Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Posters and Demos

Fluid Animation with Explicit Surface Meshes

Tyson Brochu [†] and Robert Bridson [‡]

University of British Columbia, Vancouver, Canada

Abstract

We present a method for fluid surface tracking that represents the surface as a mesh of triangulated points in space, rather than as an implicit surface function. Utilizing well-developed algorithms designed for collision detection in cloth simulation, our system is able to handle topology changes robustly and efficiently. We take advantage of the explicit surface representation to introduce a new approach to simulating surface tension. Finally, we propose a boundary element method for maintaining fluid incompressibility which uses only data points on the fluid surface, rather than a full volumetric discretization of the fluid over a grid.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Physically based modeling; I.3.7 [Computer Graphics]: Animation

---

1. Introduction

Conventional fluid simulation techniques, based on volumetric grids, have severe difficulties in modeling surface tension effects, particularly with thin fluid structures. Ad hoc procedural methods based on blobbies are also inadequate for dealing with thin structures such as sheets. We introduce a new approach to water simulation, targeting surface-tension-dominated scenarios: a boundary-only explicit surface method. By leveraging provably robust collision detection methods and only allowing topological changes which result in collision-free states, we circumvent some of the main difficulties with explicit surface tracking.

2. Explicit Surface Discretization

We discretize each fluid surface as a polygon in 2D, or polyhedron with triangular faces in 3D. We base our method on the robust collision detection and handling treatment of Bridson et al. [BFA02], which provides an efficient algorithm for guaranteeing that fixed-connectivity meshes will never suffer (self-)intersection. We extend this to incorporate the connectivity changes required for fluid simulation (such as mesh adaptation and fluid merging or pinching off), but in a conservative fashion to still guarantee that our explicit surface remains in an intersection-free state (or *legal* state) after each advection step. That is, we handle topological changes selectively, merging and separating fluid only when the resulting surfaces are collision-free, and otherwise sequentially applying repulsion forces, geometric collision impulses and rigid impact forces as needed to resolve the surface collisions, as per cloth simulation (see figure 2).

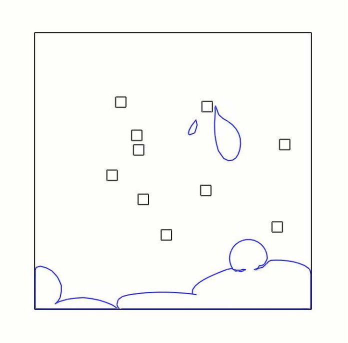

Figure 1: *A snapshot of our 2D simulation showing a fluid interacting with solid obstacles.*

We also adaptively add and remove vertices or flip edges

[†] email: tbrochu@cs.ubc.ca
[‡] email: rbridson@cs.ubc.ca

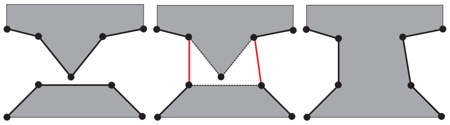

Figure 2: *Proximity event and topology change. The collision detection algorithm is run on hypothetical new faces before merging is performed.*

to maintain a good discretization, again doing so only when the resulting configuration is intersection-free, or legal. That is, we *delay* mesh connectivity changes until we know that they are safe; in practice this is typically only a single timestep due to the powerful cloth collision resolution method.

Vertex addition is straightforward since adding an additional vertex to an edge cannot introduce a collision. Long edges are split by a new vertex, and the edge's incident triangles are each divided in two.

When a vertex is scheduled for deletion (because its neighbourhood is too small in area), we check if the pseudo-motion of the surface induced by moving the vertex to its closest neighbour causes any collisions. We stress that this "motion" is not done with a real time step, but rather with the rest of the geometry held fixed and without advancing time. If the pseudo-motion causes collisions, we do not do anything; otherwise we perform the topological edge contraction, deleting that vertex (see figure 3).

Similarly, we maintain a good aspect ratio for triangular faces by applying a standard edge flip algorithm, only allowing the change if it does not induce collisions. In this case the pseudo-motion can be described by the addition of a point in the middle of the edge, splitting the incident triangles, then moving this single point to lie on the proposed new edge, at which point we can delete it again without changing the geometry. Alternatively put, we are simply detecting if anything interferes with the tetrahedron spanned by the edge's incident triangles.

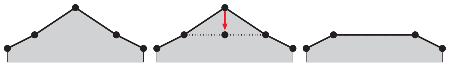

Figure 3: *Vertex deletion in 2D. The vertex pseudo-motion is checked for collisions before adaptivity is performed.*

3. Surface Tension

Rather than the conventional approach based on mean curvature driven flow [WMT05,LGF04], we model surface tension as an *actual tension* per unit length, permitting a more accurate conservative discretization. In two dimensions, we add two forces to each polygon edge, proportional to a surface tension coefficient, parallel to the directions of the two neighboring edges. In three dimensions, we add three forces to a given face, corresponding to all neighboring faces. The force is proportional to the surface tension coefficient times the edge length, in the direction normal to the edge and coplanar to the neighbouring face. We note this exactly conserves the momentum of the volume of fluid, unlike other approaches to surface tension, since these forces are always balanced by the opposite forces on neighbouring faces.

4. Volume Conservation

To achieve conservation of volume, we explicitly track the volume of fluid for each droplet. At each time step, we first add forces, such as gravity and surface tension, to the velocities on the surface. The fluid vertices are then advected with velocities averaged from incident faces, to get a *predicted* configuration, while still keeping track of the initial fluid. We compute the volume of this predicted fluid configuration, and compare it to the volume before advection. We then add a correction impulse to the applied forces in the normal direction to conserve volume, and advect again. This is a simple heuristic for the effect of pressure, plausible for low Reynolds number free-floating droplets.

5. Boundary-Only Fluid Dynamics

To supplant this volume-conservation heuristic, we are in the process of developing a more rigorous, physically valid pressure solve. We establish a harmonic partition of unity of the fluid volume, using the Boundary Element Method, and associate a volume (or mass, assuming an incompressible fluid) with each face. This mass is equal to the integral of the face's harmonic basis function over the volume of fluid. Assuming the internal flow has come to a quasi-static equilibrium with viscosity (appropriate for the small scale, low Reynolds number situations we are interested in) we postulate the mass of each face should remain constant, and calculate pressure as a Lagrange multiplier to enforce this constraint. This of course conserves total volume, i.e. the sum of the face masses. We are also working on a rigorous viscosity update to the boundary velocities, based on estimating and damping the rate of non-rigid deformation of the model.

References

[BFA02] BRIDSON R., FEDKIW R., ANDERSON J.: Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 3 (2002), 594–603.

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. *ACM Trans. Graph. 23*, 3 (2004), 457–462.

[WMT05] WANG H., MUCHA P. J., TURK G.: Water drops on surfaces. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 3 (2005), 921–929.