Advanced visualisation for SPH models

O. Garcia-Feal, J.M. Domínguez, A.J.C Crespo, A. Barreiro, M. Gómez-Gesteira, R. Canelas, C. Altomare

December 4 2015

EPHYTECH · EPhysLab
IberianSPH - 2015

Table of Contents

Introduction

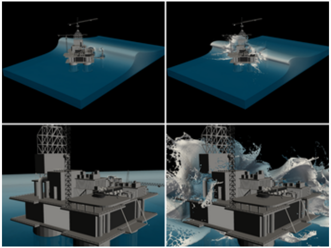
Why is visualisation important?

Visualisation is a communication tool to transmit achievements

Why is visualisation important?

Normal people don't understand about SPH.

Why is the left one better than the right one?



How can the visualization be improved?

Bring SPH simulations closer to Visual Effects Industry.

Simplified SPH models have been heavily used in many recent films.

How can the visualization be improved?

Scientists are -usually- not 3D artists.

We need suitable tools to easily produce high quality animations.

Main goals

Main goals

- Increase the impact on scientific publications and meetings.
- Improve the appearance of project presentations.
- Explore the suitability of DualSPHysics for graphical applications.

Blender

- Blender is a multi-platform open source 3D creation software.
- Includes an advanced path-tracing render engine.
- Easily customizable via Python.

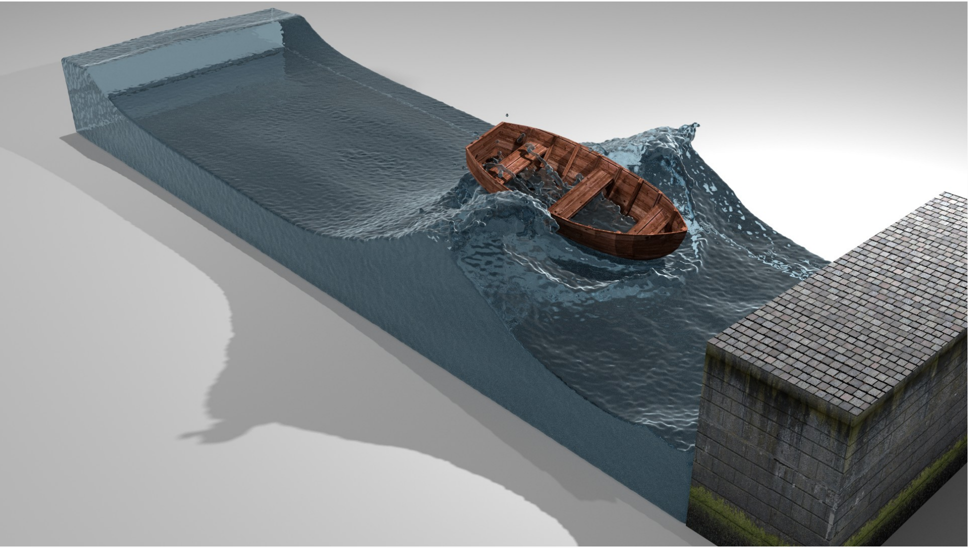
- Evolutionary Prototyping
- A first prototype is built with the initial requirements.
- This prototype will be improved, adding features and modifications as the requirements are refined.

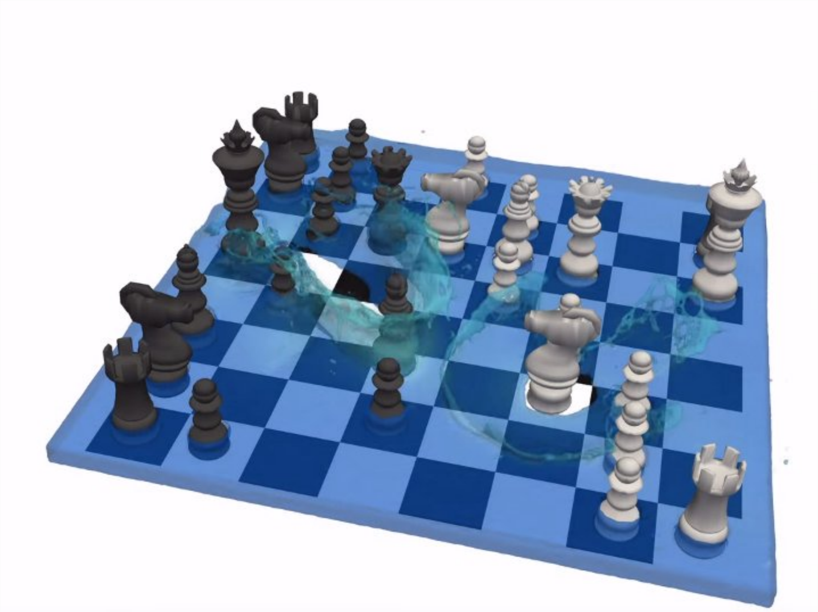
Development Stages

Initial requirements

- Load and update DualSPHysics data into Blender.
- Realistic lighting.
- Texturing support.
- Camera motion.

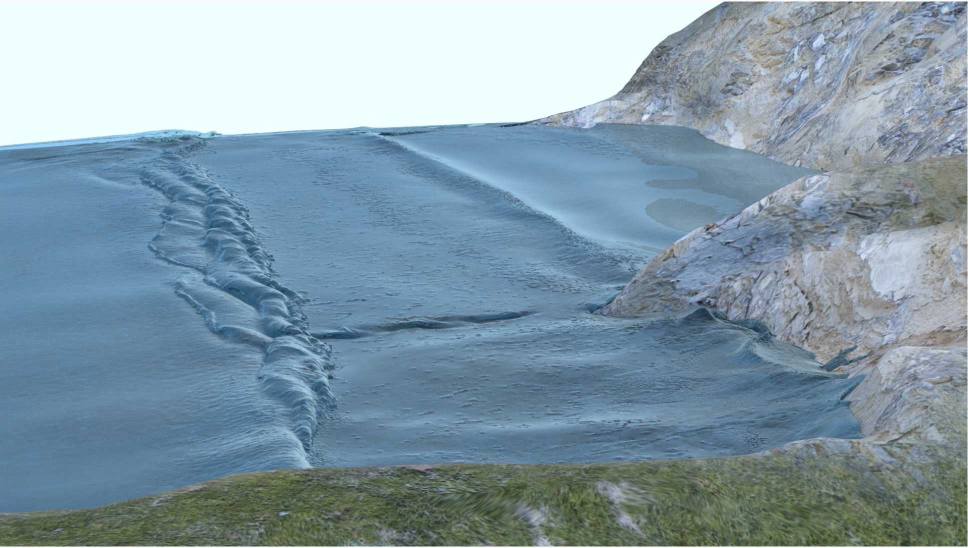
Floating boat simulation





New Chess simulation

Coastal simulation



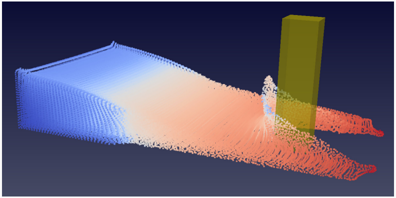
Natural water flows produce foam. This phenomenom occur due to air entrance into the fluid.



Foam simulation

- The method of [Ihmsen et al. 2012] was used.
- Allows to simulate foam effects in post-processing.
- Produces foam, spray and buble particles (from now diffuse particles) that behave different depending on their position.
- This method looks for fluid particles that can generate diffuse particles.

Strong speed differences between particles can originate foam.



To detect this diferences the scaled velocity difference is used:

$$\upsilon_i^{diff} = \sum_j \|v_{ij}\|(1 - \hat{v}_{ij} \cdot \hat{x}_{ij}) W(x_{ij}, h)$$

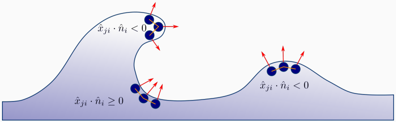
$$\hat{v}_{ij} = \frac{v_i - v_j}{\|v_i - v_j\|}$$

$$\hat{x}_{ij} = \frac{x_i - x_j}{\|x_i - x_j\|}$$

$$W(x_{ij}, h) = \begin{cases} 1 - \frac{\|x_{ij}\|}{h} & \|x_{ij}\| \leq h \\ 0 & \text{otherwise} \end{cases}$$

Foam detection: wave crests

Instabilities can occur on wave crests and this can lead to an air entrance.



Particles on locally convex surfaces should be detected.

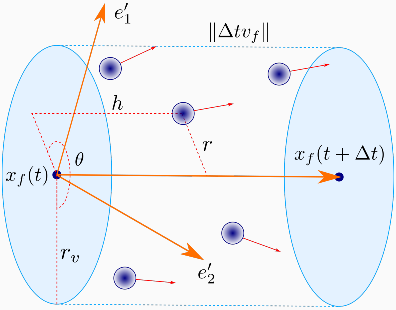
$$\kappa_i = \sum_j \kappa_{ij} = \sum_j (1 - \hat{n}_i \cdot \hat{n}_j) W(x_{ij}, h)$$

$$\tilde{\kappa}_i = \sum_j \tilde{\kappa}_{ij}$$

$$\tilde{\kappa}_{ij} = \begin{cases} 0 & \hat{x}_{ji} \cdot \hat{n}_i \geq 0 \\ \kappa_{ij} & \hat{x}_{ji} \cdot \hat{n}_i < 0 \end{cases}$$

$$\delta_i^{vn} = \begin{cases} 0 & \hat{v}_i \cdot \hat{n}_i < 0.6 \\ 1 & \hat{v}_i \cdot \hat{n}_i \geq 0.6 \end{cases}$$

Generation



Kinetic energy:

$$E_{k,i} = 0.5 m_i v_i^2$$

Clamping function:

$$\phi(I, \tau^{min}, \tau^{max}) = \frac{\min(I,\tau^{max}) - \min(I,\tau^{min})}{\tau^{max} - \tau^{min}}$$

Number of diffuse particles generated by a fluid particle:

$$n_d = I_k (k_{ta} I_{ta} + k_{wc} I_{wc}) \Delta t$$

Spray:

$$v_{spray}(t+\Delta t) = v_{spray}(t) + \Delta t\dot{g}$$

$$x_{spray}(t+\Delta t) = x_{spray}(t) + \Delta t\dot{v}_{spray}(t+\Delta t)$$

Foam:

$$\tilde{v}_f(x_d, t+\Delta t) = \frac{\sum_f v_f(t)W(x_d(t) - x_f(t), h)}{\sum_f W(x_d(t) - x_f(t), h)}$$

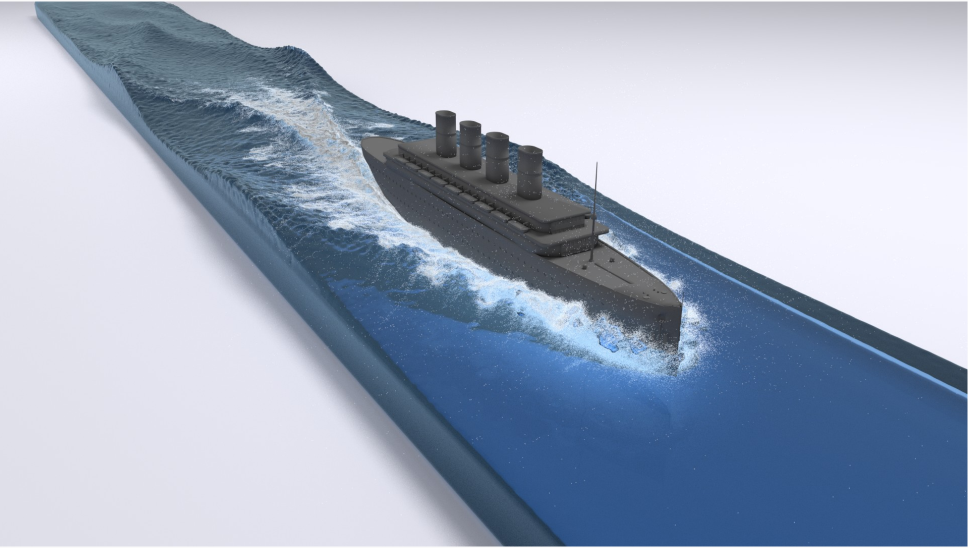
$$x_{foam}(t+\Delta t) = x_{foam}(t) + \Delta t\tilde{v}_f(x_d, t+\Delta t)$$

Bubbles:

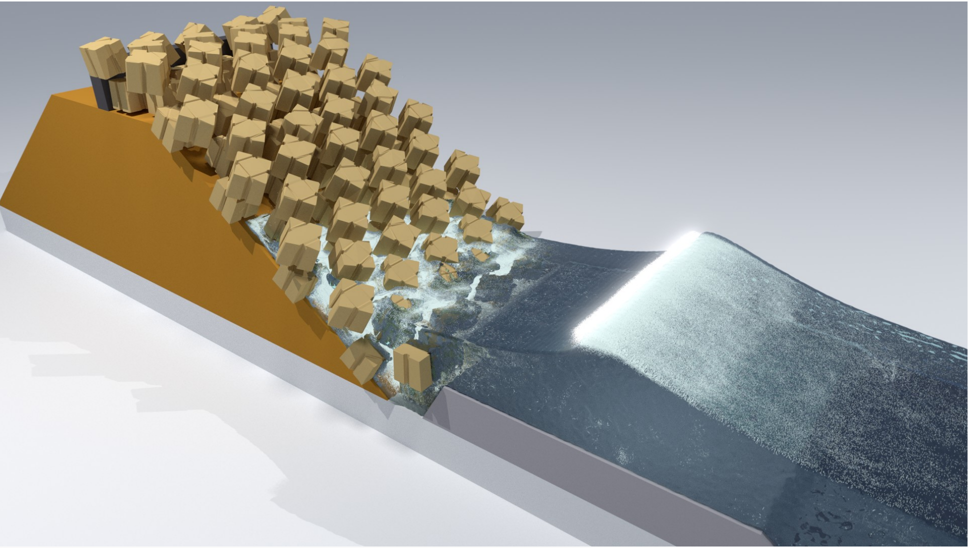
$$v_{bub}(t+\Delta t) = v_{bub}(t) + \Delta t\left(-k_b g + k_d \frac{\tilde{v}_f(x_d, t+\Delta t) - v_{bub}(t)}{\Delta t}\right)$$

$$x_{bub}(t+\Delta t) = x_{bub}(t) + \Delta t v_{bub}$$

Cruise simulation



Wave breaker simulation

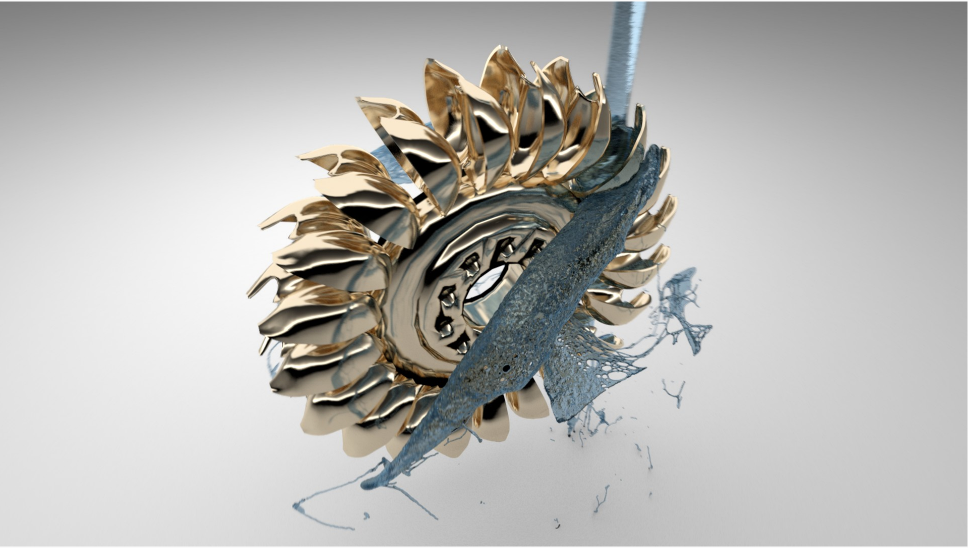


Optimize process

- Parallelize foam simulation with OpenMP.
- Develop a fast C++ library for Blender to load VTK files directly.

Office simulation

Motion Blur



Modify Python scripts?

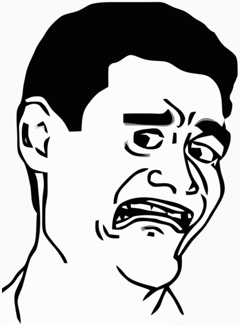
File Edit Options Buffers Tools Conf Help

```
sobj.ropeObject = False
sobj.foamObject = False
sim_objects.append(sobj)

#Fluid
sobj = SimObject()
sobj.pathName = os.getenv("HOME") + "/scratch/diffuse_out/"
sobj.fileBaseName = "Office_"
sobj.fileExtension = ".vtk"
sobj.startFrame = 1
sobj.endFrame = 200
sobj.zeroPadding = 4
sobj.objectName = "foam"
sobj.smoothShading = False
sobj.validateMesh = True
sobj.transferUVMaps = False
sobj.ropeObject = False
sobj.foamObject = True
sim_objects.append(sobj)

# Create the simulated objects if they are not in the scene

for i in sim_objects:
    try:
```

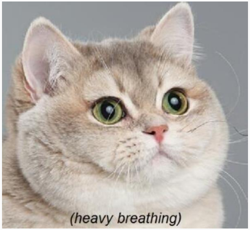
U:--- blender_script.py 95% (380,0) (Python AC)

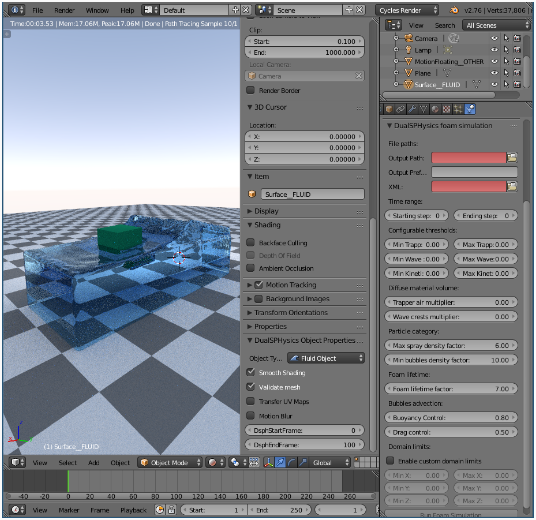
```
[TIMESTEPS]
StartingTimeStep = 1
EndingTimeStep = 100

[FOAMPARAMETERS]

# Clamp function thresholds

MinTrappedAirThreshold = 1.
MaxTrappedAirThreshold = 8.

MinWaveCrestsThreshold = 0.015
MaxWaveCrestsThreshold = 2.

MinKineticEnergyThreshold = 1.5
MaxKineticPowerThreshold = 3.5

# Amount of diffuse material created by trapper air
DiffuseTrappedAirMultiplier = 70.0

# Amount of diffuse material created by wave crests
DiffuseWaveCrestsMultiplier = 70.0
```

-:--- ejemplo.ini 43% (51,0) (Conf[Unix])



User-friendly GUI





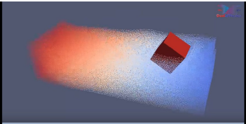
Natural Creek Simulation

Conclusions

Conclusions

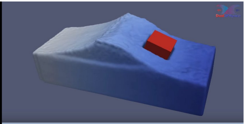
Paraview Visualization

Particle view



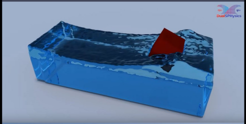
Paraview Visualization

Isosurface view



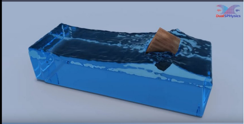
Blender Visualization

Standard cycles rendering



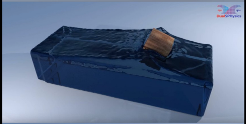
Blender Visualization

Texturing Support



Blender Visualization

Improved materials and lighting

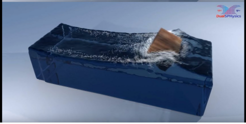


Blender Visualization

Foam simulation

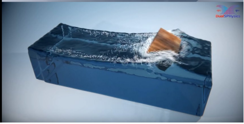
Blender Visualization

Motion Blur



Blender Visualization

Post-Processing & Compositing



1. A.C.J. Crespo, J.M. Dominguez, B.D. Rogers, M. Gomez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro y O. Garcia-Feal. **DualSPHysics: opensource parallel CFD solver on Smoothed Particle Hydrodynamics (SPH)**. In: Computer Physics Communications 187 (2014), pags. 204-216. DOI: 10.1016/j.cpc.2014.10.004 .
2. M. Ihmsen, N. Akinci, G. Akinci y M. Teschner. **Unified spray, foam and air bubbles for particle-based fluids**. In: The Visual Computer 28 (2012), pags. 669-677. DOI: http://dx.doi.org/10.1007/s00371-012-0697-9 .