Progress Report

Bruce Alcock

A Procedural, Minimal Input, Natural Terrain Plug-in for Blender

Supervisors: Kevin Glass and Shaun Bangay

Date: 15 May 2007

1 Previous Objectives

- Fix the now vaguely functional terrain erosion.
- Try eroding a variety of surfaces and checking the results don't contain artifacts.
- Try eroding a flat surface to see what kind of results can be expected in terms of completely natural erosion.

2 Progress

2.1 Musgrave Hydraulic Erosion

The downward spikes were eliminated by adding an extra condition in the algorithm (1) for breaking down the terrain at the current location, insisting that the current location's height is greater than the neighbour it's being compared to.

After much sitting and thinking about the code, a phrase from [3] comes to mind: "In a full two-dimensional implementation, one must take care to distribute water and sediment to all neighbouring vertices in amounts proportional to their respective differences in overall elevation." This explains the wavelike effect seen in 1, the left picture being the terrain heightfield

Algorithm 1 Current vertex breakdown

```
else:
  su2 = su+sv+ks*(cs-sv)
  if (av > au):
    av2 = av-ks*(cs-sv)
  sv2 = 0.0
```

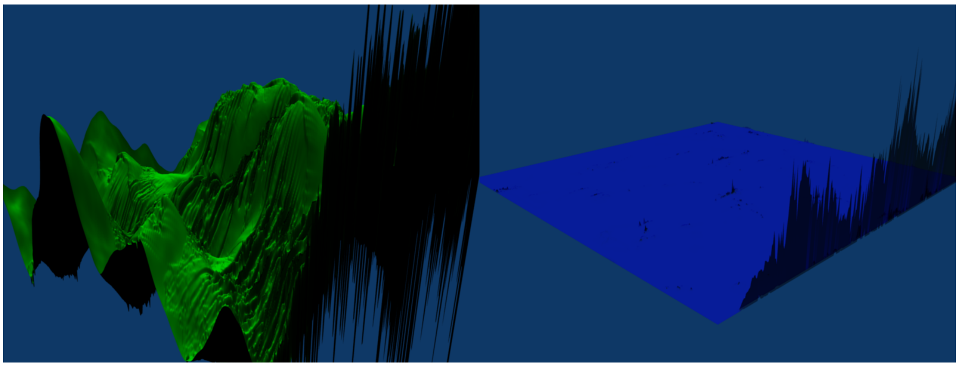

Figure 1: Wave-like erosion

and the right being the water levels. As the pictures show, all the water is being moved in one general direction, and this is because when calculating the erosion calculations one of the first neighbours (or a combination of them) is taking all the water and sediment and as such it is not being distributed evenly across the neighbours, which also explains the choppy appearance of the resulting terrain: the erosion is happening in a wave-like fashion and if viewed in this regard is actually quite realistic, except for the extreme right where huge anomalies occur because of the mass of excess water.

An attempt has been made to pre-calculate the values of the neighbours one by one but using the original center point each time and then proportionally affecting both the neighbour and center vertex proportionally, but this has resulted in many issues, including division by zero errors. Unfortunately [3] did not specify how they achieved this, leaving only the cryptic quote (see above).

2.2 Reading

In search of an implementation of [3], another paper [2] was discovered, which does detail how they calculated the neighbouring vertices' proportions. They reference [3] and use an adaptation of its hydraulic erosion algorithm, including an extension of evaporation. [2] claims very good performance of less than 2 minutes for a 400x400 grid through 500 time steps on a 500MHz PC, however this was implemented in C, which could be the source of speed.

Another paper by the same authors [1] suggests a method for using multiple values at each heightfield value, effectively using multiple heightfields to model differing materials in the ground as inspired by real ground, which means differing layers can have different rates for sediment conversion, etc, which translates to different softness levels. This allows things such as caves to occur, but the algorithm for thermal erosion (which is what the pictures show) for the multiple layers does not seem to be documented at all. This paper may be worth exploring at some stage, especially since they mention that it is not too difficult to extend the erosion algorithms onto the new representation of terrain.

3 Problems

Proportionally distributing water across neighbouring vertices.

4 Objectives For Next Week

- Fix the proportional distribution of water using the algorithm presented in [2].
- Implement evaporation of water over time [2].

References

[1] Bedrich Benes and Rafael Forsbach. Layered data representation for visual simulation of terrain erosion. In *SCCG '01: Proceedings of the 17th Spring conference on Computer graphics*, page 80, Washington, DC, USA, 2001. IEEE Computer Society.

[2] Bedrich Benes and Rafael Forsbach. Visual simulation of hydraulic erosion. In *WSCG 2002 Conference*, 2002.

[3] F. K. Musgrave, C. E. Kolb, and R. S. Mace. The synthesis and rendering of eroded fractal terrains. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 41–50. ACM Press, 1989.