Volume xx (200y), Number z, pp. 1–7

Table Mountains by Virtual Erosion

B. Beneš[1†] and X. Arriaga[2]

[1] Department of Computer Graphics Technology, Purdue University
[2] Campus Ciudad de México, ITESM

Abstract

We present an efficient algorithm for visual modeling table mountains (mesas) using erosion simulation. The method uses techniques from Computer Graphics to closely model the phenomenon, without relying on physically correct modeling. Our main goal is to devise an efficient algorithm that is geologically inspired and simulates visually plausible results in a reasonable time with certain user control over the process. The algorithm models a terrain as composed of two different materials, the hard one (rock) and the soft one (sand). The hard material is exposed to moisture and thermal changes that erode the side parts (the rimrock) of the table mountain. The eroded parts fall and change into the soft material. This material is subject of a different type of erosion. The soft material moves as sand or gravel with high inner friction. It moves slowly trying to reach an equilibrium and forms characteristic hillside of table mountains. We simulate this slow motion of soft material by a diffusion algorithm. The algorithm presented here achieves visually plausible results in reasonable time. The results are in tune with mountains observed in nature, and are comparable to the existing terrain modeling and displaying techniques.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computational Geometry and Object Modeling]: Physically Based Modeling;

1. Introduction

Procedural modeling, i.e., the process of a geometrical model generation by means of a procedure, has increasingly adopted more approaches from physically based and physically inspired modeling. This transition can be seen in many areas of Computer Graphics (CG): there is a boom of new techniques for fluid simulation, virtual plants are modeled by their growth, human body is modeled as a functional set of muscles and bones, and even the rendering algorithms are moving from the *ad-hoc* models to physics of light transportation.

We have focused on modeling terrains and landscapes in CG that comprise an important contribution to the overall realism of the resulting scene. Real terrains are heavily influenced by climate. They are modified over time by factors such as exposure to wind, temperature changes, running water, rain, and changes caused by humans and animals. Each of these factors create a different type of *erosion* to shape the terrain. These factors must be accounted for in terrain modeling and any techniques for landscape modeling should incorporate them.

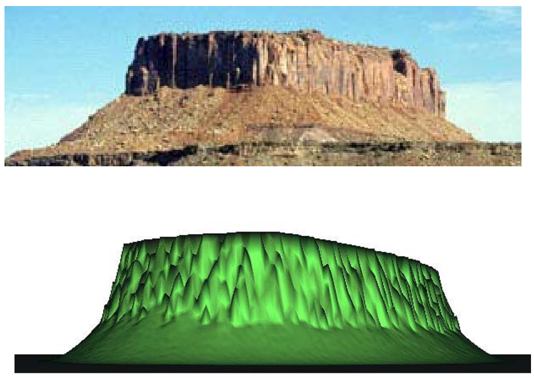

Figure 1: *Photograph of a table mountain (left) and the model obtained by erosion simulation*

A number of authors have presented novel techniques in this area, including ones based on fractals [Man83, FFC82]

† bbenes@purdue.edu

or multifractals [DFM*03], and on interactive tools for terrain modeling [Vte].

We present a new physically inspired algorithm for simulation of virtual mountains by erosion. Our algorithm generates a 3D model of table mountains or mesas (Figure 1). These mountains are typical for arid areas, deserts, and semideserts. The terrain model is generated by erosion process simulation. The landscape is represented as a regular height field that stores altitude of terrain in regularly distributed discrete points. The initial scene is composed from a rock. The rock is eroded and the material is changed into sand and mud. This softer material continues erosion by falling down and forming typical exponential hillside. We use a physically inspired model that simplifies the underlying differential equations to a set of rules, allowing for easy implementation and control of this method. The implementation runs fast and can be used at interactive framerates.

2. Previous Work

Most of the existing algorithms for terrain modeling are based on *ad hoc* techniques, with some emphasis on physical correctness.

[KMN88] presented a simple technique that provides landscape with rivers, ridges, and valleys. A user interactively sets up outlets from the landscape and the algorithm automatically generates the paths for fractal rivers. This information is sufficient to find the ridges. In the next step, valleys and ridges are interpolated by fractal patches. The result is a visually plausible model of eroded landscape with rivers, valleys, and mountains.

Probably the earliest work on terrain modeling using erosion is by Musgrave, *et al* [Mus89]. They modeled terrain as a fractal surface represented as a regular heights field that is then eroded, using thermal and hydraulic erosion phenomena. Thermal erosion is caused by thermal shocks. In this phenomenon, particles of terrain erode due to temperature differences, fall, and accumulate on the bottom of the hill. The hill has a slope whose angle is controlled by a talus angle. The second phenomenon, hydraulic erosion, is based on running water. Water dissolves soil that is then transported with the flow. After transportation, the water evaporates depositing soil.

Sumner, *et al*, extended the above thermal erosion model to simulate footprints in sand [SOH99]. Paths of bicycles as well as footprints and body prints are efficiently simulated due to the fact that the erosion algorithm is applied only to the affected areas.

Li, *et al*, presented a physical model for simulation of moving soils [LM93]. They show applications of their method to simulate mechanical activities such as caving, digging, and dumping.

Nagashima used semi-interactive approach to model eroded terrains [Nag97]. This approach interactively puts a fractal “river” on a regular height field. The river shape and the physically based model of erosion are used to erode the underlying terrain.

Chiba, *et al*, used velocity fields to create a physically based model of hydraulic erosion [CMF98]. In this technique, running water erodes the underlying terrain by the force resulting from water volume and velocity. The underlying height field is eroded according to a set of differential equations describing the erosion process.

Dorsey, *et al*, presented a method for stone erosion and rendering [DEJP99]. A surface model is covered by a volumetric layer that is then a subject of erosion. A set of differential equations describes the penetration of the material by the moisture that pushes out the soil from the volume to the surface. This gives the material a typical rusty color. These simulations result into sculptures eroded over time.

A similar approach was published in [GC99]. This approach simulated the surface erosion and corrosion by a set of particles that are put on the surface of the object and communicate.

Onoue and Nishita presented a simple but effective model of wind interaction with sand particles to simulate formation of wind ripples and dunes [KN00]. Wind captures particles of sand and carries them to different locations. The particles are then deposited at random positions, and fall to reach the gravity equilibrium.

Onoue and Nishita also presented the *virtual sandbox* [ON03]. In this model, regular height field describes the sand deposited on the ground and free particles model the sand in the air. They used a thermal erosion model, similar to [Mus89], to accommodate sand deposited on the ground to its location. Particles are moved in the air interactively. Due to the dual representation of sand, the model is able to represent the sand, the dust located on the objects, and the falling particles of sand.

Ito, *et al*, presented a volumetric approach that simulates vertical rocks typical of certain kinds of coasts [IFMC03]. The material is exposed to the influence of the environment that causes cracks in its structure and erodes.

Volumetric approaches usually present a huge demand on the storage space and the processing time of erosion is slow. The regular height fields cannot fully represent the 3D phenomena. This led us to a compromise layered data structure as an efficient tool for erosion simulation [BF01]. The RLE compression of volumetric layers of the same material provides good compression factor and can be used to simulate erosion algorithms without increasing the simulation time compared to the regular height fields.

Another hydraulic erosion technique was presented by the same authors in [BF02]. The running water dissolves soil transporting it elsewhere. The material also travels inside the water and tries to reach condensation equilibrium. As water evaporates over time, the saturation is exceeded and the soil is deposited on the bottom.

In our experience, none of the existing modeling algorithms can generate table mountains. The algorithm in reference [Mus89] applied to a hard material that is not eroded may provide visually similar results. However, the erosion simulation, as provided in this paper, provides visually more plausible results. Also our simulation inspired algorithm is closer to reality.

The paper continues with description of the formation and structure of table mountains. Next section describes the data structures and the simulation algorithm. The concluding section presents results and some open questions for future work.

3. Table Mountains Formation and Structure

Table mountains are formed over a long period but their age and origin can vary. The famous table mountain of Cape Town was formed approximately 500-600 million years ago; the mountains of Sierra Nevada were formed only 9 million years ago; and the Devil's Tower in Wyoming was formed some 40 million years ago [Rei03]. There are two basic ways table mountains are formed. The first involves a vertical lava flow that gets cooled down resulting in a hard and erosion resistant basalt. The other kind of formation results from a strong tectonic lift of very hard and compact material such as phonolite. Figure 2 shows photographs of real table mountains.

Over time, table mountains are exposed to a strong influence of erosion. Different types of erosion have different impact, and result in different table mountain shapes. For example, running water is responsible for forming one of the most amazing places in the world, the Grand Canyon (USA). Another strong influence is glacial erosion, typical in Canada. Wind erosion is usually important in deserts, and hydraulic erosion is caused by rainfalls in tropical areas. Even though the table mountains can be formed by different processes over time, the most commonly found table mountains are formed purely by *thermal weathering*.

Although the name thermal weathering suggests the influence of temperature, the principal geomorphic feature is water. It has been proven experimentally that the temperature change itself has a low impact on the surface of the material. The most important factor is the presence of atmospheric moisture that is always present to some degree even in dry arid areas. The moisture enters exposed surface of a mountain but expands at a different rate compared to the rock itself with a change in temperature. The difference in expansion may cause a strong inner tension that crushes the surface and erodes out parts of the rock that fall down. New mountain surface appears, then it is eroded again, and this process repeats. In this way, the rimrock (core) of a table mountain is

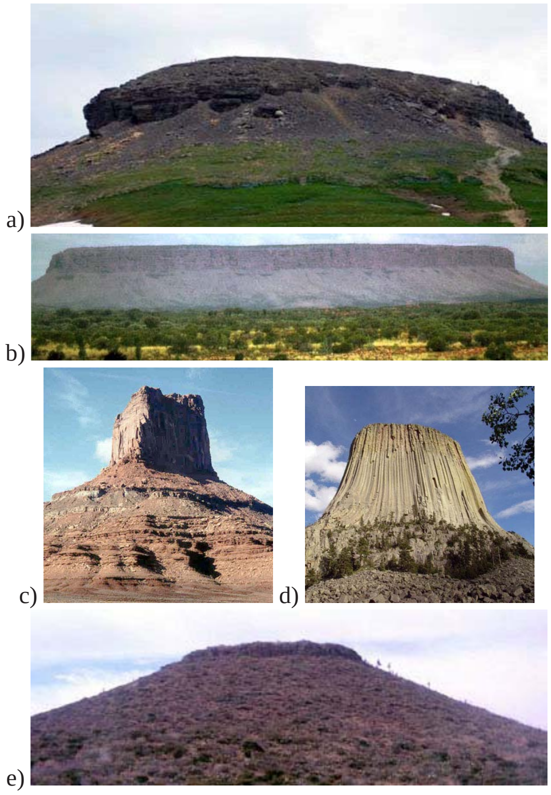

Figure 2: *Table mountains. a) Small hill b) Table Mountain (Cape Town) c) White Rim (SF), d) Devils Tower (WYO) e) Zacatecas desert table mountain (Mexico)*

thinned out but remains on the hillsides of the mountain that are covered with landslide deposits as schematically shown in Figure 3.

Over time, small quantities of material continue to fall down and greater parts – stones and rocks – may eventually fall as well. Depending on the height, they are more or less destroyed by the impact caused by the fall. In this way, material gets reallocated and the weathering process continues at a faster pace. The fallen eroded material forms the characteristic hillside of the foothills as shown in Figure 3. During the same time, the upper part of the mountain may get completely covered with the eroded material as observed in Figures 2 e) and 2 f).

There are several factors that influence the pace of the erosion process. The most important factors for the geomorphic change in the rimrock are its material properties, the amount of atmospheric moisture, and the depth to which the moisture can enter into the rock's surface. The speed of the secondary erosion, the landslide that forms the hillside of the table mountain, is influenced by many factors as well. The inner tension of the material or a well-developed cover of vegetation at the surface can slow down the erosion.

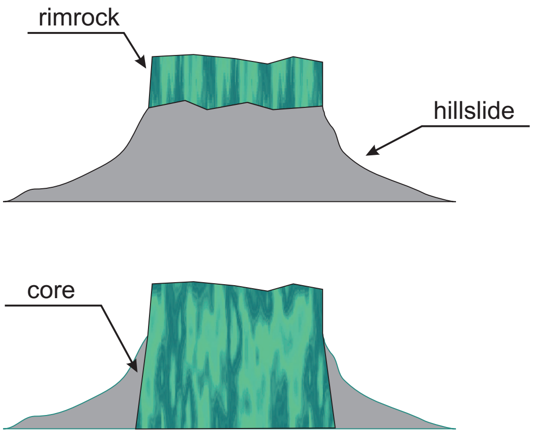

Figure 3: *Table mountain shape (top) and its inner structure (bottom)*

4. The Simulation Algorithm

4.1. Data Structures

The most commonly used data structure for terrain representation is a regular height field where each value corresponds to the altitude of a vertex. This data can be easily converted to a set of triangles and can be transformed into a data structure that is suitable for LOD and fast displaying techniques for terrains such as ROAM (Realtime Optimally-Adapting Meshes) [DWS*97] or BDAM (Batched Dynamic Adaptive Meshes) [CGG*03].

Regular height fields cannot fully represent 3D phenomena, such as caves, ledges, non-vertical holes, or pits. These phenomena can be stored by volumetric representations and operated on by fully 3D erosion algorithms [IFMC03]. A compromise data structure, allowing fully 3D representation of layered data, has also been introduced [BF01].

We use two regular height fields in our erosion simulation algorithm. We deal with two different materials, the rock (*hard material*) and the eroded material forming the hillside (*soft material*). Both materials are represented as two different layers. The hard material cannot lay over the soft one whereas the soft one must be always located on the hard one. When displaying, only the visible layer is showed. The eroded material is displayed with slightly different color than the hard one.

4.2. Erosion Simulation

The process of erosion simulation starts with the initialization of an entire scene by the definition of a hard layer. The hard layer can be defined in many different ways; we use procedural modeling forming regular or fractal shapes. The soft material results from the erosion of hard material.

The erosion simulation is run for each vertex of the regular height field and acts as a procedural gradient-based filter. Different procedures are executed at each vertex depending on the amount and the type of material in the tested location. If there is a layer of soft material the soft→soft erosion is run. If there is a layer of hard material the hard→soft algorithm is executed.

The erosion process is randomized. That is, at each step we use Gaussian random numbers to get a better visual rendering of the resulting objects. The Gaussian random numbers represent the fact that the size of the stones, eroded volumes, etc. usually follow the Gaussian distribution.

4.2.1. The hard→soft erosion

The main idea of the state change of hard material into soft material is that the surface of a rock is eroded if it is exposed to the atmospheric moisture and temperature changes. Two parts of the mountain can be exposed, the free top of the mountain and its sides. The top of the mountain is usually covered by vegetation, or a layer of already eroded sand, or soil. This isolates the top so that the top does not have a significant contribution to the erosion process. The most important surfaces for the erosion process are the free sides of mountain, or the rimrock. These sides are exposed to wind, moisture, and temperature changes and are eroded in a significant manner. The side erosion crushes the surface and parts of the rock fall down, as schematically depicted in Figure 4, and are changed into the soft material.

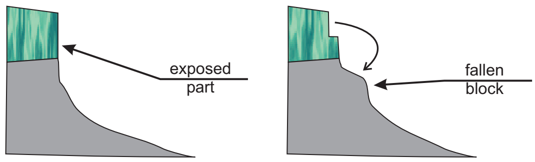

Figure 4: *The hard→soft erosion process*

The pace of erosion depends on many factors, such as the amount of atmospheric moisture, the amplitude of temperature changes, and the properties of the material. We aim to provide a technique for terrain modeling that should be easy to implement, intuitive, and fast. It can be assumed that the pace of erosion is slow (measured in centuries) compared to the changes of the outer conditions (happening on an intraday basis) and therefore, the pace can be considered as constant.

The algorithm first calculates the exposed area of the hard material (Figure 5) for each vertex of the height field. The distance between two vertices is constant, and hence, the exposed area can be expressed in terms of gradient changes. Each vertex can have at most four sides exposed to the atmospheric moisture. We calculate the sum of the exposed areas as the sum of the difference of height of the hard material of the vertex under consideration and the highest point

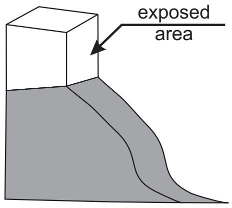

Figure 5: *The exposed area of a rimrock*

of the neighbors. The maximum result can be four times the height of the hard material and the minimum can be zero. The volume that is eroded from hard to soft is then given by an *ad hoc* formula

$$h = k_e h_{\max}^2,$$

where h is the height of the volume of the given vertex that is weathered, $h_{\max}$ is the total height of the hard layer, and $0 < k_e \leq 1$ is the pace of erosion. We have noticed, from our experiments, that a reasonable value is $k_e = 0.25$. This material is distributed to the lower lying neighbors and changed from the hard to the soft layer. The amount of distributed material is proportional to the altitude among the current and the neighboring vertices.

An example of the hard→soft transition is showed in Figure 6. The core of the mountain is eroded and is continuously thinned from all sides. The left column images show only the mountain core and the right column shows the soft layer. The last image shows the resulting mountain.

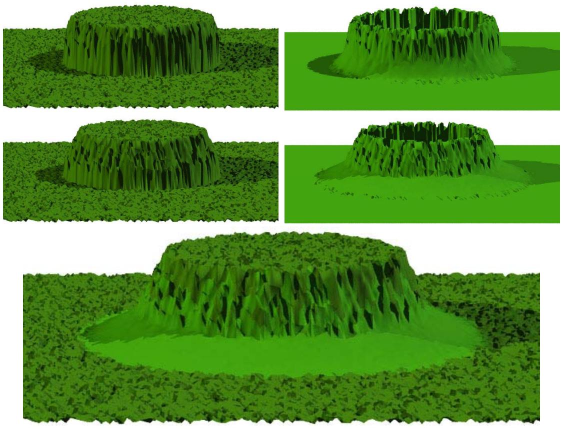

Figure 6: *Thinning of the core (left) of a mountain caused by erosion. The soft layer is displayed in the right column and both layers in the last image*

4.2.2. The soft→soft erosion

Once the soft material is deposited by the fall of the hard one it becomes a subject of another erosion process. We assume that the soft material has properties similar to sand, even though it can be composed of sand, mud, and stones. This kind of material behaves as sand or gravel with a high inner friction that tends to slow down the process of material deposition. An algorithm for this kind of erosion was originally presented in reference [Mus89] and later reused in different modifications by a number of authors [BF02, SOH99, ON03]. The material deposition is a gravity driven diffusion and its process is shown in Figure 8. Even this algorithm is not new, but we will discuss it in depth here.

For each location of the height field, the amount of soft material that exceeds the neighbors is detected and moved to the neighbors that are located below. Material is transported only to the neighboring elements so a certain time is necessary to reach equilibrium. We describe the possible cases that can occur. Suppose that a vertex has only two neighboring elements. The total height is denoted by m and is given as the sum of the hard h and the soft material s, i.e., $m = h + s$ and the height of the neighbor is $m' = h' + s'$.

If the difference $m - m'$ is less than or equal to zero, no material can be reallocated. If some material can be moved, we have to distinguish two cases that are shown in Figure 7. The top image shows the case when just a part of material can be moved, whereas the bottom row shows the case when all soft material can be reallocated.

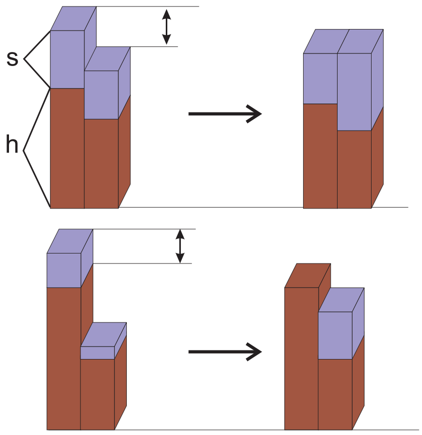

Figure 7: *Two possible cases of the soft material transportation*

The material must be redistributed to all its neighbors, but only some of them can be lower than the current element. Suppose the eight neighbors of the current element are denoted by the index $i; i = 0, 1, \ldots, 7$ and their corresponding quantities are denoted by s_i, h_i, and m_i. The element under consideration does not have an index. To get the correct distribution of the soft material, we have to first calculate the amount of material that can be reallocated. It is then redistributed according to the height to its neighbors, i.e., those that are lower will receive more material. The quantity of the soft material that is reallocated is denoted by Δs and the redistribution is described by

$$\Delta s_i = \Delta s \frac{s_i}{\text{sum}},$$

where Δs_i is the amount of material to be moved to the ith vertex, the sum is the sum of all positive altitude differences to the lower located vertices and the s_i is the actual difference to the ith neighboring vertex.

An example in Figure 8 shows the process of diffusion erosion of perfectly regular objects during the time. As can be seen, these objects tend to form similar shapes after some time. This is due to the fact that the above described diffusion material reallocation tends to soften differences among neighboring vertices.

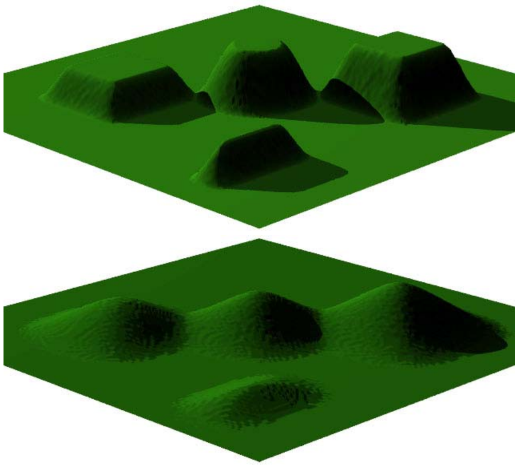

Figure 8: *Erosion of the soft material*

The complete algorithm works in the following way. Each vertex of the regular height field is checked whether it has some soft material or not. Each vertex is tested whether it has a free exposed side of hard material as well. Based on the result, either the soft→soft or the hard→soft erosion is performed. The soft→soft erosion can possibly cover the existing hard layers and the hard→soft erosion can contribute to existing layer of soft material. Special care must be taken not to exceed existing levels of material that would possibly cause oscillations in the material reallocation. The amount of material in the scene is always conserved.

5. Implementation and Results

The simulation algorithm is easy to implement. It works with regular height fields that can be easily exported from/to DEM data that can be used as an input to any GIS or for display by some LOD techniques.

The resolution of the height fields used in our simulations ranged from 300×300 to 2000×2000 elements and the simulation times of ten thousands steps ranged from about ten minutes to one hour on a 3GHz Intel CPU. We have used the Persistence of Vision ray-tracer to render most of the images in this paper. The memory requirements are related to the size of the height field and in our experiments were between 0.25MB to 15MB.

The implementation is easy and the computation time is relatively fast and depends on the number of hard elements that must be checked for possibility of being eroded and on the amount of soft vertices that must diffuse their material to neighbors.

The example in Figure 9 shows erosion of a perfectly regular shape. As can be seen the randomized mountain core thinning and the diffusion process lead to visually realistic results.

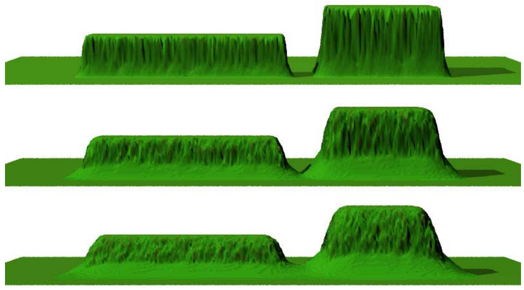

Figure 9: *Erosion of a regular shape leads to a visually realistic result*

Another example in Figure 10 shows a process of erosion on a large scale. Parts of a Perlin surface have been elevated simulating a tectonic lift. The erosion process influences mostly the elevated parts but also has some local effects. The elevated rocks cover the neighboring mountains with the eroded material.

6. Conclusion and Opened Issues

A simple but effective technique for modeling virtual table mountains is presented. The main contribution of the paper is an introduction of a simplified erosion process that leads to visually plausible models of virtual mountains. The algorithm is simple, easy to implement, and leads fast to the expected results, and is also easy to control. There are two different kinds of material, the hard material that is set when the scene is defined, and the soft material that is a result of the erosion simulation. Both materials are subjected to erosion, each in a different manner. The algorithm works with regular height fields and is compatible with existing modeling tools for creating terrains. The results can be easily used in many CG applications. Our experiments show that the algorithm provides visually plausible results in a reasonable time even for high resolution input data. The algorithm is predictive.

One of the problems of this method is that the rimrock could be better represented as a columnar joint as in the paper of [IFMC03]. We could represent fully 3D structures but we would loose the advantage of the unified representation that could complicate rendering. Another option is to implement the algorithm in a fully 3D volumetric manner. It will be interesting to see the way the voxels are eroded and the way the cracks in material are produced. Another issue, as mentioned by one of the referees, is the relation of the height field resolution to the represented details. Highly detailed height field does not represent the same details as very rough one. The algorithm should consider this fact and the eroded area should be calculated in the relation to the height field resolution. One hour calculation to simulate erosion of a very detailed table mountain is a very long time and users are not usually so patient. Apparently another opened issue is the speed of the algorithm. The aim of this paper was to show how a simple erosion algorithm can result in a realistic model. We did not focus on the rendering issues and certainly there are many things that should be improved. To give a special example, there are two different kinds of material that are textured differently and the zones where their meet should join the textures adequately.

One of the grand challenges is the search for a unified approach to erosion algorithms. There are many isolated techniques showing some kinds of erosion and providing some special cases of eroded objects. One of the most important goals is to find a generic algorithm that will be easy to control and will provide majority of the possible cases.

References

[BF01] BENEŠ B., FORSBACH R.: Layered Data Structure for Visual Simulation of Terrain Erosion. In *Proceedings of the 17th Spring conference on Computer graphics* (2001), vol. 25(4), IEEE Computer Society, pp. 80–86.

[BF02] BENEŠ B., FORSBACH R.: Visual Simulation of Hydraulic Erosion. *Journal of WSCG 10*, (2002), 79–86.

[CGG*03] CIGNONI P., GANOVELLI F., GOBBETTI E., MARTON F., PONCHIO F., SCOPIGNO R.: BDAM — Batched Dynamic Adaptive Meshes for High Performance Terrain Visualization. *Computer Graphics Forum 22*, 3 (Sept. 2003), 505–514.

[CMF98] CHIBA N., MURAOKA K., FUJITA K.: An Erosion Model Based on Velocity Fields for the Visual Simulation of Mountain scenery. *The Journal of Visualization and Computer Animation 9* (1998), 185–194.

[DEJP99] DORSEY J., EDELMAN A., JENSEN H. W., PEDERSEN H. K.: Modeling and Rendering of Weathered Stone. In *Proceedings of SIGGRAPH '99* (1999), vol. 25(4) of *Computer Graphics Proceedings, Annual Conference Series*, ACM, ACM Press / ACM SIGGRAPH, pp. 225–234.

[DFM*03] DEUSSEN O., FEDKIW R., MUSGRAVE F., PRUSINKIEWICZ P., STAM J.: Ebert, D.D. organizer, Course 41: Simulating Nature: Realistic and Interactive Techniques. *Siggraph 2003 Course Notes* (2003), 1–301.

[DWS*97] DUCHAINEAU M. A., WOLINSKY M., SIGETI D. E., MILLER M. C., ALDRICH C., MINEEV-WEINSTEIN M. B.: ROAMing Terrain: Real-time Optimally Adapting Meshes. In *IEEE Visualization '97* (Oct. 1997).

[FFC82] FOURNIER A., FUSSEL D., CARPENTER L.: Computer Rendering of Stochastic Models. *Communications of the ACM 25* (1982), 371–384.

[GC99] GOBRON S., CHIBA N.: 3D Surface Cellular Automata and Their Application. *The Journal of Visualization and Computer Animation 10* (1999), 143–158.

[IFMC03] ITO T., FUJIMOTO T., MURAOKA K., CHIBA N.: Modeling Rocky Scenery Taking into Account Joints. In *Computer Graphics International* (2003), pp. 244–247.

[KMN88] KELLEY A. D., MALIN M. C., NIELSON G. M.: Terrain Simulation Using a Model of Stream Erosion. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques* (1988), ACM Press, pp. 263–268.

[KN00] KOICHI O., NISHITA T.: A Method for Modeling and Rendering Dunes with Wind-ripples. In *Proceedings of Pacific Graphics'00* (2000), pp. 427–428.

[LM93] LI X., MOSHELL M.: Modeling Soil: Realtime Dynamic Models for Soil Slippage and Manipulation. In *Proceedings of SIGGRAPH'93* (1993), vol. 27(4) of *Annual Conference Series*, pp. 361–368.

[Man83] MANDELBROT B. B.: *The Fractal Geometry of Nature*. W.H. Freeman and Comp., San Francisco, 1983.

[Mus89] MUSGRAVE, F.K. AND KOLB, C.E. AND MACE, R.S: The Synthesis and Rendering of Eroded Fractal Terrains. In *Proceedings of Siggraph'89* (1989), vol. 23(3) of *Annual Conference Series*, pp. 44–50.

[Nag97] NAGASHIMA K.: Computer Generation of Eroded Valley and Mountain Terrains. *The Visual Computer 13* (1997), 456–464.

[ON03] ONOUE K., NISHITA T.: Virtual Sandbox. In *Proceedings of Pacific Graphics'03* (2003), IEEE Computer Society, pp. 252–260.

[Rei03] REID D.: Geological Tables – Table Mountain. *The Edinburgh Geologists 41* (2003), 24–27.

[SOH99] SUMNER R. W., O'BRIEN J. F., HODGINS J. K.: Animating Sand, Mud, and Snow. *Computer Graphics Forum 18*, 1 (1999), 17–26.

[Vte] VTERRAIN: www.vterrain.org. Internet.

Figure 10: *Mesa by erosion simulation*