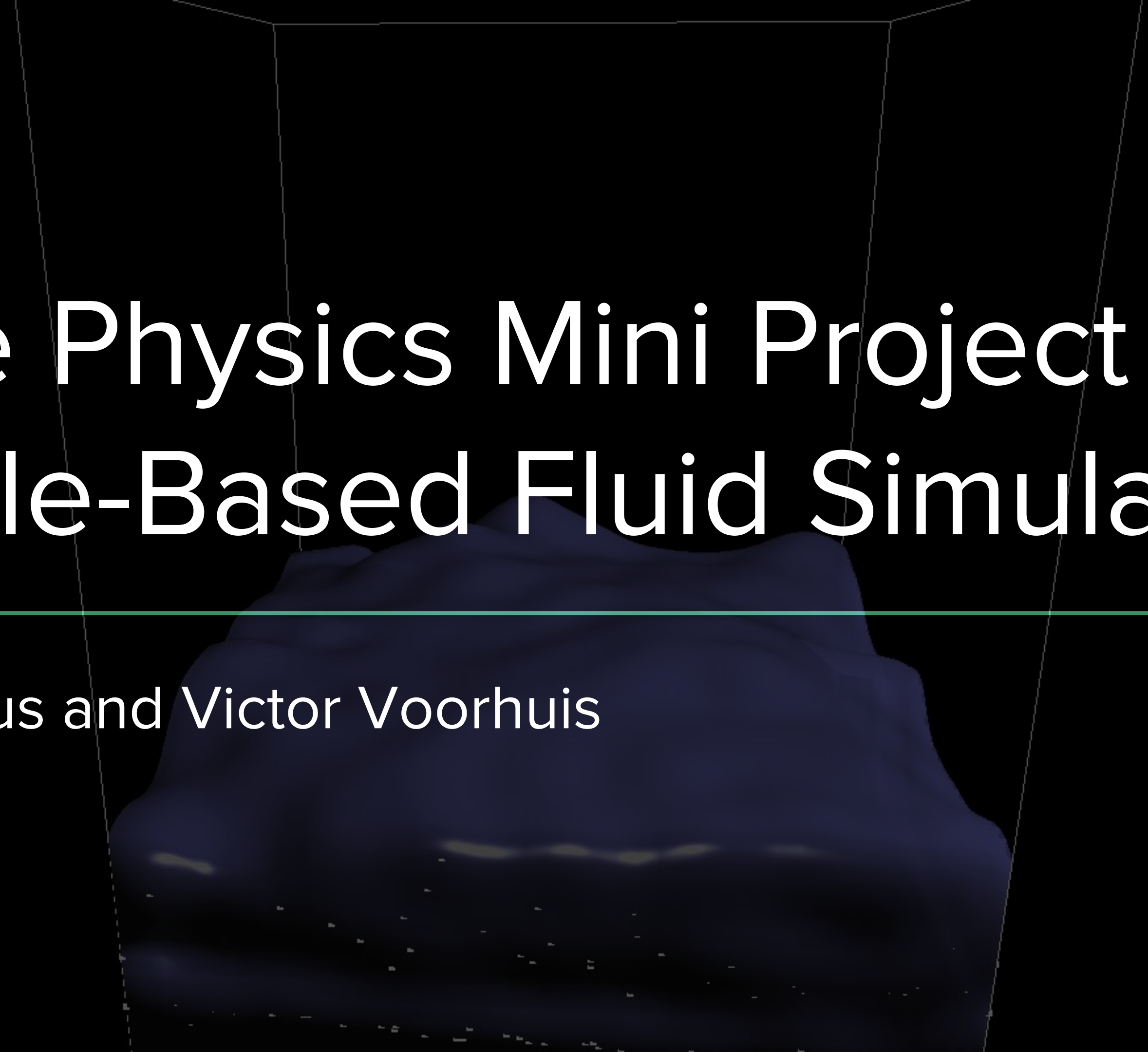
Game Physics Mini Project
Particle-Based Fluid Simulation

Toby Rufinus and Victor Voorhuis



Modeling fluids with particles

- *"Particle-Based Fluid Simulation for Interactive Applications"*
  - Matthias Müller, David Charypar and Markus Gross

  (Equations taken from this paper)

- Smoothed Particle Hydrodynamics

- Lagrangian approach

Eurographics/SIGGRAPH Symposium on Computer Animation (2003)
D. Breen, M. Lin (Editors)

Particle-Based Fluid Simulation for Interactive Applications

Matthias Müller, David Charypar and Markus Gross

Department of Computer Science, Federal Institute of Technology Zürich (ETHZ), Switzerland

Abstract

Realistically animated fluids can add substantial realism to interactive applications such as virtual surgery simulators or computer games. In this paper we propose an interactive method based on Smoothed Particle Hydrodynamics (SPH) to simulate fluids with free surfaces. The method is an extension of the SPH-based technique by Desbrun to animate highly deformable bodies. We gear the method towards fluid simulation by deriving the force density fields directly from the Navier-Stokes equation and by adding a term to model surface tension effects. In contrast to Eulerian grid-based approaches, the particle-based approach makes mass conservation equations and convection terms dispensable which reduces the complexity of the simulation. In addition, the particles can directly be used to render the surface of the fluid. We propose methods to track and visualize the free surface using point splatting and marching cubes-based surface reconstruction. Our animation method is fast enough to be used in interactive systems and to allow for user interaction with models consisting of up to 5000 particles.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

1.1. Motivation

Fluids (i.e. liquids and gases) play an important role in every day life. Examples for fluid phenomena are wind, weather, ocean waves, waves induced by ships or simply pouring of a glass of water. As simple and ordinary these phenomena may seem, as complex and difficult it is to simulate them. Even though Computational Fluid Dynamics (CFD) is a well established research area with a long history, there are still

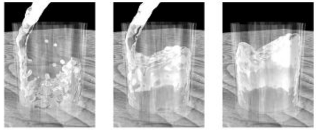


Figure 1: *Pouring water into a glass at 5 frames per second.*

Smoothed Particle Hydrodynamics (SPH)

- Field quantities defined at particle locations

- Evaluate quantities anywhere in space

$$A_S(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h)$$

- m_j: mass
- ρ_j: pressure
- $W(\mathbf{r}, h)$: smoothing kernel with radius h

- Calculate density at **r**: $\rho_S(\mathbf{r}) = \sum_j m_j \frac{\rho_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) = \sum_j m_j W(\mathbf{r} - \mathbf{r}_j, h)$

Forces working on a particle

$$\mathbf{f}_i^{\text{pressure}} = -\sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W(\mathbf{r}_i - \mathbf{r}_j, h)$$

$$\mathbf{f}_i^{\text{viscosity}} = \mu \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h)$$

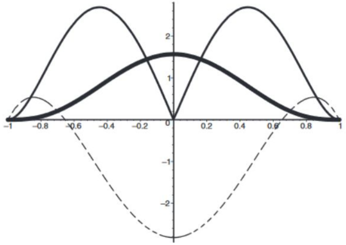
$$\mathbf{f}^{\text{surface}} = \sigma \kappa \mathbf{n} = -\sigma \nabla^2 c_S \frac{\mathbf{n}}{|\mathbf{n}|}$$

$$\left(\begin{array}{l} c_S(\mathbf{r}) = \sum_j m_j \frac{1}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) \\ \mathbf{n} = \nabla c_s \end{array} \right)$$

Smoothing kernels

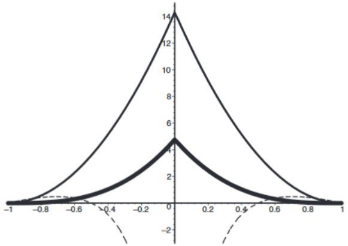
Most cases:

$$W_{\text{poly6}}(\mathbf{r},h) = \frac{315}{64\pi h^9}\begin{cases}(h^2-r^2)^3 & 0\le r\le h\\ 0 & \text{otherwise}\end{cases}$$



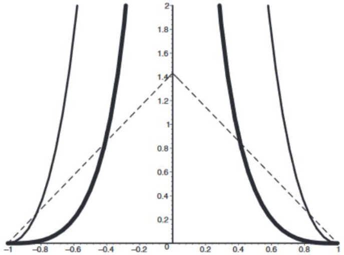
Repulsion (prevents vanishing gradient near distance 0):

$$W_{\text{spiky}}(\mathbf{r},h) = \frac{15}{\pi h^6}\begin{cases}(h-r)^3 & 0\le r\le h\\ 0 & \text{otherwise,}\end{cases}$$



Viscosity (prevents negative Laplacian):

$$W_{\text{viscosity}}(\mathbf{r},h) = \frac{15}{2\pi h^3}\begin{cases}-\frac{r^3}{2h^3}+\frac{r^2}{h^2}+\frac{h}{2r}-1 & 0\le r\le h\\ 0 & \text{otherwise.}\end{cases}$$



Implementation: Used Tools

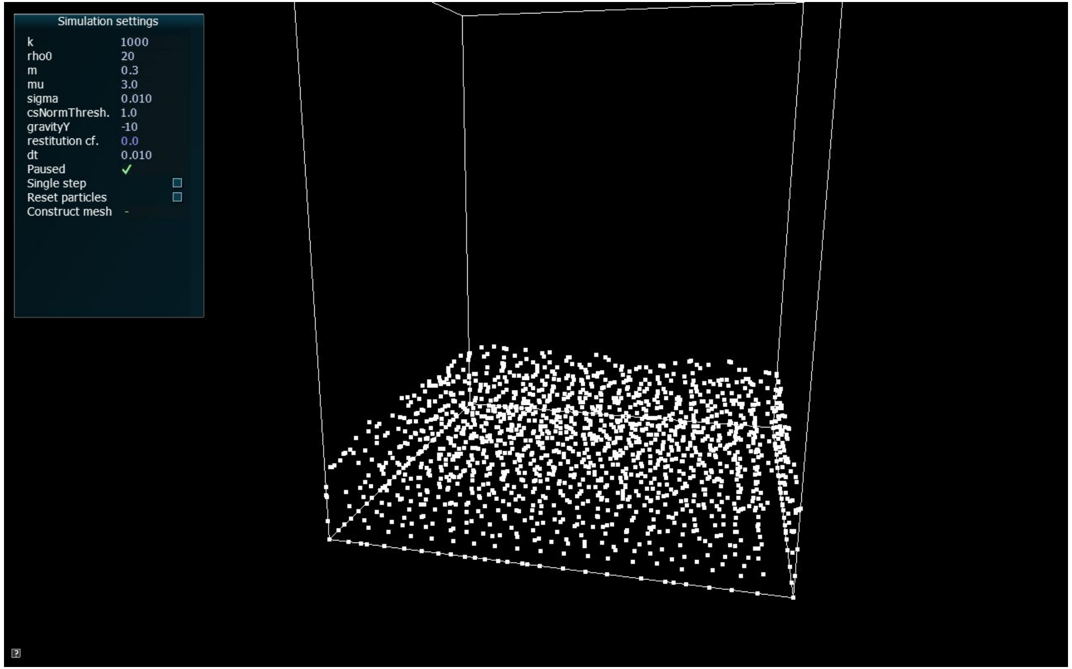
- OpenGL - GLEW, GLFW
- GLM
- AntTweakBar
- OpenMP
- PolyVox
- tinyobjloader

Mesh construction using Marching Cubes

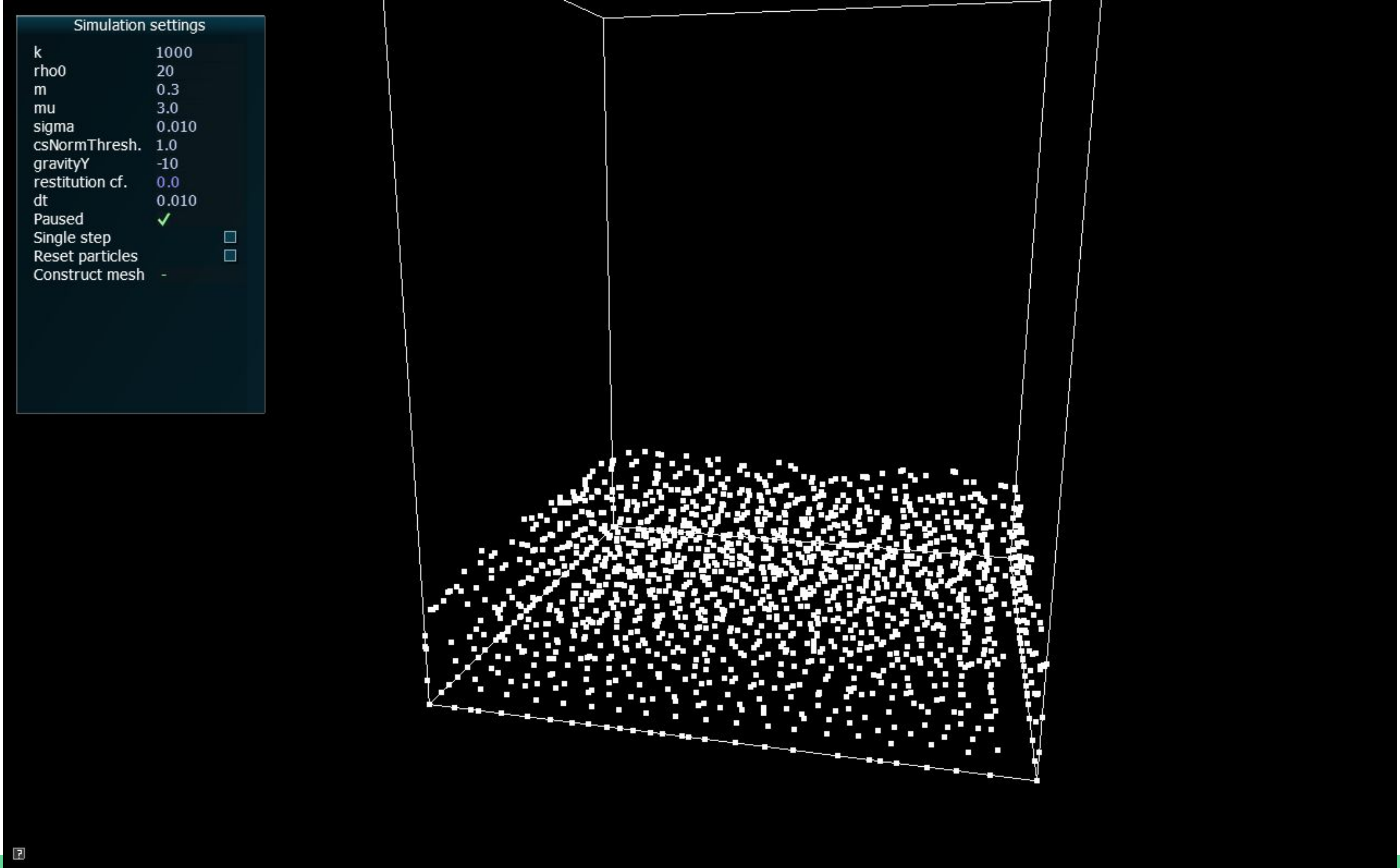
- Use voxel grid
- 61 * 51 * 51 = 158661 voxels

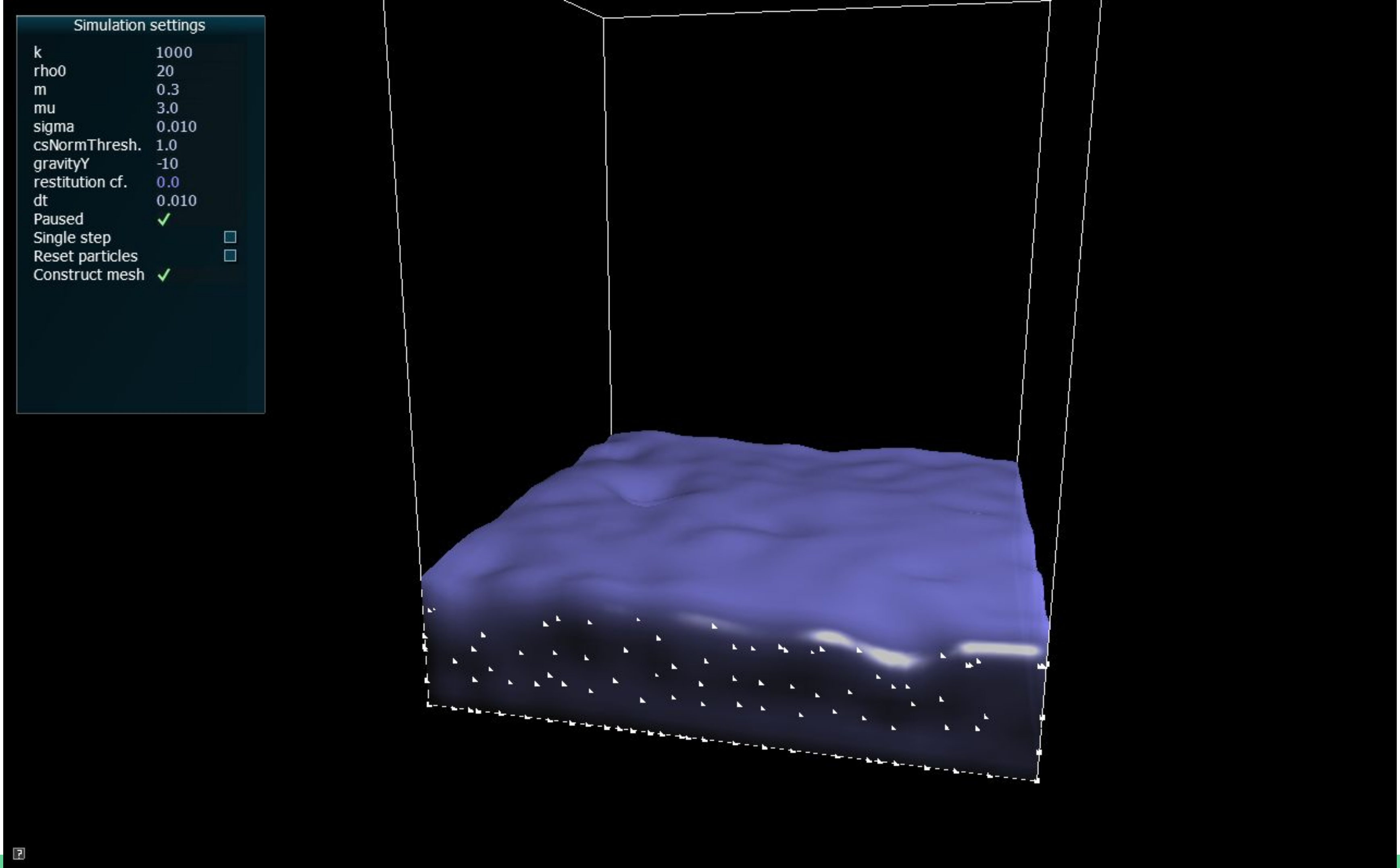
- Calculate density at each voxel
- Density above threshold: solid voxel

- PolyVox library: Marching Cubes



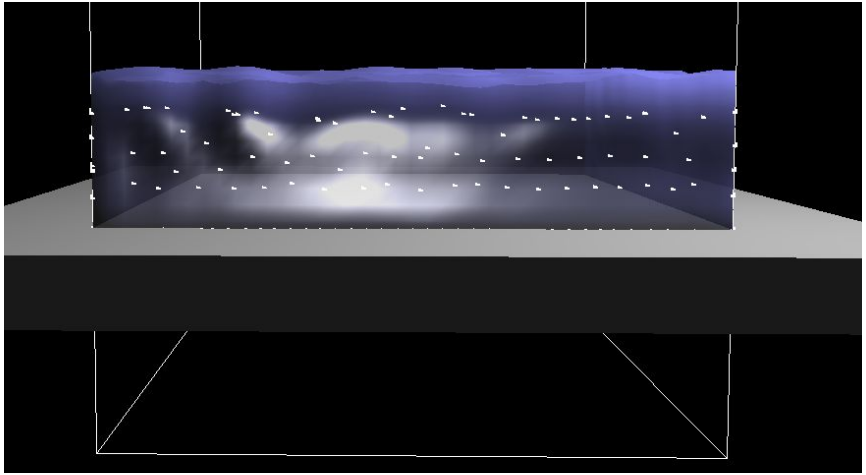
- Performance: multithreading, SIMD, lookup table

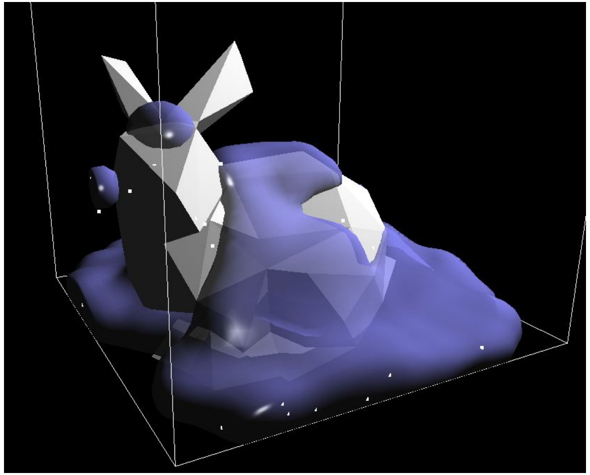

Simulation settings
k 1000
rho0 20
m 0.3
mu 3.0
sigma 0.010
csNormThresh. 1.0
gravityY -10
restitution cf. 0.0
dt 0.010
Paused
Single step
Reset particles
Construct mesh -


Simulation settings
k 1000
rho0 20
m 0.3
mu 3.0
sigma 0.010
csNormThresh. 1.0
gravityY -10
restitution cf. 0.0
dt 0.010
Paused
Single step
Reset particles
Construct mesh

Mesh-particle collisions

- Test intersections: does line segment movement intersect triangle?
- For closest intersection: linear projection, apply impulse
- “Intersection method” described in “*Smoothed Particle Hydrodynamics in Flood Simulations*” by Michal Chládek and Roman Durikovic





Demo