OPENVDB ADOPTION

JOHN JOHANSSON
SOFTWARE ENGINEER

SIGGRAPH 2013
• Introduction
• Fluid simulation
  • Workflow
  • Implementation
• Cloud advection
• Volume modeling
• Cyclone
  • FX toolkit
  • Fluid simulation
  • Volume modeling
• Developed for three years
• Used in production the last year
  • Ender’s Game
  • Black Sky
  • Maleficent
  • Commercials
CYCLONE

- Specifically designed for sparse grids
  - Initially used DB-Grid [Museth, 2009]
  - Blocked data structure
  - Algorithms implemented to operate on blocks
  - Well suited for threading
  - Straightforward to replace with OpenVDB
- Integrated in Houdini
FLUID SIMULATION

- FLIP Solver
  - Hybrid particle/grid solver
- Proprietary particle representation
  - Independent of third party package
  - Distributed simulations
- OpenVDB grid representation
  - Fluid surface (Level Set)
  - Fluid velocities (Staggered Vector grid)
  - Boundary objects, emitters, forces, sinks
  - Particle search acceleration

<table>
<thead>
<tr>
<th>Particle Index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Velocity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Radius</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
WORKFLOW

- SOP workflow preferred by artists
- Particle data passed as shared pointers
- Custom looping mechanism for simulations
- OTLs for fluid boundaries, forces etc
WORKFLOW
WORKFLOW

Additional SOPs

• Delete particles based on Level Set
• Advect particles and grids
• Surface particles
PARTICLE CREATION

- Use Level Set as guide
- Resample Level Set to sampling interval
- Create bool mask of inside region
- For each active voxel of mask
  - Subdivide voxel
  - Jitter particle position in sub voxel
  - Subdivision could be varied based on distance to surface
PARTICLE CREATION
PARTICLE GRID

- Accelerate particle search
  - KD-Tree, Hash table, Uniform grid
- Approach similar to [Nielsen et al., 2007]
- Extend Leaf Node with particle index array
- Store a Particle Bin per voxel
  - Two unsigned integers
  - Offset into index array
  - Number of particles
PARTICLE GRID

Build Grid

- For each particle
  - Compute voxel coordinate
  - Activate voxel
  - Increment particle counter for voxel
  - Store particle index in array
  - Store linear voxel offset in external array

- For each Leaf Node
  - Sort index array based on voxel offset
  - For each active voxel
    - Update offset into index array
INTERPOLATION

- Interpolate particle attribute to grid
  - Velocity, density, viscosity etc.
- Weight value based on distance and kernel
  - Initialize Particle Grid and copy topology
  - Dilate topology based on weight radius
  - For each active voxel
    - Find particles in neighborhood
    - Compute weighted value
SURFACING

• Build Level Set from particles
  • Distance to closest particle
  • Weighted average of position and radius [Zhu and Bridson, 2005]
• Only interested in particles near the surface

• Initialize Particle Grid and copy topology
• Dilate and erode topology by weight radius
• Compute morphological difference and use to allocate grid
• For each active voxel
  • Find particles in neighborhood
  • Compute signed distance value
• Signed Flood Fill
SURFACING

Cyclone Surfacer (Closest distance)

OpenVDB From Particles

18.3 Million particles, Surface grid 2x resolution, No post processing
SURFACING

Cyclone Surfacer (Weighted average)

OpenVDB From Particles

18.3 Million particles, Surface grid 2x resolution, No post processing
Problem:
- Animate clouds with rotational motion
- Advect density grid in forcefield
- Model forcefield using animated points
  - Interpolate point velocities to grid
- Semi-Lagrangian advection
  - Too much numerical dissipation
- Particle based density advection
  [Wrenninge et al., 2011]
- Built advection tool as OTL using Cyclone and OpenVDB nodes
PARTICLE ADVECTION

- Compute Level Set from Cloud density
- Initialize particles based on Level Set
- Interpolate density from grid
- For each frame
  - Advect particles
  - Reseed particles if needed
  - Interpolated density to grid
ADVECTION COMPARISON

Semi-Lagrangian advection

Particle advection
ON THE HORIZON

- Volume modeling
- Storm + OpenVDB
- Improve workflow
- Points, Splines, Surfaces
ACKNOWLEDGEMENTS

- Software
  - Doug Roble, Craig Zerouni, Fredrik Salomonsson, Kree Cole-McLaughlin, Blake Sloan
- FX TDs and artists
  - Dennis Blakey, Brian Gazdik, Eddie Smith, James Atkinson, Daniel Stern, Charles-Felix Chabert, Thomas Reppen, Karl Kohlman
- OpenVDB team

