Hierarchical Erosion of Synthetical Terrains

Bedřich Beneš, Ivo Marák, Pavel Šimek, Pavel Slavík*
DCSE, Czech Technical University,
Karlovo nám. 13, Prague, Czech Rep.
e–mail: {benes|marak|simekp|slavik}@sgi.felk.cvut.cz

Abstract
This paper deals with the visual simulation of synthetical terrain erosion. The previously published techniques were precise but very time consuming. We present here an efficient approximation of terrain erosion. The erosion algorithm uses regular height field as input data and works on different levels of resolution. The error caused with the multiple resolution is then partially reduced using visually plausible local approximation.

Keywords
realistic image synthesis, modeling, simulation, synthetic terrain erosion.

1 Introduction

There is one problem with fractal terrains [6]. If we measure statistical parameters of the average fractal mountain, then turn it upside-down and measure the statistical parameters again, the data obtained will be equal. Compared visually, the fractal mountain looks the same even turned upside-down. This is almost never the case in nature, except for some geologically new landscapes [4]. Moreover, the fractal techniques [2, 4, 7], provide only visual approximation of real terrains.

Real landscapes are *eroded*. Their shape has been changed with long term influences of temperature changes, rain, running water, wind, *etc.* These

*This work was supported by the *Grantová agentura České republiky* and the *Interni grant CVUT No. 3097475-336.*

processes cause movement of the material and, as a consequence, landscapes change their appearance and statistical parameters.

One of the common problems of previously published methods is their high time complexity. This paper deals with the erosion caused by temperature changes known as *thermal weathering* [6] and introduces new efficient method approximating these processes. This approximation is derived from multiresolution and works with different levels of resolution of the regular height field. The approximation error is then partially reduced by a visually plausible approximation.

This paper is organized as follows. Section 2 presents a brief overview of the previous work. Thermal weathering is described in section 3. Section 4 presents the algorithm principle; the results and the implementation remarks can be found in section 5.

2 Previous work

Considering the criteria of terrain erosion, fractal terrain generation techniques can be divided into two classes. The first class of algorithms provide (generate) already eroded terrains [2, 3, 7, 8, 9]. The second group describes erosion of *any* terrain [5, 6], regardless whether the data representing the terrain was obtained with some artificial technique or measured data was used.

The methods of both classes have their place in computer graphics and the applications range from virtual reality to the film industry. However, we must note one important difference. The erosion algorithm run on real data (*e.g.,* obtained from satellite photos) can be considered as a simulation and therefore provides practical results useful in ecology, GIS, *etc.*

2.1 Eroded terrain generating techniques

Prusinkiewicz and Hammel [9] use context sensitive rewriting processes based on a random midpoint displacement method [7] on a triangular grid. Their algorithm generates landscape and river simultaneously. This method creates one non-branching river as result of the context sensitive L-system [10] operating on triangles.

The spectral synthesis method [7] (also called Fourier transform filtering) is another representative of this kind of fractal modeling method. The principle of the method involves generation of Fourier coefficients in such a way, that inverse transformations provides a fractional Brownian surface (*fBm*) of a defined fractal dimension.

Detailed description of the method can be found in [7].

Pickover [8] uses a logistic function for generation of "extraterrestrial terrains". This method has the ability to create ridges, craters and streaks. Its advantage is that it can be used for tuning of already created terrains.

Kelley *et. al* [3] published an algorithm consisting of two distinct steps. The first step represents the creation of a drainage network of rivers according to geologically known data and behavior. In the second step, the terrain is modeled as a set of surfaces under tension fitting previously generated data.

2.2 Terrain erosion models

Musgrave *et al.* [6] introduced techniques which are independent of the terrain creation algorithm and can be applied to already generated data represented as regular height fields. They introduce two methods: *hydraulic erosion* and *thermal weathering*.

Hydraulic erosion is caused by the presence of water in the form of rain. The basic method of transport is as follows: part of the material is dissolved in the water, the water runs down hill and after the water evaporates the material stands in a new place. This approach can be easily extended height from the field representation to the set of triangles used in [1].

Thermal weathering is caused by temperature changes causing small portions of the material to crumble and pile up on the bottom of an incline. Both methods are controlled by several coefficients.

The following sections are further explanations of the thermal weathering method. The algorithm will be explained there.

Marák *et. al* [5] simulate several types of erosion using context sensitive rewriting of matrices representing terrain patches. The patches are rewritten according to a certain user-defined set of rules representing an erosion process. The advantage of this method is that the algorithm can be controlled using some external rules.

3 Thermal weathering

Thermal weathering [6] uses a two dimensional matrix of real numbers (the regular height field) as input data. The regular height field represents a terrain patch. Let A denote the regular height field of resolution $n \times n$ consisting of elements a_{ij}, $i, j = 1, ..., n$. These elements represent the z-coordinate (or the amount of material) at a position $[i, j]$. The well known notion of matrices will be used in the following text.

The iterative algorithm visits every point a_{ij} of the A and redistributes some material to its neighbors having lower z-coordinate. Assume an element a_{ij} and its eight neighbors denoted by $b_0 = a_{i-1\ j-1}$, $b_1 = a_{i-1\ j}$,...,$b_7 = a_{i\ j-1}$ (see Figure 1).

b_2	b_3	b_4
b_1	a_{ij}	b_5
b_0	b_7	b_6

Figure 1: An element a_{ij} and its eight neighbors b_k.

The number of $b_k, k = 0, ..., 7$ having lower altitude than a_{ij} is denoted by d. If d is equal to zero, there is no neighbor having lower altitude than

a_ij, so there is no material to move and the algorithm continues with next the element $a_{ij} \in A$.

The distance to the neighbor having the next (highest) lower value than a_{ij} denoted by m is given by

$$m = \begin{cases} min\{a_{ij} - b_k;\ b_k < a_{ij},\ k = 0, ..., 7\} & \text{if } \ d > 0 \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

The value denoted by r_{ij}

$$r_{ij} = \frac{d}{d+1} m \tag{2}$$

is the amount of material redistributed from a_{ij} to b_k with $b_k < a_{ij}; k = 0, ..., 7$. So clearly the new value of a_{ij} is

$$a'_{ij} = a_{ij} - r_{ij}$$

and the new value of material added to the neighbors of a_{ij} denoted by b'_k is given by

$$b'_k = \begin{cases} b_k + \frac{a_{ij} - b_k}{\sum_{k=0}^{7}(a_{ij} - b_k)} r_{ij}, & \text{if } b_k < a_{ij} \\ b_k & \text{otherwise} \end{cases}$$

In other word, the amount of the material r_{ij} of a_{ij} is redistributed to the lower neighbors in portions according to the distance $a_{ij} - b_k$. This effect is demonstrated in Figure 2.

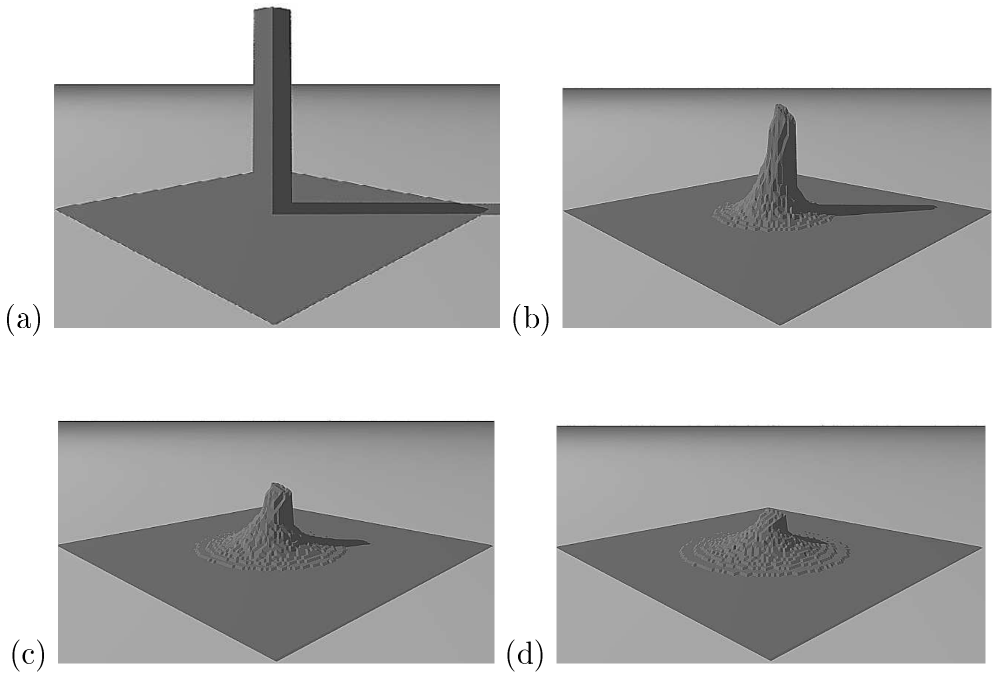

Figure 2: (a) Experimental mountain (b) after 100 iterations, (c) after 200 iterations, and (d) after 300 iterations of the thermal weathering method has been applied.

The algorithm visits every element a_{ij} of A and redistributes the total amount of material denoted by ε in every iteration.

$$\varepsilon = \sum_{i=1}^{n} \sum_{j=1}^{n} r_{ij}$$

The value of ε converges to zero with the number of iterations (attractor of the iterative process is a plane) as shown in Figure 3. We use two conditions

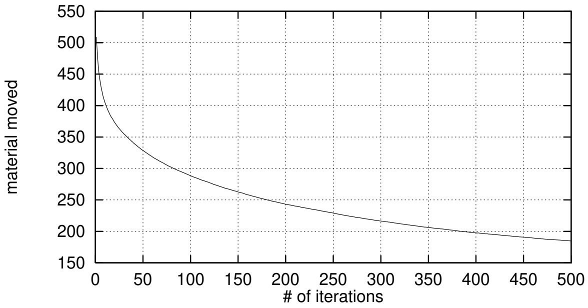

Figure 3: Amount of removed material as a function of number of iterations.

when the erosion process is stopped

1. after a certain number of iteration, or
2. when ε is below a certain value (*e.g.*, 90% of its maximal value).

4 The approximation

The basic principle of the algorithm is simple. In order to decrease the total time of simulation we decrease the resolution of the height filed A, then we run the erosion algorithm as described in section 3 and finally, the resolution is enlarged to the previous size.

Let A^0 denote the regular height field of the resolution $n \times n$ with its elements denoted by a^0_{ij}. We denote by A^s the regular height field of resolution $n/2^s \times n/2^s$ consisting of elements a^s_{ij}, where s is the level of subdivision. The elements of the subdivided height field are computed as the average of four elements from A^{s-1}

$$a^s_{ij} = \frac{1}{4}\left(a^{s-1}_{2i\ 2j} + a^{s-1}_{2i+1\ 2j} + a^{s-1}_{2i\ 2j+1} + a^{s-1}_{2i+1\ 2j+1}\right) \tag{3}$$

We should note here, that the computation of A^s involves s steps. Computation of A^2 from A^0, computation A^4 from A^2 *etc.* according to formula 3.

The thermal weathering algorithm (section 3) uses input data A^s. The next step of the simulation is the reconstruction of the matrix A^{s-1} from the eroded matrix A^s. The elements a^{s-1}_{ij} are evaluated from A^s as

$$a^{s-1}_{ij} = \frac{1}{4} a^s_{\lfloor i/2 \rfloor\ \lfloor j/2 \rfloor} \tag{4}$$

Note again, that the evaluation of A^0 from A^s involves s times the evaluation of formula 4. Both calculation (3 and 4) are less expensive than the erosion process from section 3.

The important factor is the error caused by the approximation. We have measured the *root mean square* (RMS) error between the regular height field eroded with the original method and with the subdivision technique applied.

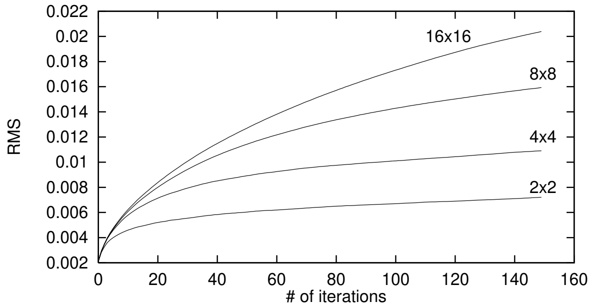

Figure 4: RMS as a function of subdivision and number of iteration.

Figure 4 shows the RMS according to the number of iterations and the subdivisions.

Because the elements of A^s were computed as averages we have lost details - averaged areas were not eroded. The erosion process in small areas can be approximated with a simple smoothing convolution filter involving blurring of the edges [11]. We use a convolution filter of size 3×3 for every element of the regular height field. Its new value is calculated as the average of the area of its eight neighbors and itself. The effect of such post processing of the eroded terrain is shown in Figure 5.

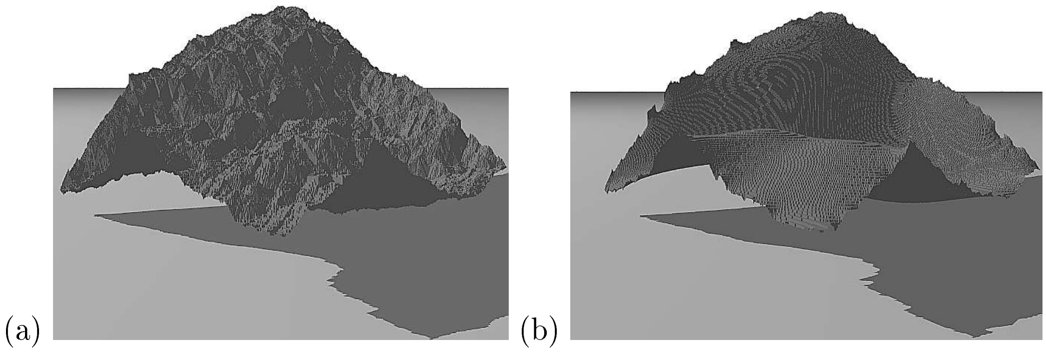

Figure 5: (a) fractal landscape after 500 iteration, in subdivision 16×16 without smoothing (b) the same terrain with smoothing.

The algorithm scratch has the following form:

for (j=0; j<NumberOfIteration; j++){
  for (i=1; i<s; i++) Calculate(A^i from A^{i-1});
  RunErosion(A^i);
  for (i=s;i>0;i++){
    Decompose(A^i from A^{i+1});
    Smooth(A^i);
  }
}

5 Implementation and results

Silicon Graphics *Octane* with *Solid Impact* graphics board and two *R 10000* processor on 195MHz with 256MB memory was used. The height filed in resolution 512×512 points was eroded 200 times in 12 minutes with no subdivision. Table 1 shows the speedup of the method as a function of the resolution for both cases – when smoothing was or was not used.

subdivision	**no smoothing**	**smoothing**
16×16	12.5	11.58
8×8	11.2	10.71
4×4	7.89	7.51
2×2	3.44	3.43
no subdivision	1	1

Table 1: Speedup as a function of subdivision to subdivision.

As we have mentioned in section 4, the smoothing decreases the RMS and the results are more visually plausible. Figure 6 shows the RMS as a function of the subdivision of the smoothed height field and of the non-smoothed one. Figure 5 shows the difference between the fractal landscape eroded with smoothing used and not used.

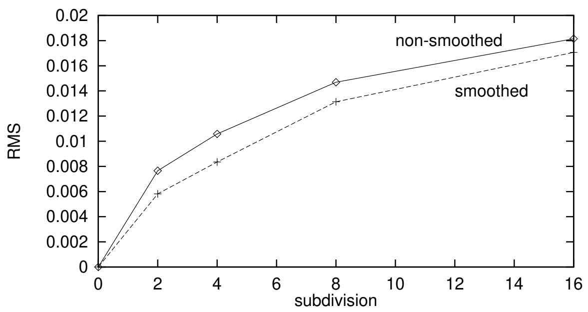

Figure 6: RMS is reduced with smoothing when subdivision is used.

The program is written in C. We have used *Open Inventor* for quick preview and ray tracer *Persistence of Vision* for high quality pictures.

6 Conclusions and future work

We have presented a new efficient approximation of terrain erosion algorithm called thermal weathering [6]. The algorithm uses different levels of resolution and the generated result can be used as a preliminary model (a preview) or the RMS error must be considered. The error caused by the approximation can be reduced with a convolving smoothing filter.

The problem with this technique is the need for recalculation of the averaged matrix after every iteration. Future work includes solving of this problem, as well as implementation of more methods than thermal weathering. Another improvement would be the use of some other data structure than height field. This representation is very space consuming especially for the cases as in Figure 2, where most of the altitudes are the same.

References

[1] A. Dixon and G. Kirby. A Data Structure for Artificial Terrain Generation. *Computer Graphics Forum*, 13:37–48, 1994.

[2] A. Fournier, D. Fussel, and L. Carpenter. Computer Rendering of Stochastic Models. *Communications of the ACM*, 25:371–384, 1982.

[3] A. Kelley, M. Malin, and G. Nielson. Terrain Simulation Using a Model of Stream Errosion. *Computer Graphics*, 22(4):263–268, 1988.

[4] B. Mandelbrot. *The Fractal Geometry of Nature.* W.H.Freeman, San Francisco, 1982.

[5] I. Marák, B. Beneš, and P. Slavík. Terrain Erosion Model Based on Rewriting of Matrices. In N. M. Thalmann and V. Skala, editors, *Proceedings of the WSCG'97*, volume II, pages 341–351. University of West Bohemia Press, Feb. 1997.

[6] F. Musgrave, C. Kolb, and R. Mace. The Synthesis and Rendering of Eroded Fractal Terrains. *Computer Graphics*, 23(3):11–1–11–9, 1989.

[7] H. Peintgen and D. Saupe. *The Science of Fractal Images.* Springer–Verlag, New York, 1988.

[8] C. Pickover. Generating Extraterrestial Terrain. *IEEE Computer Graphics and Applications*, 17:18–21, 1995.

[9] P. Prusinkiewicz and M. Hammel. A Fractal Model of Mountains with Rives. In *Proceedings of Graphics Interface'93*, pages 174–180, 1993.

[10] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants.* Springer-Verlag, New York, 1990.

[11] A. Watt and M. Watt. *Advanced Animation and Rendering Techniques: Theory and Practice.* Addison-Wesley, Reading, 1992.