A proposition of erosion algorithm for terrain models with hardness layer

Korneliusz K. Warszawski 1, Sławomir S. Nikiel 2

1 Faculty of Electrical Engineering, Computer Science and Telecommunications, University of Zielona Góra, Poland
2 Faculty of Economics and Management, University of Zielona Góra, Poland
k.warszawski@weit.uz.zgora.pl, s.nikiel@wez.uz.zgora.pl

Abstract: Processes of erosion occurring in natural environment depend on two major factors. The first is the strength of erosion force, e.g. wind, rainfall or water flow. The second is the terrain hardness or its tolerance to erosion forces. In this article we propose a method of modelling terrain erosion process where the force is uniformly distributed over the entire model with local distribution of varying terrain sensitivity. For the simulations we use two-layered terrain model. The first layer contains information about heights distribution (height-field) and simulate topography of the terrain. The second layer stores data defining its hardness (hardness-field) that represents different geological structures in the terrain.

Keywords: terrain modelling, erosion simulation, terrain hardness, hardness-field, virtual environment

1. Introduction

Modelling erosion processes is important for virtual environments, urban development, highways planning and military logistics. Data for modelling and visualization of digital terrain models can be imported from existed topographical and geological records. This approach is suitable for GIS-based (Geographic Information System) models. However, it is usually limited to the surface of selected land. Considering virtual environments used in military simulations, film industry and video games, a number of algorithms and techniques have been applied to the generation of synthetic land forms. They offer practically unlimited possibilities in terrain synthesis [1, 2, 3].

Usually, the landscape models are based on the fBs (fractional Brownian surface). Extending the standard one-dimensional fBm (fractional Brownian motion) is done with an additional parameter defining fractal dimension of the path structure [4, 5, 6]. Two-dimensional array of fBs elevations over a polygonal mesh can be a good approximation to a terrain [1, 7]. With a subdivided sphere as initial mesh, we can model the whole fractal planets [8, 9].

Visual realism of artificial landscapes can be improved by adding lakes and rivers. Effect of erosion caused by flow of spout can be simulated by methods based on CFD (Computational Fluid Dynamics) [10, 11], which are based on the physical model of the fluid flow. Visually, results of simulations are on acceptable level. However, the cost of CFD computations is prohibitive to real-time terrain synthesis. Alternative methods based on Fractal Theory can be used. Rainfall and river-based erosion model were presented by Nagashima [12] who used the Midpoint Displacement algorithm [7, 13] as indicator of erosion edge, then making it deeper.

An alternative approach to fractal-based terrain modelling is to use particle systems. The technique was originally presented by Reeves [15] as a fast method for real-time modelling of objects with irregular and dynamic shapes with undefined surface like clouds, fire or image of explosion. It can be also used for modelling of vegetation. A parallelized implementation of particle systems was done by Sims [16]. Benes and Roa [17] used the technique to simulate sand (or snow) shift and its interaction with other objects in a virtual environment. As shown by research of Benes [18] and Kristof et.al. [19], particle systems can be also used to simulation of erosion processes. Due to its specific characteristics, the method can be used to enrich terrains with rivers and canyon structures.

2. Terrain model

Similar to work of Benes and Forsbach [20, 21] we constructed our terrain model as a layered representation. However, in our case we used two layers with independent records. Both layers are discrete $m$-by-$n$ matrices, where $m, n \in \mathbb{N}$. The first layer stores information about heights (height-field), while the other contains hardness data (hardness-field).

All of our simulations are based on synthetic data. The height-field layers are generated with particle downfall algorithm [22]. For hardness-field synthesis, we used the modified Poisson Faulting method [23]. A sample of generated model layers is shown in Figure 1.

![Sample of model layers](image)

(a) height-field layer (rendered in Terragen)  
(b) hardness-field layer

Figure 1. Sample of model layers.

3. Erosion process

Long term erosion processes are usually based on land flattening in downstream transportation of geological material through terrain areas. The classic modelling approach does not require the information about the soil susceptibility for erosion forces, like its internal chemical structure or density, and usually smoothens the terrains. The hardness layer adds valuable information to terrain model.

In our simulations, we select two major phases in the erosion process. The first phase is a landslide where reduction of height-field is applied. The second phase is a land levelling
where we simulate downstream material transportation. The structure of erosion simulation algorithm is presented in Figure 2.

### 3.1. Landslide procedure

In the landslide phase, we make height modifications of the terrain model with regard to the erosion force and underlying geological structure. The latter is generated in related hardness-field model. A sample of solely landslide procedure results is shown in Figure 3 and the initial model for that sample is presented in Figure 1.

Let \((D)\) be a hardness-field layer of terrain model. Let \((F)\) be an erosion force which affects the whole surface of the model, and where \(\{F \in \mathbb{R} \mid 0 \leq F \leq 1\}\). Let \((i,j)\) be a pair of indexes, where \(i, j \in \mathbb{N}\). Let \((d_{i,j})\) be a hardness data stored in a cell from the hardness-field layer \((D)\), where \(\{d_{i,j} \in \mathbb{R} \mid 0 \leq d_{i,j} \leq 1\}\) and let it be inversely proportional to the erosion force. Then the loss of height \((\Delta h)\) can be calculated from the Equation 1.

\[
\Delta h = \begin{cases} 
\frac{F}{d_{i,j}}, & \text{if } d_{i,j} > 0 \\
0, & \text{otherwise}
\end{cases}
\]

Let \((H)\) be a height-field layer and let \((h_{i,j})\) be a record stored in the layer, where \(\{h_{i,j} \in \mathbb{R} \mid \min \leq h_{i,j} \leq \max \mid \min \neq \max\}\). Then the new value of the terrain structure cell \((h'_{i,j})\) will be modified by the loss of height \((\Delta h)\) factor as is presented in Equation 2. In exception for cells which has no hardness defined in which case, the terrain surface is levelled to the ground level.
A proposition of erosion algorithm for terrain models with hardness layer

3.2. Land levelling procedure

In the second phase we simulate the downstream material transportation. Each record of the height-field layer after landslide phase is recalculated as an average value stored in that record and its neighbourhood. We assumed that the material can be transferred only in range of one cell. According to cell position, it can have three neighbours when cell is located in corner of height-field layer, or it can have five neighbours if it is located on layer border, or it can have eight neighbours in all other positions. A sample of solely land levelling procedure results is presented in Figure 4 and the initial model for that sample is shown in Figure 1.

Let \((h'_{i,j})\) be the landslide height record stored in the cell of layer \((H)\). Let \((i, j)\) define indexes of that cell. Let \((k)\) be the number of the cell neighbours, where \(\{k \in \mathbb{N} \mid k = 3, 5, 8\}\). Then new cell height record \((h''_{i,j})\) calculation can be constructed as in Equation 3.

\[
h'_{i,j} = \begin{cases} 
  h_{i,j} - \Delta h, & \text{if } d_{i,j} > 0 \\
  0, & \text{otherwise}
\end{cases}
\]

(2)

Figure 3. Sample results of landslide procedure (rendered in Terragen).

(a) after 50 steps  (b) after 100 steps

Figure 4. Sample results of land levelling procedures (rendered in Terragen).

(a) after 50 steps  (b) after 100 steps
\[
    h''_{i,j} = \frac{\sum_{p=i-1}^{i+1} \sum_{q=j-1}^{j+1} h'_{i,j}}{k + 1}
\]

4. Results

Above discussed partial erosion procedures combining height-field and hardness models gives results useful in virtual environments. The combination offer better visual simulation tools for erosion processes.

Our experiments revealed that the impact of land levelling phase on the final model is definitely larger than of the solely landslide phase. To avoid rapid flattening of the landscape we choose to limit the number of levelling procedures in the simulation of the erosion process. We called this factor the levelling ratio and it has form of \((1/x)\) which means that one land levelling procedure is executed after every \(x\) landslide phases. Sample results of levelling ratio influence for final model is shown in Figure 5 and the initial model for that sample is presented in Figure 1.

The sample landscape presented in Figures 1, 3–5 was 300 × 300 resolution model with 4\(^{th}\) class (\(\lambda\)) hardness layer [23] and erosion force value set to \(1 \times 10^{-3}\).

Figure 5. Sample results of full erosion process after 100 steps (rendered in Terragen).
The average performance of the implemented prototype model was near real-time. The test experiment was executed on standard Intel Core i7-4771 CPU and was repeated 1000 times as a sequential implementation. Terrain layers were generated at each step. The erosion force was set to $1 \times 10^{-4}$ and levelling ratio was set to 1 : 1. The hardness layer was the 5th class. For test groups we chose models resolution between $500 \times 500$ and $900 \times 900$ nodes. The performance diagrams for landslide and land levelling procedures is shown in Figure 6.

![Figure 6. Performance graphs.](image)

5. Conclusions

Application of terrain models with hardness layer in erosion introduces soil classes for landslide simulation process. As in natural terrains, there are areas which erode with different pace. The internal structure built from softer geological materials is washed out quicker than the sectors of terrain built from harder minerals. The landslide phase reflects the direct pressure of the erosion forces where land levelling phase reproduces movement of soil elements. The method performance shows almost linear nature of the algorithm. The medium resolution of model, can be successfully used in application running on standard workstation or home computers.

The further research will be concentrated on adopting the method for voxel-based models and parallel implementation of the simulation algorithm. This could be potentially helpful in the studies on effect of erosion in three-dimensional models, i.e. caves, rock shelves or cliffs. Figures 7–10 presents additional examples of algorithm results.

Acknowledgements

The author is a scholar within Sub-measure 8.2.2 Regional Innovation Strategies, Measure 8.2 Transfer of knowledge, Priority VIII Regional human resources for the economy Human Capital Operational Programme co-financed by European Social Fund and state budget.
Figure 7. Model size $120 \times 80$ nodes, $\lambda = 3$, $F = 1 \times 10^{-3}$, levelling ratio $1/7$.

Figure 8. Model size $120 \times 80$ nodes, $\lambda = 5$, $F = 1 \times 10^{-2}$, levelling ratio $1/5$.

Figure 9. Model size $800 \times 800$ nodes, $\lambda = 5$, $F = 1.5 \times 10^{-3}$, levelling ratio $1/5$.

Figure 10. Model size $320 \times 240$ nodes, $\lambda = 7$, $F = 2.5 \times 10^{-3}$, levelling ratio $1/10$. 
References

Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG ’02), pp. 79–86. Plzen–Bory, Czech Republic, 2002. ISSN 1213-6972.
