Cartesian Grid Fluid Simulation with Irregular Boundary Voxels (sketch_273)

Doug Roble* Digital Domain
Nafees bin Zafar† Digital Domain
Henrik Falt‡ Digital Domain

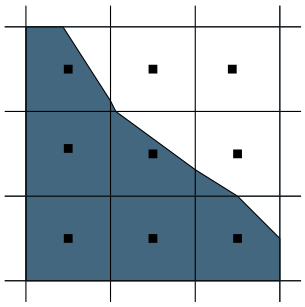

Figure 1

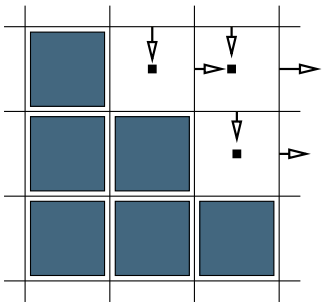

Figure 2

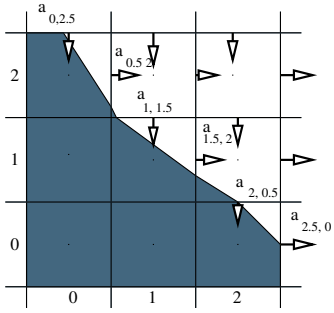

Figure 3

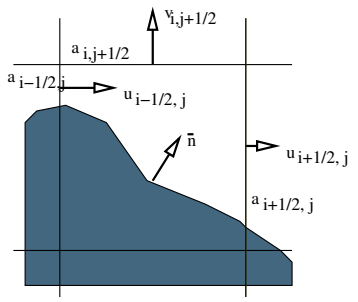

Figure 4

1 Introduction

There are two grid-based techniques for simulating the motion of fluid: finite difference and finite element. Finite element methods fit a deformed grid to the details of the simulation while finite difference methods impose a regular grid on the simulation details. Most computer graphics CFD solutions have been based on finite difference techniques. These are much easier to implement and work with but details are lost in the regular grid discretization.

This sketch presents a modification to a regular grid finite difference method so that voxels on the boundary deform to capture more accurate detail. [Losasso et al. 2004] approached this problem by refining the simulation using an octree data structure for the velocity field and surface representation. [Johansen and Colella 1998] has developed a similar technique to ours, but produced a non-symmetric matrix that they solved with multi-grid techniques.

In a typical finite difference fluid simulation, the boundary and fluid surface representations have more detail than the velocity grid. Figure 1 shows an example of a 2D simulation where a high detail boundary is embedded in a velocity grid. The boundary is discretized by marking voxels as either boundary or non-boundary so that it fits the velocity grid (Figure 2). This introduces grid artifacts to the simulation. It is quite difficult to simulate a gently sloping feature - that would discretize into long plateaus of voxels.

We propose a more accurate approach that still retains the simplicity of a regular grid. Instead of marking a voxel as boundary or non-boundary, we simulate any voxel that has at least some non-boundary area (Figure 3). This requires some changes to the fluid simulation, most notably the divergence calculations.

2 Simulation Changes

- First, instead of just marking a voxel as boundary or non-boundary, we compute the non-boundary area of each of the faces of a voxel. If a voxel face is completely covered with boundary, its area is 0.0, if completely open the area is 1.0. If the boundary is represented as a level set, this can be done using a technique similar to marching cubes.
- The divergence must be calculated differently. The key is to notice that the divergence equation $\nabla \cdot \bar{u} = 0$ forces the mass flux across each face to balance. And since we are simulating incompressible fluid, the density of the fluid can be ignored. So, for a 2D example, this must hold:

  $a_{i+1/2,j}u_{i+1/2,j} - a_{i-1/2,j}u_{i-1/2,j} + a_{i,j+1/2}v_{i,j+1/2} - a_{i,j-1/2}v_{i,j-1/2} = 0$

  where $a_{i+1/2,j}$ is the open area of a face and $u_{i+1/2,j}$ is the velocity component normal to the face. This form of the divergence can be derived using the Divergence Theorem. [Losasso et al. 2004] explain this in detail and arrive at a linear set of equations:

  $\sum_{faces}((\Delta t \nabla p)_{face} \cdot \bar{n})A_{face} = \sum_{faces}(\bar{u}^*_{face} \cdot \bar{n})A_{face}$
- Solving the above equations for p and correcting the velocities produces divergence free velocities - as long as the areas are taken into account. Interpolation of the velocity at a point in the staggered grid must be aware of the area or mass loss/gain will occur. Figure 4 illustrates some of the details needed in a typical boundary voxel. Velocities are pushed to the center of the non-boundary area. If the velocity component of a face is requested, the true position of the velocity is used. Dirichlet boundary conditions ($\bar{u} \cdot \bar{n} = 0$) are enforced using the true boundary, not the discretized boundary.

These changes to finite difference fluid simulation now track fluid close to boundaries much more accurately. This reduces volume loss and the griding artifacts in the fluid.

References

JOHANSEN, H., AND COLELLA, P. 1998. A cartesian grid embedded boundary method for poisson's equation on irregular domains. *Journal of Comp. Physics*, 147, 60–85.

LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. 23*, 3, 457–462.

*e-mail: doug@d2.com
†e-mail:nafees@d2.com
‡e-mail:hfalt@imageworks.com