HYDRAULIC EROSION

Generalized solution to modelling hydraulic erosion

- Moving water dislodges and transports material

Bedrich Benes, Vaclav Tesinsky, Jan Hornys and Sanjiv K. Bhatia

The erosion algorithm

- Navier-Stokes
  - Calculate the flow of water
- Erosion – deposition model
  - Changes happen in the layers between water and erosion material
- Material transportation model
  - Dissolved material is transported within the water

Requirements

- Model environment as a regular voxel grid
- Liquid, air and material voxels
- Inflow (source) and outflow (drain) voxels

Navier-Stokes

- Models movement of incompressible liquid
- Assumes approx. constant temperature and density

$$\nabla \cdot \mathbf{u} = 0$$

$$\partial \mathbf{u}/\partial t = -(\mathbf{u} \cdot \nabla)\mathbf{u} - (1/\rho)\nabla p + \upsilon \nabla^2 \mathbf{u} + \boldsymbol{f}$$

$\mathbf{u} = U\mathbf{i} + V\mathbf{j} + W\mathbf{k}$: velocity field

ρ: liquid density

p: pressure

υ: liquid visocity

$\boldsymbol{f}$: external force

$\nabla \cdot \mathbf{u} = \partial U/\partial x + \partial V/\partial y + \partial W/\partial z$

Navier-Stokes

- CFD too expensive
- Use approach introduced by Foster, Metaxas and Fedkiw
- Produces velocity field at each voxel face and pressure at each voxel centre
- From now on referenced as the «Navier-Stokes solver»

Erosion-deposition model

- Classify voxels:
  - **FULL**: voxel is full of water and some amount of dissolved material.
    - m = amount of dissolved material
    - $0 <= m < c_{max}$ (saturation point)
  - **EMPTY**: nothing but air
  - **MAT**: contains some material.
    - m = amount of material
    - Boundary voxels: $0 < m < 1$
    - Non-boundary voxels: $m = 1$

Erosion-deposition model

- Voxel state changes
  - **FULL -> MAT**: deposition of material
    - Occurs towards the bottom of the voxel
    - Occurs when $m > c_{max}$
    - $m – c_{max}$ excess material distributed to all **FULL** adjacent voxels
  - **MAT -> FULL**: erosion of material
    - Occurs when $m = 0$

Erosion – deposition: mathematics

- Model presented by Langendoen

$$\partial C/\partial t + (\mathbf{u} \cdot \nabla)C = E - D$$

C: Concentration of sediment mass

E: entrainment rate of material (erosion)

D: deposition rate

- E and D differ for cohesive and cohesionless materials

Cohesive materials

- Molecules stick together and allow for surface tension
- Ex: water

$$E = e(\tau/\tau_{ce} - 1)$$

$$D = \omega d(1 - \tau/\tau_{cd})$$

$e = 0.01$: erosion-rate constant

$d = 0.01$: deposition-rate constant

$\omega = 0.2$: fall velocity within the liquid

$1 - \tau/\tau_{cd}$: probability particle will stick to material

τ : bed shear stress

τ_{ce} : shear stress strength of material

τ_{cd} : denotes when particles start to deposit

Cohesive materials

- τ_{ce} and τ_{cd} can be given experimental values
  - The paper uses 10 for both
- τ is given by material friction f, viscosity u and density ρ

$$\tau = f \rho u^2 / 2$$

Cohesive materials

- The erosion – deposition model accounts for material transportation
  - But we already do this in the Navier-Stokes equations
  - Thus we only need to solve:

$$dC/dt = E - D$$

  - This is solvable by Euler's method

Cohesionless material

- Little tendency to stick together
- Ex: Sand
- Equilibrium sediment mass
  - Sedimentation = diffusion
  - Basically, the material is «at rest»

Cohesionless material

- The erosion deposition model becomes:

$$E - D = (1/T)\,(C\_hat - C)$$

C_hat : equilibrium sediment mass

C : actual concentration of material

T : time-scale representing the adjustment of sediment mass from C to C_hat

  - Paper defines T as a material constant

- $C\_hat = \min(k||\mathbf{u}||, c_{max})$
  - k is constant, paper uses 0.7
  - Deposition higher when fluid velocity low
  - Erosion is highet when fluid velocity high

Implementation

- 1. Simulation of erosion
  - Focus of the paper
  - OpenGL previewer – user can stop simulation when it does not proceed in the desired manner
- 2. Visualization

Implementation

- Read initial conditions
- For each timestep:
  - Solve fluid dynamics problem by using the Navier-Stokes solver.
    - Gives pressure field p and velocity field **u**, which is used in following steps
  - Calculate erosion and deposition
    - Expanded in next slide
  - Solve for material transportation within the fluid
    - No details given in paper

- Step 2: calculate erosion and deposition
  - Cohesive model
    - Find all voxels on boundary of water and material
    - Apply equations discussed above to compute erosion
    - Transfer corresponding amount of material from MAT to FULL voxels
    - Calculate deposition from MAT voxels
  - Cohesionless model
    - If C > C_hat, apply deposition
    - If C_hat > C, compute erosion by equations discussed above

Results

- Fully 3D hydraulic erosion model

- Ability to simulate vastly different scenes
  - River meanders
  - Low hill sediment wash
  - Natural water springs
  - Receding waterfalls

Results

- Performs well in modelling
- Not appropriate for real-time performance
  - 3 GHz PC, scene with 450 000 voxels
    - 17 seconds per frame
- Notable simplification: Navier-Stokes solver does not consider density change in FULL voxels