Real time Erosion simulation

River erosion computer simulation

OpenGL river

Shallow water computer simulation

Navier-Stokes Equations (to do with water simulating)

Lit review sections:

Real world example

Water simulation methods

Terrain generation

Erosion methods

Physics based erosion methods use different types of fluid simulations. Most of them are expensive to run and contain data dependencies that make it hard to execute on parallel hardware.

The first method was a partial simulation of valleys and rivers. Another one uses Navier stokes equation on a 3D regular grid that simulates the erosion and another used a simplified Newtonian physics model for velocity computation on a 2D grid.

Some methods were found to help make this process a lot less expensive and able to be ran in real time. these include, 2D Navier-Stokes equations, a shallow water simulation which allows for a simulation with real time erosion**[2]** and a method that uses virtual pipes which is key for executing in parallel.

Erosion model uses 2D uniform grid where each cell holds – terrain height, water height, suspended sediment amount, water outflow (being the pipes to adjacent cells), velocity vector, thermal erosion outflow flux.

They also use a scale that dampers the erosion equation the deeper the water gets because water moves slower the deep down you go which means the sediment is less disturbed.

**Notes from [1]**

Different types of erosion – weathering is a small scale type of erosion that usually affects stones and rocks. It creates rifts on their surface in a thermal or chemical way. Thermal weathering causes disintegration of material by thermal shocks in the presence of moisture that has bigger dilatations that the rock. Chemical weathering is caused by a chemical reaction between a liquid on the surface and the material itself.

Denudation is another type of erosion. The first type of denudation is gravity-conditioned mass movement where gravity pulls pieces of sand and gravel down the slope. The next type of denudation it splash erosion which is caused by raindrops falling onto a material causing an elliptical footprint. The last type of denudation is fluvial. This is the most important for this project as it is when water flows down a surface picking up different materials as it moves.

Full 3D hydraulic erosion was a paper that uses Navier-Stokes equations which are coupled with material transportation and solved on a 3D grid. This simulation is able to simulate receding waterfalls, river bed and river bank erosion, meander break, etc. this model provides high quality results but is also very expensive to run which makes it impossible to run in real time.

**Notes from [2]**

The contributions of this paper are: integration of three erosion algorithms, extension to a layered terrain representation, adaptation of the existing algorithm to use the pipe-model for water transportation, implementation of the algorithms on the GPU and selective erosion of sub-tiles on the GPU.

Terrains can be created by using regular height fields. These are useful as they are efficient for erosion simulating and rendering. But they do not support using multiple materials for different layers. A technic that would be able to support this is layered representation where each 2D grid location holds a 1D array of all the materials that are layered in that location with each one having their own height depending on how much of that material there is**[4]**.

The fluid simulation technic used in this paper is the hydrostatic pipe-model also called the column-based model which categorizes the water volume into columns with constant physical properties. This method gets the pipes are used to stabilize the water pressure cased by differing levels between pipes. This process is easily parallelizable which makes it well suited for usage on a GPU. Calculation description on page 4.

Three erosion algorithms are used in this paper. Forced-based, dissolution-based and direct material transportation through sediment slippage.

Forced based erosion uses the forces created by running water and the effect it has on the surrounding terrain description of calculations on page 5.

Dissolution based is based on an observation that a slowly moving water penetrates the underlying soil and creates a layer of slowly moving loose soil that accumulates on the bottom until it reaches an equilibrium.

Material slippage is where material exposed by erosion or being deposited gets affected by gravity. However there is an internal tension that prevents materials to continuously fall. This can be characterized by the talus angle. If the talus angle is exceeded a certain amount of material is removed from the location.

The order at which this paper integrates each algorithm is: first water in input into the system via rain, user interaction, or user defined water sources. The system then runs through the pipe model algorithm to calculate which direction the water in moving next. Then the force-based and dissolution-based erosion steps are evaluated. And then the last step is the material slippage. After this everything is rendered for the next frame.

**Notes from [3]**

I think this paper is the first to implement a fast real time hydraulic erosion model. Other than that not much else on this paper

**Notes from [5]**

This paper talks about 2 different methods for simulating water. Grid based and particle based. The best grid based method to use is the pipe model. This is where a cells in a 2D grid are connected via pipes. This can be 4 pipes or 8 pipes per cell. 4 pipes is best to use as its faster while still allowing for a realistic simulation. Although 8 pipes would be more realistic it is much slower to compute. The most commonly used particle based simulation is smooth particle hydrodynamics. This method tracks particles in varying locations instead of using a fixed grid. Each particle represents a quantity and mass. The quantity is calculated by a weighted sum of the neighbouring particles. An advantage of using particle based methods over a grid based method is that the only area that is simulated is where there are particles rather than the entire scene needing to be simulated like with a grid based method. This only becomes a problem when simulating larger bodies of water like rivers or lakes as a lot of the particles will be sitting at the bed of the river or lake taking up processing power whilst not doing anything. This problem has been worked on though with the shallow water equation which limits the amount of particles to a certain depth.

**Notes from [6]**

This paper improves on the shallow water equations and implements a way of using a 2D grid and particle method of simulating water. Not all of this paper will help as its mostly with large bodies of water like the ocean and simulating waves.

**Notes from [7]**

simplifies the 3D Navier Stokes equations to get simulation in real time.

**Notes from [8]**

Uses velocity field to simulate erosion. They use a 2d arrays that holds water quantity, velocity vector, collision energy and elevation. They obtain the velocity field by arranging water particles on each grid point of the terrain model and examine the flow of each individual water particle. They then use this information to obtain the collision energy which is then used for the erosion. This method is quite slow as the program ran a 128x128 array with 50 steps in 5697s.

**Notes from [9]**

Video on unreal engine 4 water physics. Could be used as a comparison for looks and performance.

**Notes from [10]**

Uses a fast method for animating rivers but has very limited interactivity.

**Notes from [11]**

Literature review on different terrain modelling methods.

**Notes from [12]**

[1] <https://diglib.eg.org/bitstream/handle/10.2312/EG2011.short.057-060/057-060.pdf?sequence=1>

[2] <https://www.cs.purdue.edu/cgvlab/www/resources/papers/Benes-2007-Real-Time_Erosion_Using_Shallow_Water_Simulation.pdf>

[3] <https://cgg.mff.cuni.cz/~jaroslav/papers/2008-sca-erosim/2008-sca-erosiom-fin.pdf>

[4]<https://data.exppad.com/public/papers/Layered_data_representation_for_Visual_Simulation_of_Terrain_Erosion.pdf>

[5] <https://xing-mei.github.io/files/erosion.pdf>

[6] <https://www.modeemi.fi/~daemou/mindtrek12.pdf>

[7] <https://matthias-research.github.io/pages/publications/hfFluid.pdf>

[8] <http://graphics.uni-konstanz.de/publikationen/Neidhold2005InteractivePhysicallyBased/Neidhold2005InteractivePhysicallyBased.pdf>

[9]<https://static.aminer.org/pdf/PDF/000/593/535/terrain_simulation_using_a_model_of_stream_erosion.pdf>

[10] <https://www.youtube.com/watch?app=desktop&v=EfzhMqZyilI&ab_channel=ImaginaryBlend>

[11] <https://artis.inrialpes.fr/Publications/2009/YNBH09/riversEG09.pdf>

[12] <https://onlinelibrary.wiley.com/doi/full/10.1111/cgf.13657>

<https://www.researchgate.net/publication/259398063_Level_of_Detail_for_Real-Time_Volumetric_Terrain_Rendering> - volumetric data

<https://perso.liris.cnrs.fr/eric.galin/Articles/2009-arches.pdf> - more volumetric stuff

<https://diglib.eg.org/bitstream/handle/10.2312/conf.EG2013.stars.155-173/155-173.pdf?sequence=1&isAllowed=y> – something on multiple methods

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=5ca07a56725f8ae6c74778a86a4736ebaab6add6> – Sparse Voxel Octree

<https://graphics.tudelft.nl/~rafa/myPapers/bidarra.3AMIGAS.RS.pdf> - talks about procedural generation

<https://onlinelibrary.wiley.com/doi/epdf/10.1111/cgf.12530?saml_referrer> – Terrain modelling from feature primitives

<https://epubs.siam.org/doi/abs/10.1137/1010093> - fracional brownian motions, fractional noises and applications

<https://dl.acm.org/doi/pdf/10.1145/2461912.2461996> - Terrain Generation Using Procedural Models Based on Hydrology

<https://users.math.yale.edu/~bbm3/web_pdfs/encyclopediaBritannica.pdf> - mandelbrot set paper on something

<https://link.springer.com/chapter/10.1007/978-3-642-84574-1_34> - explains some of fractal equations.

<https://onlinelibrary.wiley.com/doi/full/10.1111/j.1467-8659.2010.01827.x> - survey of noise functions

<https://core.ac.uk/download/pdf/34480918.pdf> - pdf version^

<https://dl.acm.org/doi/epdf/10.1145/566570.566636> - improved perlin noise

<https://dl.acm.org/doi/epdf/10.1145/325165.325247> - perlin noise

<https://reader.elsevier.com/reader/sd/pii/S1569190X07000111?token=84D28D08B34808D6170A48E5C23B134AC777FACD5FAF7CA919E51C533C7269862CFED2FC981AB99E7281699BDE3346A0&originRegion=eu-west-1&originCreation=20230301223630>

<https://www.sciencedirect.com/science/article/pii/S1569190X07000111>

<https://static.aminer.org/pdf/PDF/000/593/535/terrain_simulation_using_a_model_of_stream_erosion.pdf>

<https://core.ac.uk/download/pdf/38910806.pdf>

<https://scholarworks.calstate.edu/downloads/m900nt80d>