Particle Systems for Real-Time Rigid-body, Fluid, and Flocking simulations

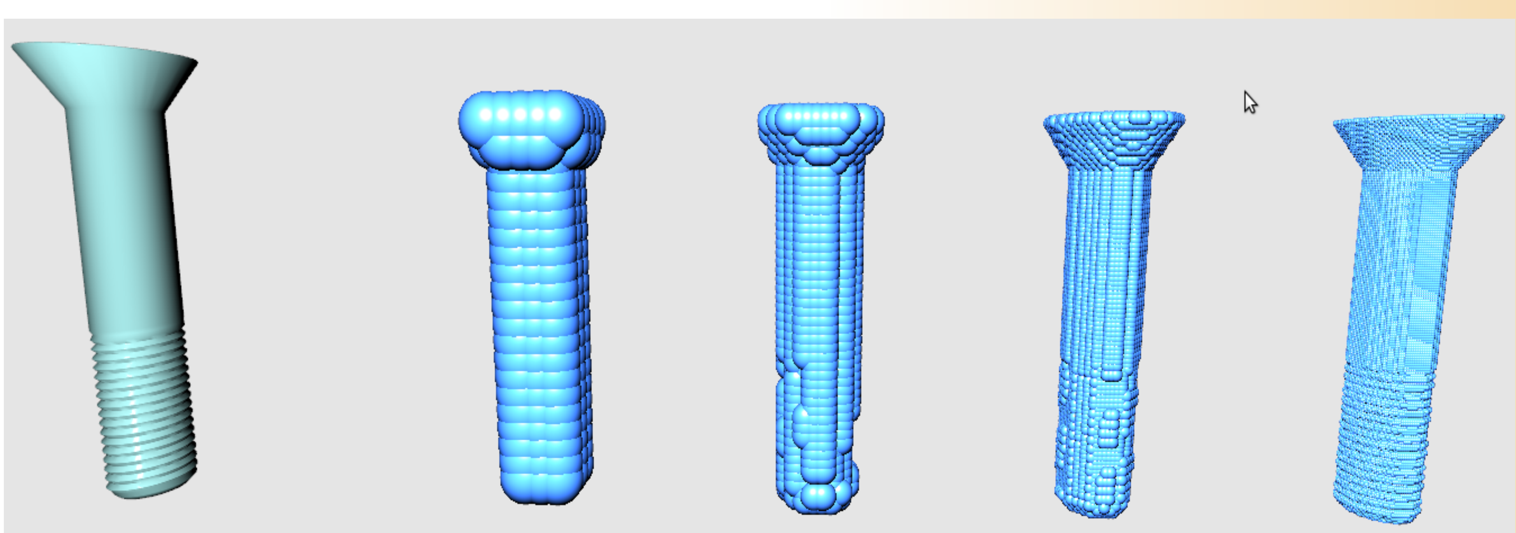
By Andrew Young and Gordon Erlebacher

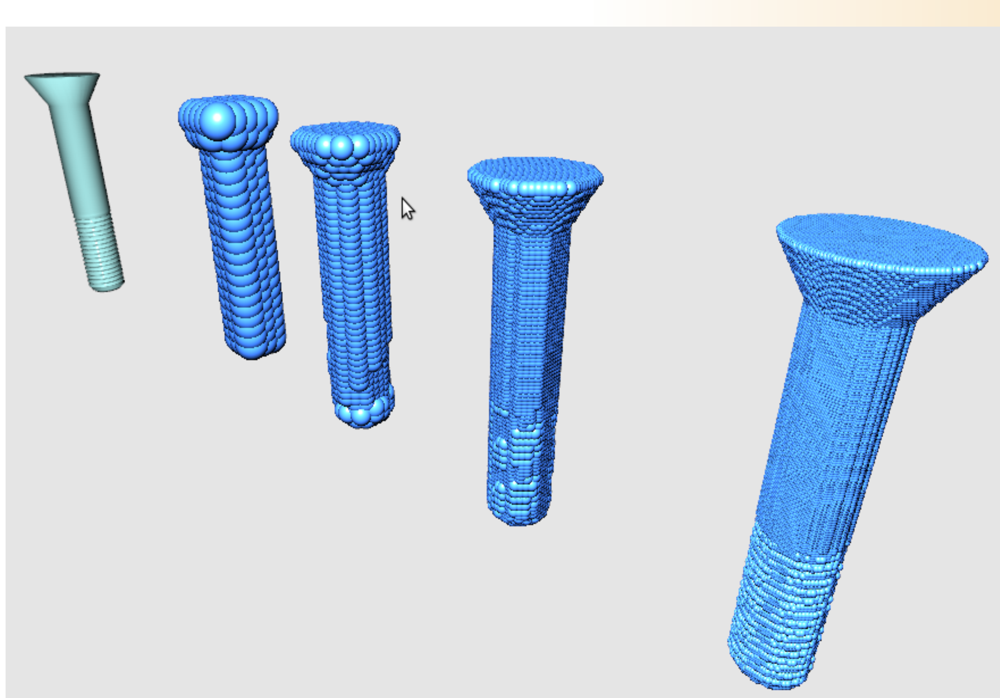
Voxlelize the Mesh

Smoothed particle hydrodynamics (SPH) simulations have an unstructured mesh composed of many particles. Representing rigid-bodies as a set of particles will greatly simplify the calculation of interaction forces between fluids and rigid-bodies. Unfortunately, Rigid-bodies are typically represented as triangular meshes. One would like to have a technique which turns a triangular mesh into a particle representation of that same shape.

The process of creating a particle representation from a triangular mesh is called voxelization. Using OpenGL, one can render a mesh in slices. Initialize the first framebuffer to zeros. Then render the mesh to the depth of one slice of the mesh. For each successive slice, switch the OpenGL blending mode to XOR. Doing so will change pixels which are "on" to "off" only if the incoming signal is "off". Likewise pixels that are "off" change to "on" only if the incoming signal is "on". Effectively, the process turns on when it passes through the surface and only turns off again once it passes through th other side of the mesh.

The voxelization process outputs a 3D Texture where a red pixel indicates the voxel belongs to the inside of the mesh. Therefore, one must iterate over all the voxels adding particles to the domain at a position relative to the r,s, and t indices of the 3D Texture. The user must decide provide a global offset and rotation when actually adding the particles to the domain.





On the far left the triangular mesh is rendered. Each model after this model contains a particle representation of the same model with each step containing twice as many particles as the last

Computing Interaction Forces

During rigid-body simulations, one must compute interaction forces between particles. The computation of interaction forces is categorized by the types of particles interacting. For instance, the framework currently calculates three types of interactions. The interaction between two rigid-body particles, the interaction between a fluid and rigid-body particle, and the interaction between two fluid particles. The interaction forces between fluid particles is defined by the SPH method.

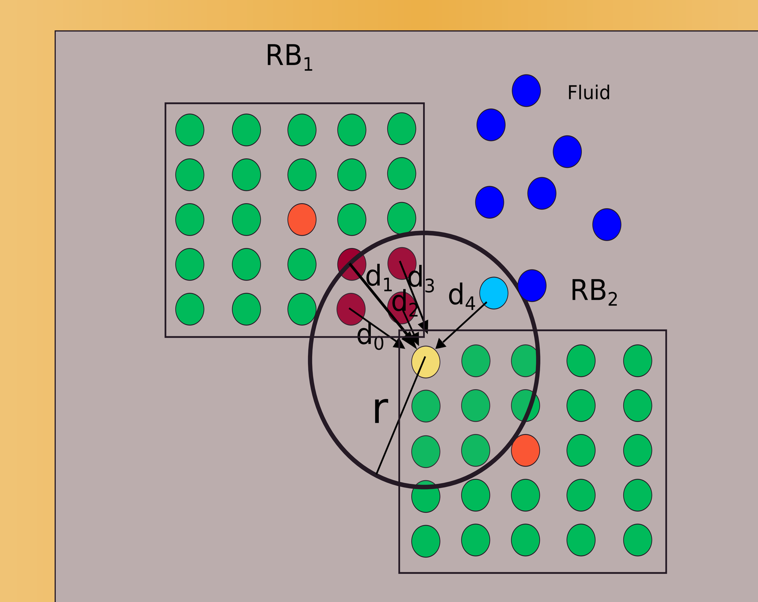
Consider the force between two rigid-body particles. The force is defined as follows:

$$F_s = -k(r - |d_{ij}|)\frac{d_{ij}}{|d_{ij}|}$$

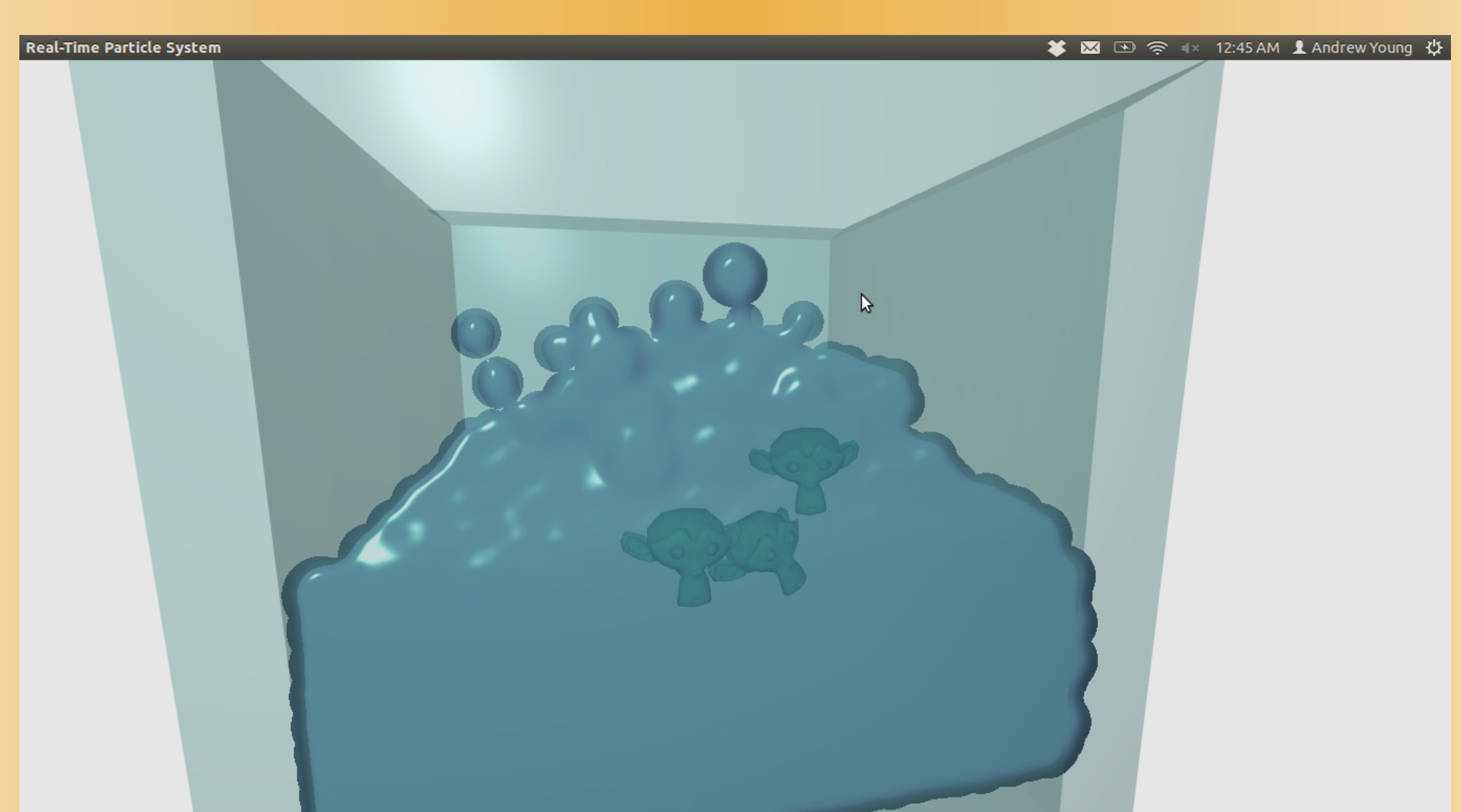
$$F_d = \eta(v_j - v_i)$$

$$F_{i,j} = F_s + F_d$$

In the above equations, k represents the spring coefficient, η represents the dampening coefficient, dij represents the relative distance between particle i and j, and r is the interaction radius. Several other models exist for computing the interaction between two rigid-bodies. However, many of the other models are non-linear and require much more computation. The aim of the framework is real-time thus other methods are not currently considered in this framework. We use the same model for interacting fluid and rigid-body particles.



The yellow particle is influence by 4 particles of RB1 and 1 fluid particle. Note, rigid-body particles do not interact with particles in the same rigid-body.



Suzanne monkey heads being dropped into a tank of water. The monkey causes the water to splash.

Integrate over all the Forces

The final step of the simulation requires integrating the acceleration and velocity to obtain the position of each rigid-body. The framework uses leapfrog as the default integration method. Using leapfrog for integration allows for more consistency between the SPH simulation and the rigid-body simulation. Also the Leapfrog method is more stable than Euler's method for numerical integration.

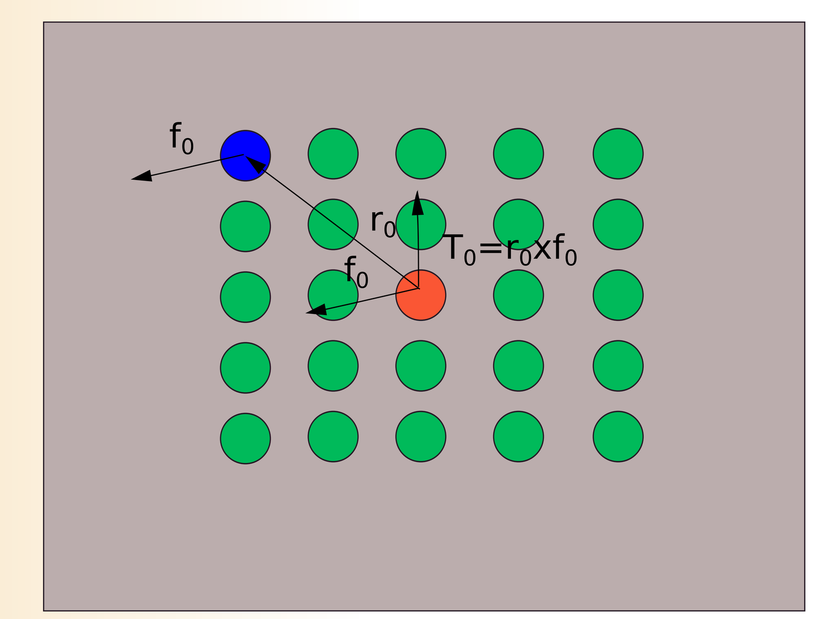
Because each rigid-body is decomposed into a set of particles, one must sum the forces across all particles of a rigid-body before integrating the rigid-body acceleration. A rigid body has two forces, a linear force and an torque force. To compute the linear force on a rigid-body one must sum the forces of all particles which compose that rigid-body. The computation of the torque force requires the sum of cross products between the distance-from-center and the linear force for each particle in the rigid-body.

$$\mathrm{F}_{linear} = \sum_{i=0}^{N} f_i$$

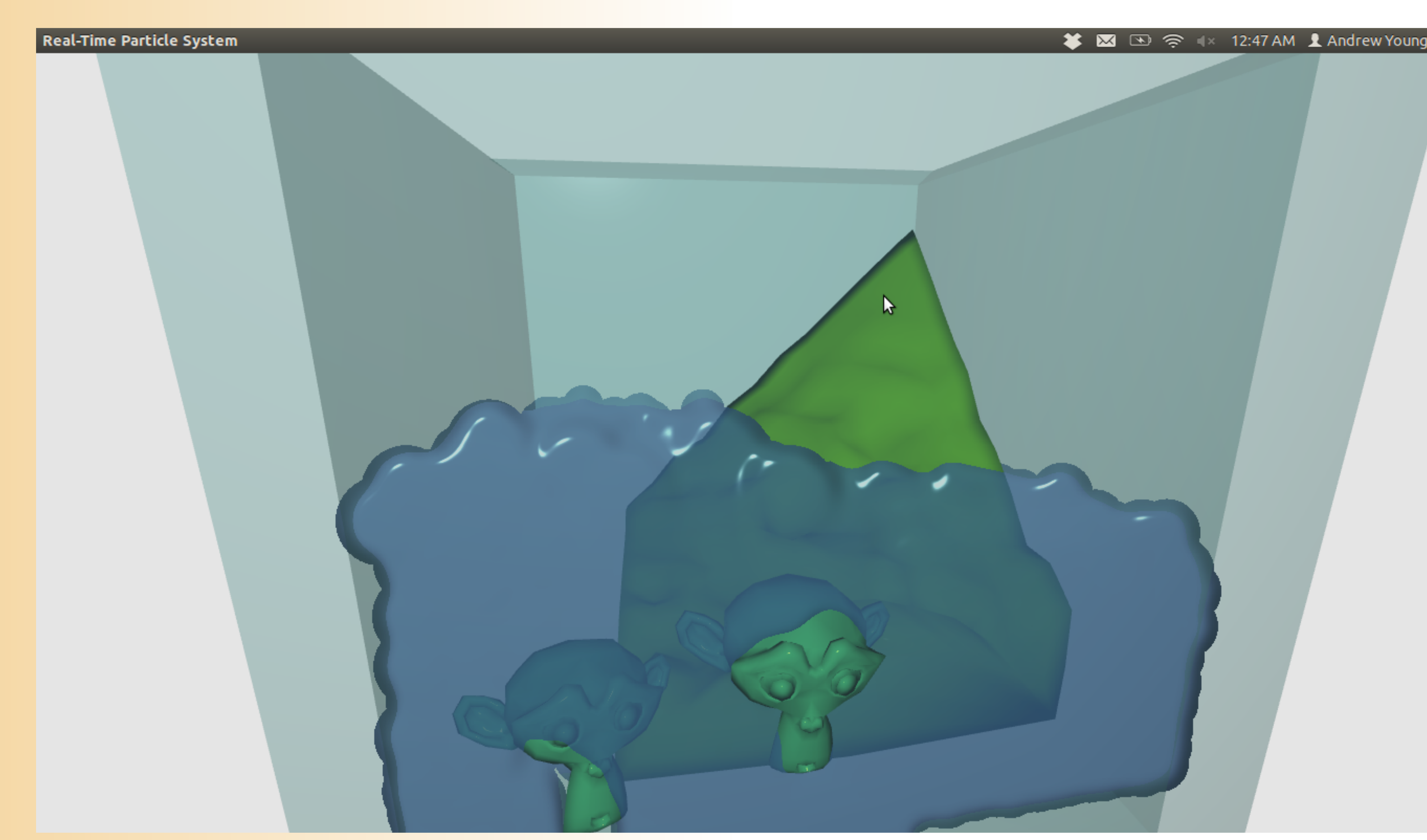
$$\mathrm{F}_{torque} = \sum_{i=0}^{N} r_i \times f_i$$

$$\mathrm{r}_i = pos_i - pos_{com}$$

Integrating the center-of-mass forces yields an updated linear and angular velocity. The velocities are integrated to update the position and rotation of the rigid-body. The final step uses the new position and rotation to update each particle's position.



The force on the blue particle contributes to the center-of-mass linear force and torque forces.



The monkey heads slide down the mountain and push the water up the side of the tank.

Scientific Computing
Florida State University