Physically-Based Interactive Sand Simulation
Marta Pla-Castells, Ignacio García-Fernández and Rafael J. Martínez-Durá
Instituto de Robótica, Universidad de Valencia, Spain

Introduction

- Interactive terrain simulation is necessary in some Virtual Reality applications. It is a complex modeling problem.
- In simulation for training realism is a must, because the trainee has to acquire skills based on the behavior of the environment.
- For this reason, when the simulation involves terrain manipulation the soil model must be physically-based.
- Terrain is usually represented as a grid or height field.
- In the Computer Graphics literature, models have been proposed mainly for terrain animation purposes [1, 2].

Main Goal

- Our main goal is to provide a model of sand:
  - that is physically-based;
  - that considers collisions and horizontal displacement of sand;
  - with reaction forces;
  - and that can be used in real-time, interactive applications.

Physically-Based Sand Model

- We adapt a classical model from the physics literature [3] to be used over the discrete representation of terrain used in Computer Graphics.
- This model was previously presented in a theoretical work [4], and here we propose its application to Computer Graphics.
- A local rule checks for steep regions, and causes avalanches that reduce the slope, like in cellular automata formulation of sandpiles.
- The model is interactive: contact forces can be computed and applied both to the sandpile and to the object in contact.

Horizontal Sand Displacement

Terramechanics

When a portion of soil is displaced, it is considered as a solid that slides along a pre-defined interface, causing what is called a fracture [5].

Discrete Model

- The variables of the terramechanics model are computed using the discrete model of sand.
- The fracture region and the horizontal force on the tool are computed.
- If the tool advances the material is pushed, sliding along the fracture. Every cell’s height above the fracture increases accordingly.

Numerical Experiments

- We have implemented two tests, T1 and T2:
  - T1 corresponds to a cube with unit edges falling on a flat ground.
  - T2 corresponds to the same cube pushing material horizontally.
- The tests have been repeated for different grid densities, recording the time necessary to run 10000 steps.
- To use an scenario easy to reproduce, the physics library Open Dynamics Engine (ODE) has been used.
- Collision detection has been done by means of ray casting.

Results

Results for Test T1 (Only Vertical Forces Appear)

- All times are below 0.2ms.
- Although the increase of the cost is not linear, results are very good even for dense grids.
- This results indicate that the sand model is suitable for real-time applications.

Results for Test T2 (Case with Lateral Displacements)

- All times are below 2ms.
- Again, the results are good even for dense grids.
- In this case, scalability of the model is not so good as in T1.
- The CPU cost of the collision detection and horizontal material displacement has been obtained (in %)
- The code corresponding to these two tasks is not optimized, and it takes 60%–80% of the CPU.
- Results are good. But even better results are possible, with optimization.

Conclusion

- The model is physically-based.
- It considers sand pile evolution, interaction and horizontal material displacement.
- The situations where horizontal displacement happens are more CPU demanding, but still the model can be run in real-time in complex applications.
- Optimization strategies have to be investigated:
  - Parallelism,
  - Multi-scale analysis of the grid,
  - Hardware accelerated collision detection,
  - Implementation of the model on GPU.

References