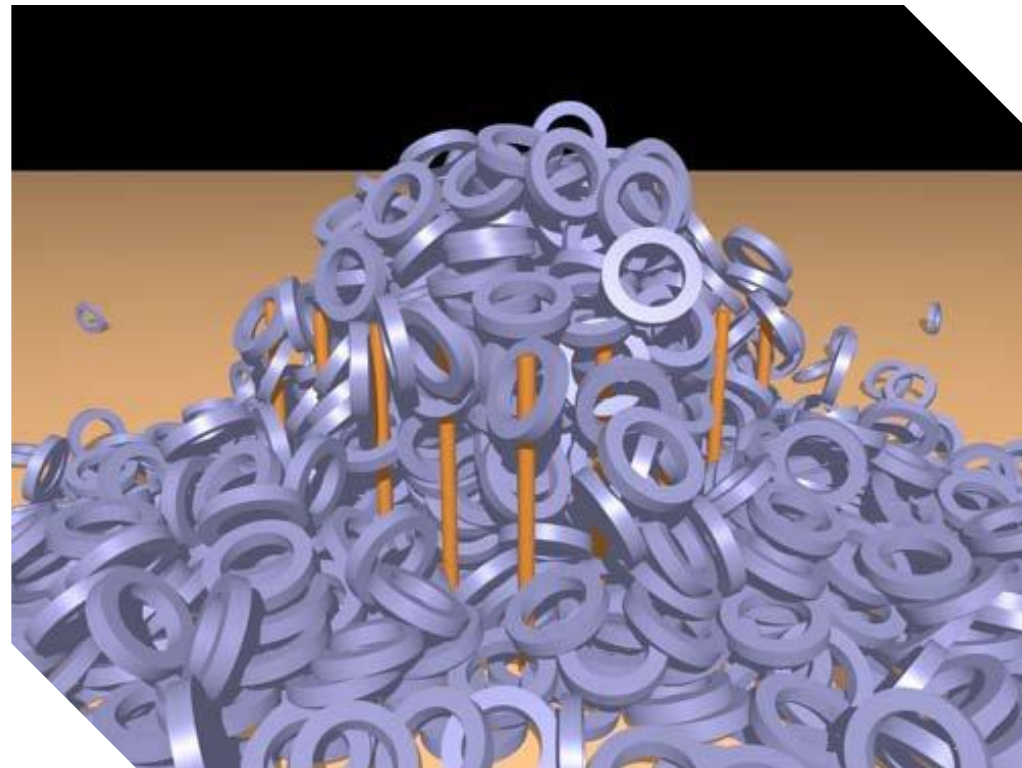


Nonconvex Rigid Bodies with Stacking

Eran Guendelman (Stanford University)

Robert Bridson (Stanford University)

Ronald Fedkiw (Stanford University)





Introduction

- Simulation of nonconvex rigid bodies focusing on interactions such as collision, contact, friction and stacking
- Focusing on obtaining a particularly appealing simulation of rigid bodies emphasizing large scale problems with many frictional interactions
- Geometry representing with triangulated surfaces and signed distance function defined on a grid
- Propose novel approach to time integration merging in collision and contact processing
- Propose new shock propagation algorithm
- Demonstration on variety of problems



Topics

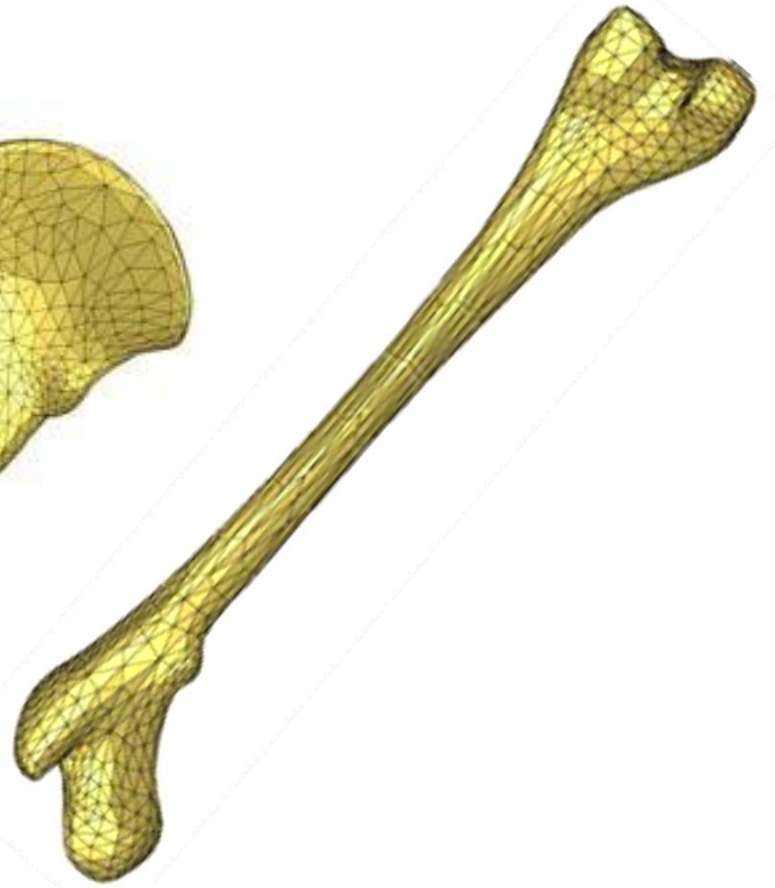
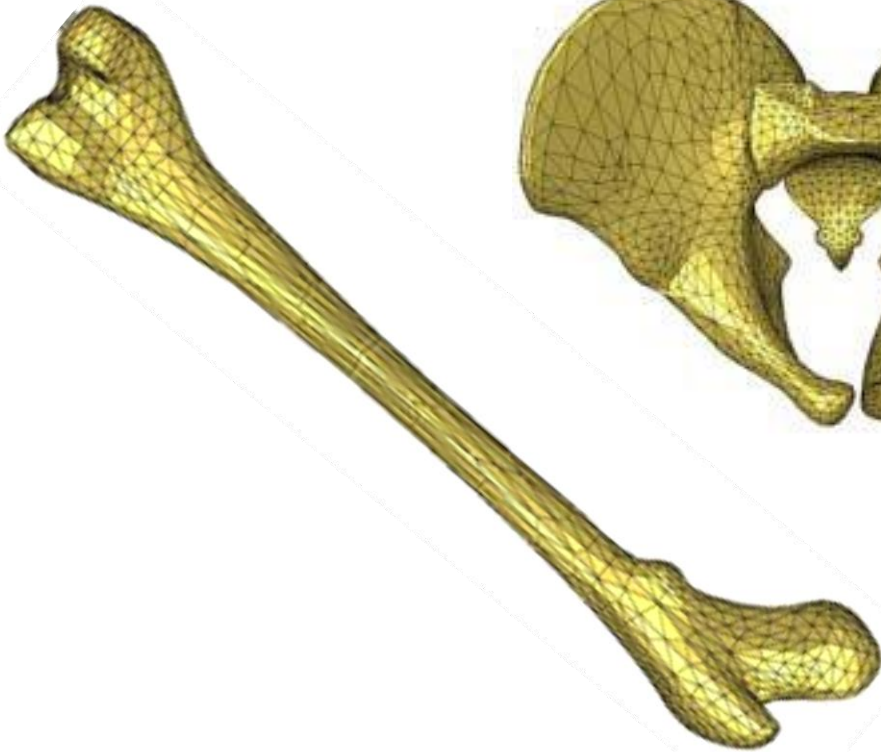
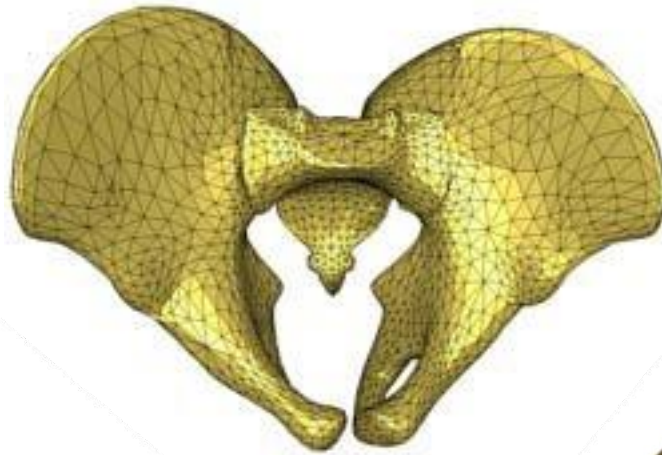
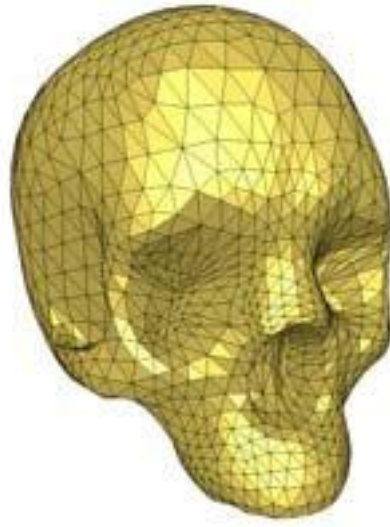
- Related work
- Geometric Representation
- Interference Detection
- Time integration
- Collisions
- Static and kinetic friction
- Contact
 - Contact graph
 - Shock propagation
- Rolling and pinning friction

Related work

- [Hahn 1988] proposed collisions with static friction if the results was in friction cone, otherwise used kinetic friction. Used threshold velocities. If velocitie was smaller then threshold, objects were assumed to be in contact.
- Implicitly defined surfaces were used for collision modeling by [Terzopoulos et al. 1987] to create repulsive force fields around objects and [Pentland and Williams 1988, Sclaroff and Pentland 1991] exploited fast inside/outside test.
- [Milenkovic 1996] used position based physics to simulate the stacking of convex objects and propose ways of making the simulation appear more realistic

Geometric representation

- Rigid bodies do not deform
 - Typically represented with triangulated surfaces
 - For efficiency, store the object space representation with center of mass
 - Also store an object space signed distance function stored on uniform grid or an octree grid depending on whether speed or memory is deemed to be important in subsequent calculations
 - They use negative values of Φ inside the rigid body and positive values of Φ outside so that the normal is defined as $N = \nabla \phi$



Geometrická reprezentácia

- SDM (V) – Is $N \times N \times N$ regular grid, where each unit cell with center point p stores the signed distance to the closest point on the surface of some volume V
- Pros: efficient overlap test, fast contact generation and penetration depth computation for arbitrary shaped, non-convex objects with complex and highly tessellated geometry, suitable for real-time applications as games
- Cons: huge amount of memory necessary for large scenarios, large number of redundant contacts generated during collision detection

Geometrická reprezentácia

- A signed distance function can be calculated quickly using marching method
- Pros for using both geometric representation methods:
 - One can use the signed distance function to quickly check if a point is inside a rigid body and if so, intersect a ray in the $N = \nabla \phi$ direction with the triangulated surface to find the surface normal at the closest point
 - This allows the treatment of very sharp objects with their true surface normals, but if desired signed distance functions provide smoother and less costly representation [Pandolfi et al. 2002]

Interference detection

- Using vertices of the triangulated surface as sample points to testing between two implicitly defined surfaces
- It's not sufficient test to detect edge-face collisions when both edge vertices are outside the implicit surface
- Error is proportional to edge length
- Solution: mesh with small triangles, we can ignore the error

Interference detection

- They don't consider time dependent collisions
- Limiting the size of time step based on translational and rotational velocities of the objects and the size of bounding boxes
- Accelerations:
 - Inside/outside test can be accelerated by labeling the voxels they are completely inside and outside (in each level of the octree)
 - Labeling the minimum and maximum values of ϕ in each voxel
 - Bounding boxes and spheres are used around each object in order to prune points before doing full inside/outside test



Interference detection

- Also they use a uniform spatial partitioning data structure with local memory storage implemented using hash table in order to quickly narrow down which rigid bodies might be intersecting.

Time integration

- The equations for rigid body evaluation are
- $x_t = v$ position, $q_t = \frac{1}{2}\omega q$ (1.) orientation, F - net force
- $v_t = F / m$ (2.) $F = mg$
- $L_t = \tau$ angular momentum, $L = I\omega$, $I = RDR^T$ inertia tensor (R is orientation matrix, D is diagonal inertia tensor on object space)
- τ - torque
- Using simple forward Euler time integration for listed equations
- Collisions require impulses that discontinuously modify velocity

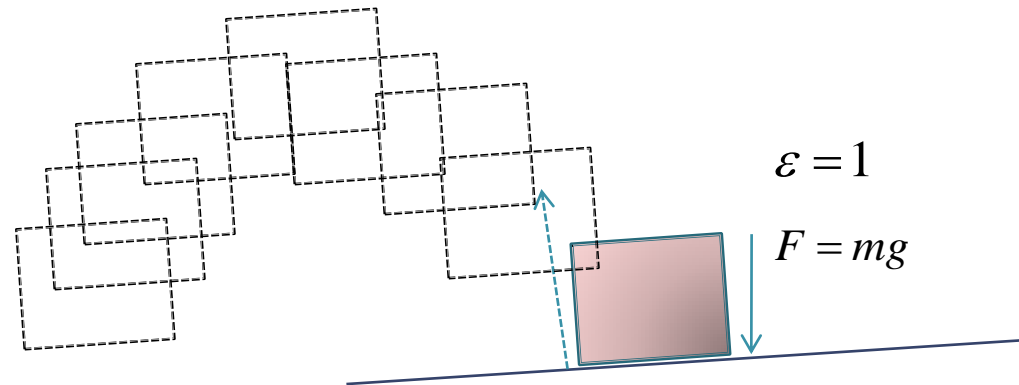
Time Integration

- A novel aspect of our approach is the clean separation of collision from contact without the need for threshold velocities
- They propose a following time sequencing:
 - Collision detection and modeling
 - Advance the velocities using equation 2.
 - Contact resolution
 - Advance the position using equation (1.)

Time integration

- Advantages of this time stepping scheme:
 - consider block sitting on a inclined plane with large coefficient of restitution $\varepsilon = 1$ and friction is large enough that block sit still.
 - In standard time stepping scheme velocities are updated first, followed by collision and contact resolution
 - During the position and velocity update box starts fall during the effect of gravity.
 - In the collision detection detect low velocity collisions and since $\varepsilon = 1$ the block will change direction and bounce upwards at angle down the incline
 - Box will eventually fall back to plane and continue bouncing

Time integration



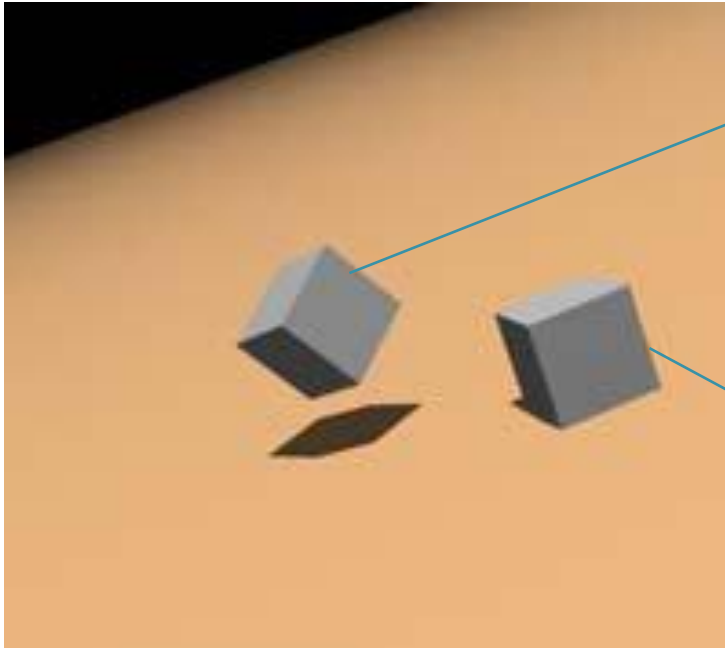
- The same phenomenon causes object sitting on the ground to vibrate as they are subjected to a number of elastic collisions
- Many authors use *ad hoc* threshold velocities in attempt to prune these causes

Time integration

- New time stepping algorithm
 - All objects at rest have zero velocities so in the collision processing we do not get an elastic bounce
 - Next gravity is integrated into velocity and then the contact resolution algorithm correctly stops the objects, so they remain still
 - Last step - position update, nothing happens, the process is repeated
- The key of the algorithm is that contact modeling occurs directly after the velocity is updated with gravity because it resolves forces and the velocity update is where the forces are included in the dynamics.

Time integration

- Experiment
- $\varepsilon = 1$



Standard stepping scheme, box is bouncing down because of $\varepsilon = 1$

Novel stepping approach. Box is sliding down, noticeably better results

Collisions

- When there are many interacting bodies, it can be difficult to treat all collisions
- Propose a method that simultaneously resolves collisions
- It's not a physically correct, but plausible solution
- Collisions are detected by predicting where the objects will move to in the next time step.
- Same technique is used to predict contacts
- Detection of collision and contact is on the same predicted position of the objects

Collisions

- If objects current positions and velocity are x and y , we use for interference the predicted position $x' = x + \Delta t(v + \Delta t g)$ and apply collision impulses to the v
- During contact processing they use predicted position $x' = x + \Delta t v'$ and apply impulses to the new velocity v'
- $v' = v + \Delta t g$ was updated in the velocity update
- Overall structure of the algorithm consist of first moving all rigid bodies to their predicted locations and then identifying and processing all interacting pairs



Collisions

- For each intersecting pair, they identify all the vertices of each body that are inside the other
- They use a method that can deal with non-convex objects with multiple collision regions and multiple interfering points in each region
- Use the standard algebraic collision laws to process the collisions

Collisions

- New aggressive optimization for the point sampling
- First we label all intersecting points and apply collision to the deepest point.
- Instead of re-evolving the objects and repeating the expensive collisions detection algorithm, we keep the objects stationary and use the same list of initially interfering points for the entire procedure.
- After processing a collision all separating points are removed from the list
- Then the deepest non-separating point is identified and the procedure is repeated until the list is empty

Collision

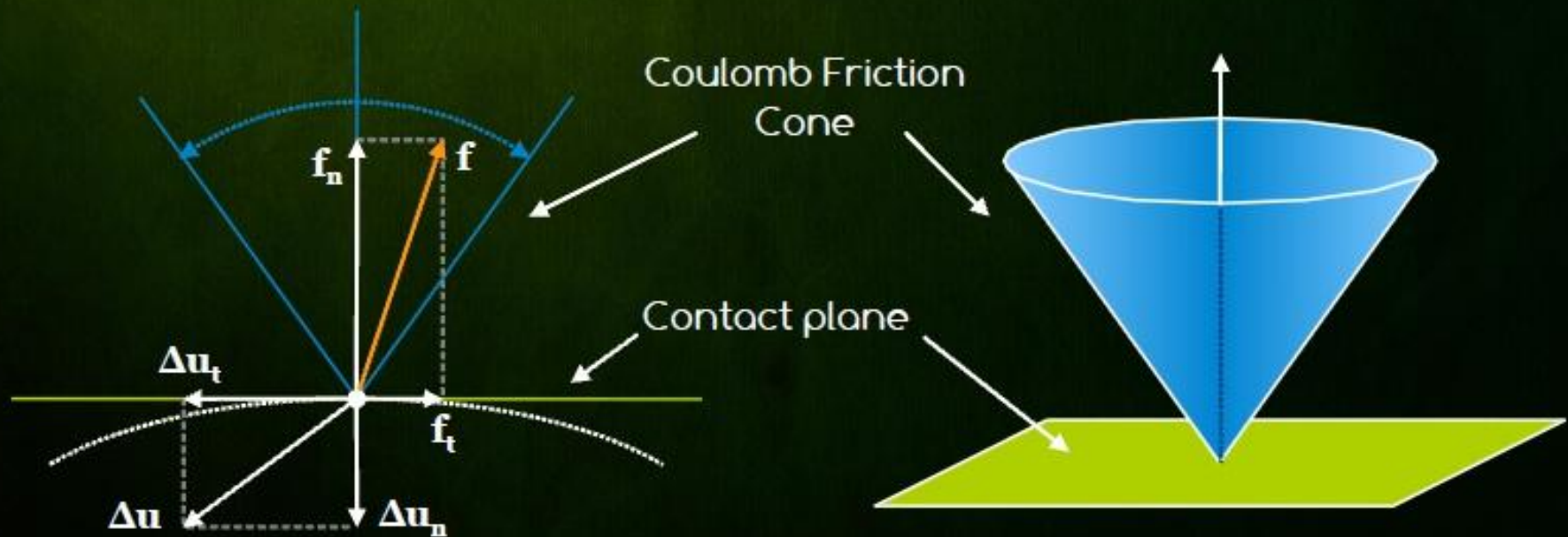
- Each body is assigned a coefficient of restitution
- When two bodies collide, we use minimum between the two coefficients to process collisions
- u_{rel} original relative velocity at the collision point
- With normal N and tangential component
- $u_{rel,n} = u_{rel} \cdot N$ $u_{rel,t} = u_{rel} - u_{rel,n} N$
- Then we apply equal and opposite impulses j to each body to obtain $v' = v \pm j/m$ and $\omega' = \omega \pm I^{-1}(r \times j)$ where r points from their respective centers of mass to the collision location
- New velocities at the point of collision will be $u' = u \pm Kj$

Collision

- Finally $u'_{rel,n} = u_{rel,n} + N^T K_T N j_n$
- K_T is the sum of individual individual K's and
- $j = j_n N$ is our frictional impulse
- So given final relative normal velocity $u'_{rel,n} = -\epsilon u_{rel,n}$ we can find impulse j
- Immovable static object can be treated by setting $K=0$ and not updating velocities

Collisions

- Friction cone



Static and Kinetic Friction

$$u'_{rel} = -\mu u_{rel,n} N$$

- Modifying above algorithm to account kinetic and static friction
- Each body is assigned coefficient of friction
- Assume the bodies are stuck at the point of impact due the static friction and solve for the impulse
- Set $u'_{rel,t} = 0$ so that $u'_{rel} = -\epsilon u_{rel,n} N$ allows us to solve $u'_{rel} = u_{rel} + K_T j$ for impulse j
- if j is in the friction cone, the point is sticking due to static friction and j is an acceptable impulse.
- Otherwise we apply sliding friction

Contact

- The goal of the contact processing algorithm is to resolve the forces between objects.
- After few iterations of collision processing alg. we update the velocities of all rigid bodies and move to contact resolution
- The coefficient of restitution is $\varepsilon = 0$
- Processing interacting pairs in order determined by they list
- Multiple iterations are needed especially for bodies sitting on top of other

Contact

- We identify all the vertices of each body that are inside the other
- We have found our point sampling method to be satisfactory
- They use the same equations to process each contact impulse that were used in the collision algorithm, except we set $\varepsilon=0$
- Start with the deepest point of interpenetration that has a non-separating relative velocity
- Then a new predicted position can be determined and the process repeated until all points are either non-overlapping or separating

Contact

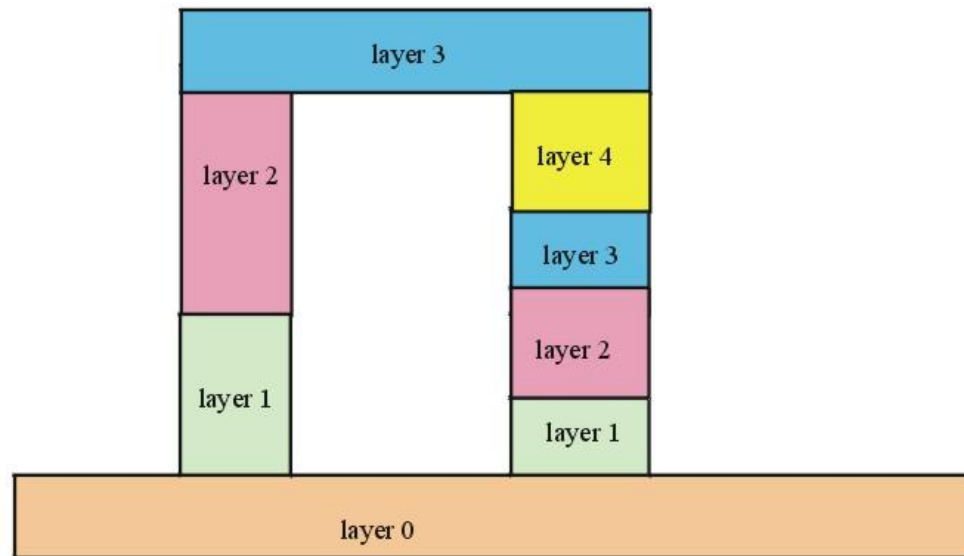
- The aggressive optimization alg. Is not so accurate for contact resolution as for collisions
- To improve accuracy:
 - Gradually decrease the elasticity ε
 - First iteration $\varepsilon = -.9$
 - Second iteration $\varepsilon = -.8$
 - ...
 - Final $\varepsilon = 0$
- Negative coefficient indicates slowing the object down

Contact graph

- Intention of identifying which bodies or groups of bodies are resting on top of others
- Adding directed edge pointing towards the falling object from the other object
- For stack of cubes we get a contact graph that points from the ground up one cube at a time to the top of the stack
- Objects are grouped into the same level if they have a cyclic dependence on each other
- The purpose is to suggest an order in which contacts should be processed

Contact graph

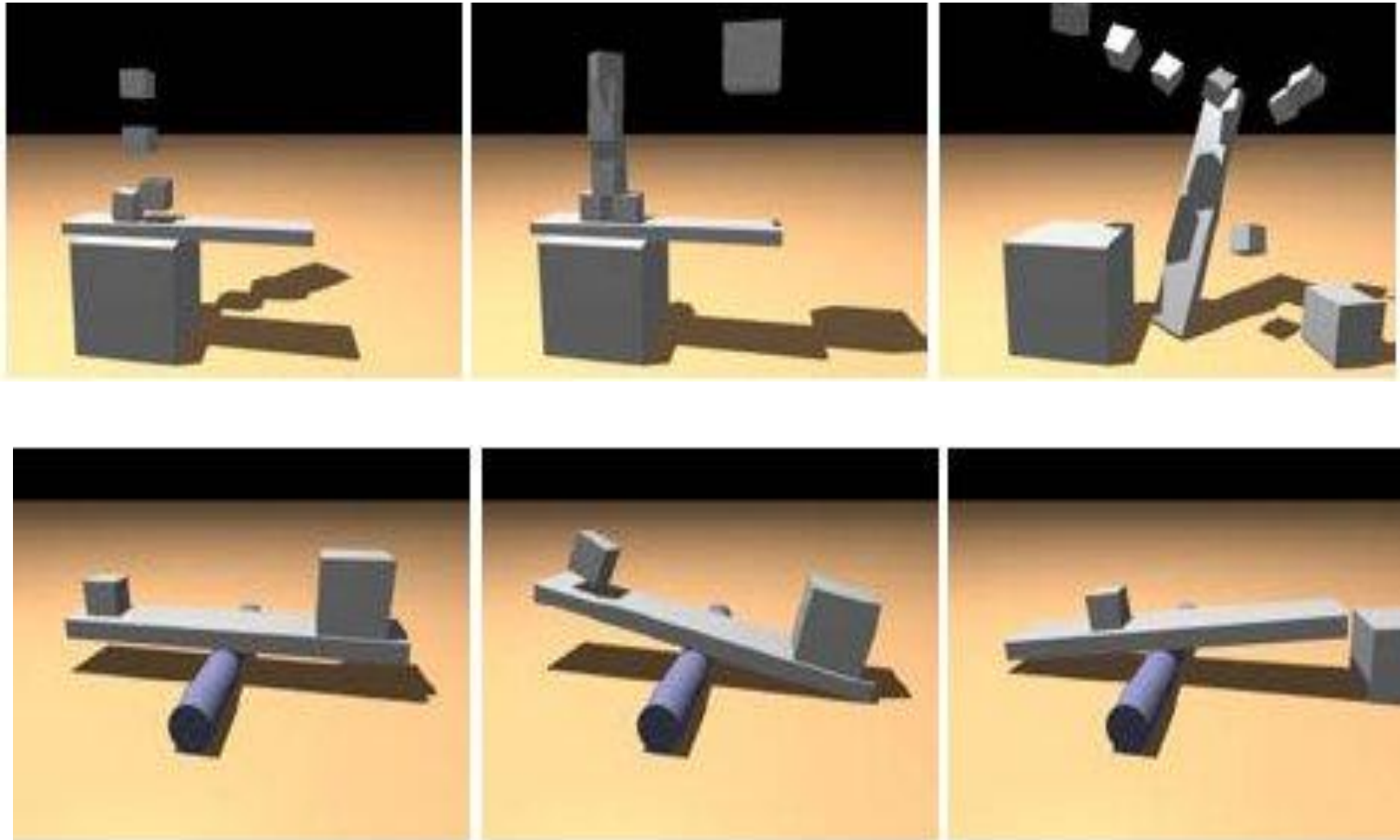
- Contact pairs found for level i are put into a list and reated in any order a number of times
- Then moving to next level



Shock propagation

- Proposed shock propagation method is applied on the last swoop through the contact graph
- After each level is processed, in the last sweep all the objects are assigned infinite mass (matrix K is set to 0)
- Benefit: if an object on lower level with infinite mass is found to be in contact with a higher –level object, its motion is not affected
- Once assigned infinite mass, object retain this mass until shock propagation phase is completed

Shock propagation



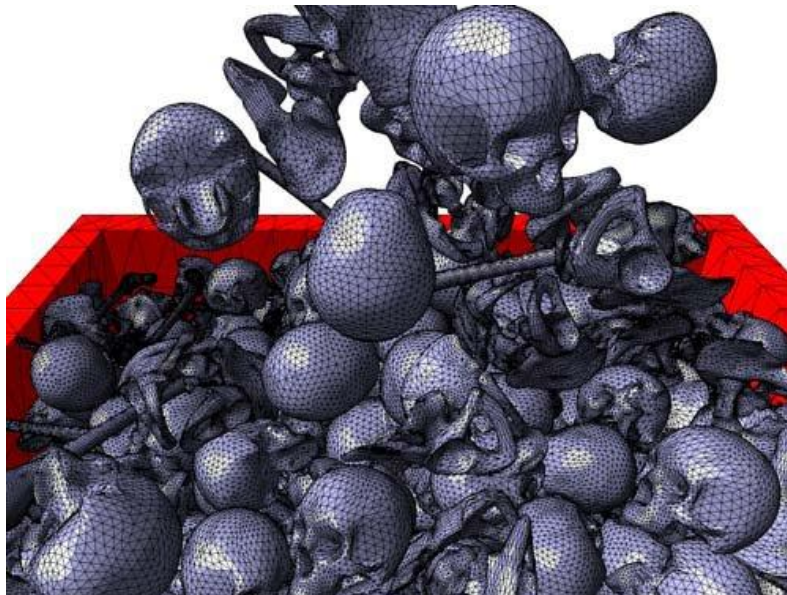
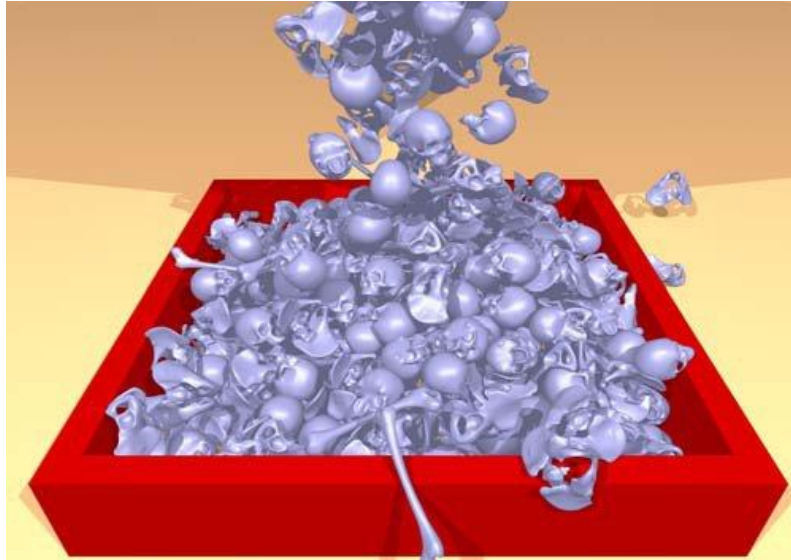
Rolling and Spinning Friction

- An frozen object under of influence of static friction has still freedom to roll and spin
- Proposed approach to treat it as kinetic and static friction
- Denote μ_s μ_r coefficients of rolling and spinning friction depending on local deformation of the object
- Both are based on relative angular velocity ω_r with normal and tangential components $\omega_{rel, n} = \omega_{rel} \cdot N$
 $\omega_{rel, t} = \omega_{rel, n} \cdot N$
- Normal component governs spinning, tangential rolling

Rolling and Spinning Friction

- Modify this reducing the magnitude of the normal and tangential components by $\mu_s \dot{j}_n$ and $\mu_r \dot{j}_n$
- To keep object from reversing, both of this reductions are limited to zero, otherwise preserving the sign.
- Object are sticking $\rightarrow u'_{rel} = 0$
- We have new angular velocity ω'_{rel}
- Next we construct an impulse to achieve both proposed velocities, relative velocity and relative angular velocity

Výsledky



- Testing for large numbers of non-convex objects with high resolution triangulated surfaces falling into stacks with multiple contacts points
- 500-1000 rings = 3-7min/frame
- 2.8 mil triangles per simulation
- 5 collision iterations
- 10 contact iterations
- Single shock propagation using friction

Sources

- Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K. 1987 Elastically deformable models (SIGGRAPH 87)
- Pentland, A., and Williams, J. 1989 Good vibrations: Modal dynamics for graphics and animation
- [Pandolfi, A., Kane, C., Marsden, J., and Ortiz, M. 2002] Time-discretized variational formulation of non-smooth frictional contact
- ...



Thank you for your attention !