Physically-Based Simulation
Position Based Fluids

Group 17
Mengdi, Memedi, Danielello
Demo

27K Particles ~ 30 fps
Demo

27K Particles ~ 30 fps

42K Particles ~ 15 fps

125K Particles ~ 4 fps
Final State

• PBF Simulation
  – Parallelized advance (CPU)
  – Fast parallelized neighborhood search
  – Moving Boundaries

• Rendering
  – Instanced rendering
  – Dynamic particle coloring
Implementation Details

Multithreading

- Standard std::Threads
- Synchronized through barriers
- Responsible for one chunk of particles
- No pool or shutdown
Implementation Details

Neighborhood search

- Atomic increase
- Prefix Sum sequential

Slide from “Fast Fixed-Radius Nearest Neighbors: Interactive Million-Particle Fluids – Rama C. Hoetzlein, Graphics Devtech, Nvidia” presentation in 2014
Implementation Details

Simulation Boundary

- Simple AABB
- Boundary Particles
- Best Combined
Implementation Details

SPH Kernel Functions

- Expensive
- Precompute
- Replaced by LUT

\[
W_{\text{poly6}}(r, h) = \frac{315}{64\pi h^9} (h^2 - |r|^2)^3
\]

\[
\nabla W_{\text{spiky}}(r, h) = \frac{45}{\pi h^6} (h - |r|)^2 \frac{r}{|r|}
\]

M. Macklin – Position Based Fluids, Presentation Slides for SIGGRAPH2013.
Implementation Details

Dynamic Particle Coloring

- Trade off between realism and real-time performance
- Float color gradient from blue to white [0, 1]
- Weighted mixture of particle velocity and height
- Bright wave tops, darker deep water
Lessons Learned

• Libigl / Geometry Course Simulator Framework
  - Easy to start, hard to change
  - Rewrite time consuming

• Position Based Fluids
  – Simplifications don’t make it simple to get right

• Realtime is really difficult to achieve
  – any computation is too much
Timeline

- Constraint Solver: 15/11
- Simulation steps: 24/11
- Improved Shading: 15/12
- Scenes: 03/11
- Optimizations / GPU port: 10/12

✓: Completed
✗: Not completed