RIVERS AND LAKES IN ARTIFICIAL LANDSCAPE

Jan Zelený
Master Degree Programme (2), FIT BUT
E-mail: xzelen08@stud.fit.vutbr.cz

Supervised by: Michal Hradiš
E-mail: ihradis@fit.vutbr.cz

ABSTRACT

This paper presents algorithms for generating height map of random terrains with influence of various erosion steps. I will present my novel algorithms for adding realistic rivers and lakes in these artificial landscapes.

The methods are designed also for modularity and any of the individual generation steps could be replaced by real-world data or previously generated height maps.

1. INTRODUCTION

Artificial landscapes are important in many fields of computer graphics including flight simulation, movie animation and especially computer gaming.

The techniques described in this paper focus on modeling the most important macroscopic features of landscapes, such as mountains, valleys, rivers and lakes. In this paper I will refer and shortly describe method for generating height map, and for adding thermal and hydraulic erosions for more realistic output. I will present and describe novel method for compute river paths (from their origin, we will call them "primary rivers"), river junctions, flow rate and method for compute lakes with their discharge ditches (called "secondary rivers"). These methods are designed for good visual appearance with low computation cost, not for geological accuracy.

2. GENERATING OF HEIGHTMAP

Many methods aimed to generating simple height map exist. Many of them such as mid-point displacement or Perlin noise [5] can (with bad chosen constants) produce height maps with axis-aligned local extremes. I prefer Fault formation [3] method, which after many steps produces rough, but enough randomized height map. In every step, it divides area onto two parts and adds some low height to one of them.

3. EROSIONS

The output from Fault formation looks like geological faults of tectonics plates. There are many influences in nature that smooth these faults over a long time. I will describe two of them: Thermal and Hydraulic erosion.

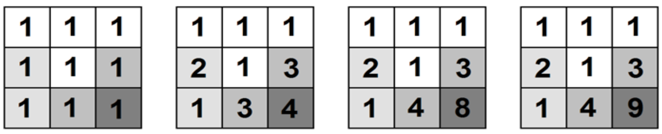
Thermal erosion comes when slope is too declivous. Part of material moves down the hill and fills gaps or valleys. Algorithm for simulation of this nature phenomena is following [7]: Let h, h_i, d_i be the actual height, neighbors i height and their difference: $d_i=h-h_i$. If d_i exceeds some threshold T, part of material (affected by constant c) is moved down the hill: $d_i > T: h_i = h_i + c(d_i - T)$.

Water erosion comes if landscape is not a dry desert. In that cases, there are at least seasonal rivers that modify terrain in slightly different way than thermal erosion does [1, 4, 6]. This erosion consists of 4 steps: Water appears ($w_{t+1} = w_t + K_r$), takes some material ($s_{t+1} = s_t + w_t * K_e$ and $h_{t+1} = h_t + w_t * K_e$), move down the hill and sediments dispersed soil $h_{t+1} = h_t + s_t$), let w_t, s_t, h_t be the water, dispersed soil and map height, K_r, K_e be the constants for rain and erosion rates. Ratio of soil erosion is moreover based on water velocity, slope and mass of soil already dispersed in water. Rate of sedimentation is influenced by similar attributes but also on vaporization or land soaking. Transportation of surface water with dispersed material is based most on heights of neighbors.

4. COMPUTING PRIMARY RIVERS

Landscape is not complete without realistic-looking drainage system with rivers and lakes. This is usually made by hand or is missing in common generated artificial landscapes.

I introduce new stable algorithm for generation of these two features of landscape. It computes "primary rivers", then lakes and "secondary rivers". First step is calculating drainage areas for every point of height map. First the method sets every point of map to constant small area c. Then it sums (for every point a) constant c and areas a_i of all its neighbors, for which a (current point) is the lowest neighbor: $a = c + \sum a_i$. After many iterations we get sum of all incoming drainage areas into this point. We can see 4 iterations in pic. 1.



Picture 1: 4 iterations in "drainage area size map" computation (c = 1).

This produces a map of drainage area sizes. Primary rivers paths can be extracted from the drainage area size map by Canny edge detector [2]. Finally rivers must be reconnected, because Canny edge detector can't make edge junctions.

If we want to know river volume, we simply divide flow rate (derived from drainage area map) and river velocity (in most simple case based on slope).

5. ADDING LAKES AND SECONDARY RIVERS

After previous step, rivers have origins somewhere in mountains and flow down the hill to the basins of valleys. In that places the second method adds lakes. Their volume is sum of all incoming rivers drainage area multiplied by rain rate constant: $V = \sum a_i * K_r$.

Picture 2: a) Valley basin, b) Lake filling, c) Overflowing and d) Neighbor connecting.

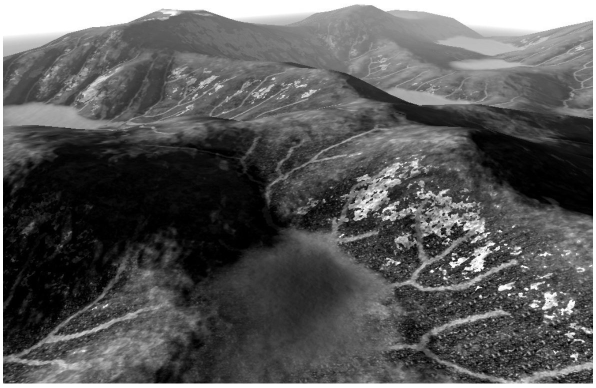
We start with valley basin and iteratively increase size of lake until we get enough volume for our lake (illustrated on picture 2). If lake overflows, secondary river appears and fills another lake. If this overflows backwards, we connect both lakes and fill them together.

6. SUMMARY AND FUTURE WORK

These contribute describes methods for generating height map, applying thermal and hydraulic erosions and introducing two novel stable algorithms for computation rivers, their junctions, drainage areas and lakes. These new methods can be used also on obtained real-word data, or previously computed height maps.

This methods are part of my master degree thesis which goes deeper in artificial landscape generation and presents methods for fractal modeling vegetation (based on L-systems) and simulating this ecosystem (with self-thinning model). Moreover it contains novel methods for real-time visualization of landscape and vegetation with programmable GPUs.

All methods that were described in this paper are already implemented as well as fractal vegetation and ecosystem simulation. Further I will improve ecosystem simulation, GPU based rendering and optimizing (see picture 3 with results).



Picture 3: Results. Visualizations of Artificial landscape.

REFERENCES

[1] Shearer, P.: Fun with erosion, Introduction to SIO Computing, 2005.

[2] Canny, J.F.: A computational approach to edge detection. IEEE: 679-698, 1986.

[3] River., Ch.: Game programming gems, volume 1-4, Charler River Media, 2002- 2007.

[4] Tucker., E. G., Bras., R. L.: Hillslope processes, drainage density, and landscape morphology, Water Resources Research, 1998.

[5] Perlin, K.: Perlin Noise, www: 2.1.2008: <http://en.wikipedia.org/wiki/Perlin_noise>.

[6] Beneš., B.: Real-Time Erosion Using Shallow Water Simulation, 4th Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS", 2007.

[7] Olsen., J.: Real-Time Procedural Terrain Generation, Department of Mathematics And Computer Science (IMADA), 2004.