Screen space fluid rendering with curvature flow

Bram Musters Maarten Terpstra

March 4, 2015

Screen space fluid rendering

Screen space fluid rendering is performed using Smoothed Particle Hydrodynamics (SPH)

Curvature flow

Curvature flow is BLA

Who, where and when?

- Wladimir J. van der Laan et al.
- NVIDIA
- Rijksuniversiteit Groningen
- 2009

Why?

- Games
- No

Related work

Overview of method

1. Surface depth is written to render target
2. Surface depth is smoothed
3. Thickness is written to second render target
4. Dynamic noise texture is generated on the surface of the fluid
5. Smoothed surface depth, noise texture and an image of the scene behind the fluid is combined into the final rendering fluid

Surface depth

- Particles rendered as spheres
- Native depth test
- No explicit splatting

Surface depth smoothing

- Remove jelly-like appearance
- Gaussian blur is unsuitable
- Curvature flow

Curvature flow

- Minimize curvature
- Transform the surface along its normal direction
- **Screen-space**: transform the surface along the z direction
- Amount of transformation depends on the magnitude of the mean curvature

$$\frac{\partial z}{\partial t} = H \tag{1}$$

$$H = \frac{\nabla \cdot \hat{n}}{2} \tag{2}$$

Thickness

- Objects become less visible if water is in front of it
- Attenuate colour of object based on "thickness"

$$T(x,y) = \sum_{i=0}^{n} d\left(\frac{x - x_i}{\sigma_i}, \frac{y - y_i}{\sigma_i}\right) \tag{3}$$

- Additive blending based on this measure

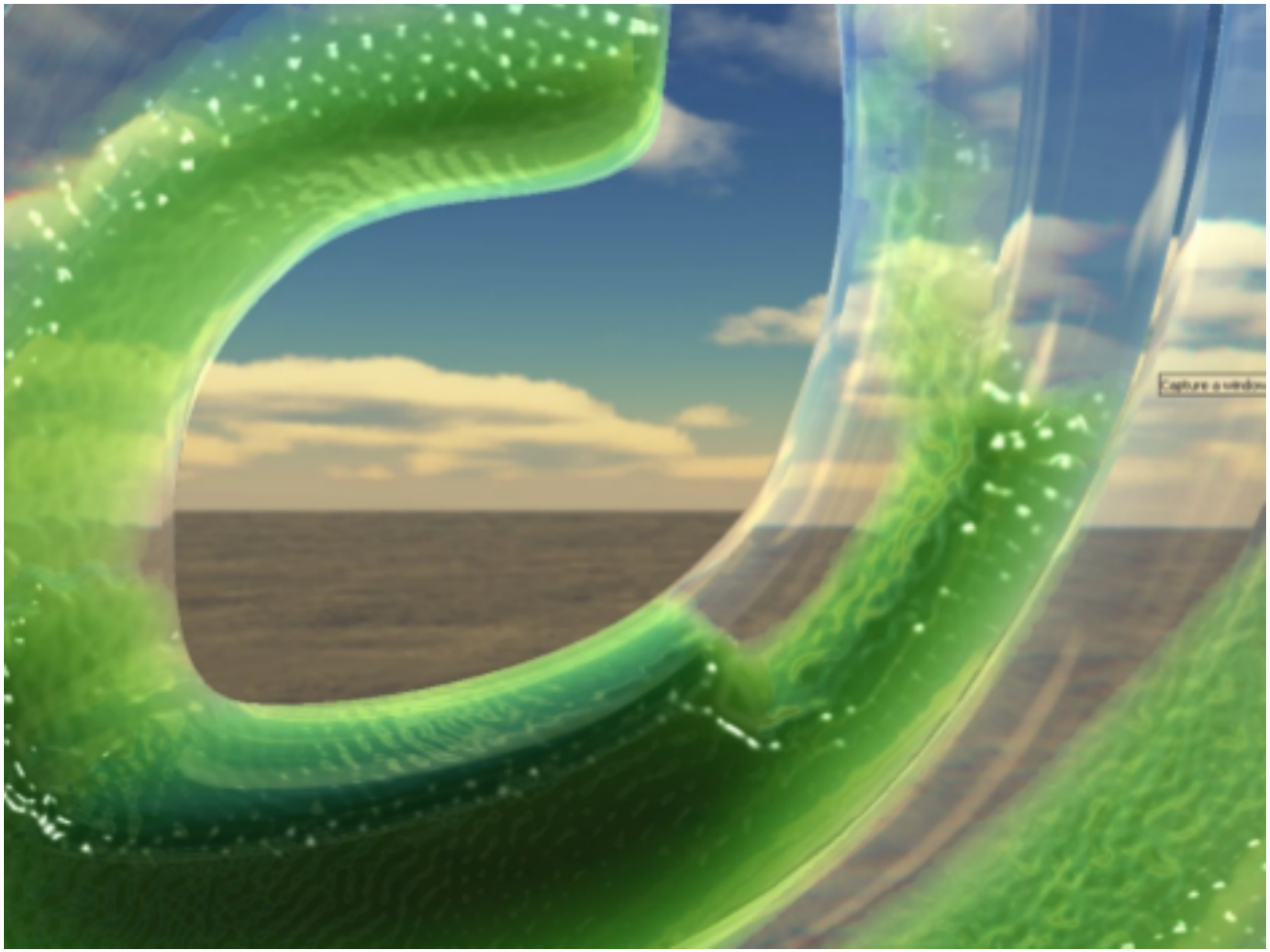
Noise

- Smooth surfaces are unrealistic
- Add noise to surface to add detail
- Assign Perlin noise to each particle
- Make them contribute less as they submerge

Results

- 64000 particles
- 2-3 times slower than Gaussian smoothing
- Much better results

Results



Results