Eurographics Workshop on Natural Phenomena (2007)
D. Ebert, S. Mérillou (Editors)

Goblins by Spheroidal Weathering

M. Beardall, M. Farley, D. Ouderkirk, C. Reimschussel, J. Smith, M. Jones and P. Egbert

Department of Computer Science, Brigham Young U., Provo, Utah, USA

Abstract

Height map models of terrain are computationally efficient but can not represent terrain with concave surfaces. We present an algorithm for generating sandstone goblins using a simulation of spheroidal weathering. Sandstone goblins are a kind of hoodoo which are characterized by rounded concave shapes. The weathering simulation uses bubbles centered on axis aligned voxels to approximate geometry-dependent effects of spheroidal weathering. We demonstrate that the algorithm, together with appropriate surface textures, produces visually plausible goblins at near interactive speeds for most simulation parameters.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

Terrain is a critically important feature in visual storytelling for scenes that are set in the outdoors. In a story, the nature and appearance of terrain can provide powerful visual cues for mood, feeling and setting. Algorithms to support the efficient generation of terrain have been invented, investigated and commercialized [DAZ06, eoS06].

We address the problem of generating concave terrain with minimal artistic and technical effort. We focus on sandstone goblins as a starting point. Sandstone goblins are irregularly shaped features less than 10 meters high which are found, among other places, in Goblin Valley State Park, Utah. Sandstone goblins are a good starting point because the process by which they are created is understood and vegetation plays little or no role in their creation.

The use of concave terrain in movies, commercials and animation indicates that directors find this kind of terrain useful in telling a story. Sandstone goblins in particular have been used in film to represent places that are different and unusual. The sandstone goblins at Goblin Valley State Park, Utah, were used as the setting for an alien planet in the motion picture Galaxy Quest and used to represent movement west by a group of American pioneers in the nineteenth century in a recent documentary [Whi06].

With only a few exceptions, existing algorithms for generating terrain are not suitable for generating concave terrain features because they are based on height maps. A height map stores an elevation value for every point on a grid of terrain locations. Concave surfaces requires multiple elevation values for a single grid point.

(a) (b)

Figure 1: *(a) A synthetic goblin placed in front of a photograph from Goblin Valley State Park, Utah. The synthetic goblin was created using a simulation of spheroidal weathering. (b) The original photograph. (Photograph courtesy of Leping Zha, www.lepingzha.com)*

Progress has been made in concave terrain generation but more work is needed. Prior algorithms for concave terrain generation include a procedural approach [GM01] and physically-based methods which use voxel grids [IFMC03, BTHB06]. The procedural approach is designed for terrain which can be described by analytically differentiable equations. Many terrain features can not be described by such equations. The physically-based approaches mimic rock fall in jointed rock [IFMC03] and hydraulic erosion [BTHB06]. Our physical model describes a different geological process is computationally more efficient. A complete system for concave terrain feature generation would include tools from all three physically-based models.

We present an efficient algorithm for concave terrain generation based on spheroidal weathering. Spheroidal weathering is a geological process in which sharp edges and corners are reduced to rounded surfaces in exposed rock. Spheroidal weathering applied to a cube of homogeneous material results in a sphere. Visually interesting shapes are created when spheroidal weathering acts on rock which weathers unevenly.

The algorithm we describe is efficient and produces visually plausible results. In our approach, the terrain surface is represented using axis-aligned voxels of uniform size. Spheroidal weathering for "active" voxels on or near the surface is simulated by computing the ratio of air to stone within a fixed bubble-shaped volume around each voxel. The amount of decimation which occurs at each voxel is a function of that ratio and the voxel's resistance to weathering. A voxel's resistance to weathering varies within and between bedding plane layers.

The running time of the algorithm is sensitive to the number of active voxels on or near the surface and to the bubble radius. With 100,000 to 110,000 active voxels, the algorithm requires between 33 and 1891 milliseconds to complete each simulation step (on average) using bubbles of radius 1 and 7 respectively and requires only 122 MB of memory. The images used in this paper were generated using several hundred simulation steps each, contain around 100,000 active voxels in many of those simulation steps and the bubbles generally have radius 4.

The resulting geometry, when properly textured, is a visually plausible duplicate of a goblin. Figure 1 contains a goblin produced by our algorithm placed in front of a photograph from Goblin Valley. The image on the right side of the figure shows the original photograph to highlight the resemblance.

In the next section we review closely related work from geology and in both weathering and erosion models used in computer graphics. Section 3 briefly describes a geological process by which goblins may have been formed. Section 4 contains our computational model of spheroidal weathering. In Section 5, we describe how to use a tool based on our model of spheroidal weathering and in Section 6 present results from the use of that tool. In Section 7, we close with a few concluding remarks and a discussion of future work.

2. Related Work

This work is inspired by a geological model and is similar to other work on weathering and erosion as forces for changing model geometry. We differentiate between erosion and weathering in that weathering is the weakening of rock through exposure to air and water while erosion is the removal of weakened rock material. In this section, we discuss only the most relevant related work. A more complete survey of terrain generation can be found in [DL05].

2.1. Geological background

We have based our algorithm on the geological processes described in Milligan's article on the geology of Goblin Valley State Park, Utah [Mil03]. Milligan attributes goblin creation to a combination of joint systems, spheroidal weathering and differing resistances to weathering. This model is described in more detail in the next section.

Sarracino, Prasad and Hoohlo present a mathematical model of spheroidal weathering [SPH87]. Sarracino's model describes the flow of water (along with other elements) through solid rock in which the rock is not eroded in the process. In this setting, the effects of spheroidal weathering can only be observed when a rock is cut and internal weathering rings are exposed. Sarracino presents a model for the three dimensional case and solves the resulting series of differential equations for the two dimensional case.

Our model is based on counting discrete units of material to estimate curvature rather than directly using differential equations to describe curvature. Estimating curvature by counting discrete units of volume was presented in [Vic84] and shown to be linearly related to the actual curvature of a sphere in [PCG92].

2.2. Weathering

In this section, we focus on related work in weathering. Our algorithm is also related to prior algorithms on erosion of three dimensional objects, such as [VM03], and surface texturing algorithms based on cellular automata like [GC01]. The key distinctions between our work and this previous work are that we erode a 3D object globally across the entire surface rather than just at a given point or in a given direction and that we construct a triangular mesh from the cellular automata model rather than imposing a cellular automata model on an existing mesh.

Terrain modification through weathering appeared in Musgrave, Kolb and Mace's seminal paper on terrain generation using fractals [MKM89]. Musgrave creates a terrain using fractional Brownian motion (fBm) then refines the terrain through simulations of hydraulic erosion and *thermal weathering*. The thermal weathering process in particularly relevant to our work. The thermal weathering model reduces steep slopes, such as cliff faces, to talus slopes at the angle of repose by transferring material between terrain vertices. The amount of material transported is a function of the slope between adjacent vertices. Musgrave's weathering model is appropriate for mountainous terrain, but does not admit the variety of concave terrain surface features that can arise through weathering in other contexts.

Dorsey et al. present a detailed model of chemical weathering of stone objects in urban environments [DEJ*99]. In this work, stones are narrowly defined as rocks which have been shaped by humans. The model simulates fluid flow through rock and the resulting decomposition of minerals. The model includes so-called *corestone weathering* which has the same effect as spheroidal weathering. Our weathering model does not explicitly simulate the flow of water through rock and the corresponding changes in chemistry. Instead, our weathering model simulates corestone weathering, presumably including the effects of chemical weathering, directly from the geometry. Dorsey's model uses slabs of voxels, rather than axis-aligned voxel grids, in order to model complex geometry often found in human-shaped stones. Rocks in nature exhibit simpler shapes which make grids of axis-aligned voxels more appropriate.

Chen et al. use γ-ton tracing, which is similar to photon mapping, to propagate the effects of weathering processes through a scene [CXW*05]. The emphasis in this approach is correctly propagating the effects of weathering between the objects in a scene. For example, stains due to dripping water from one object to another can be convincingly modeled. After γ-ton tracing a scene, changes to the geometry can be made by removing material from incident γ-ton rays and adding material at subsequent interaction points. Our weathering model samples local geometry to compute changes to geometry rather than propagating events through the scene. This means we can not accurately model colluvium accumulation. Chen's model required 90 minutes of computation time to weather scenes with on the order of 50,000 vertices. Our algorithm is more efficient.

In summary, our weathering model is phenomenological and intended for terrain, like Musgrave's, but we allow for concave surfaces. Our model is less physically based than Dorsey's and even Chen's. This allows our implementation to run at near interactive speeds for most simulation parameter settings but we have omitted some effects included by Dorsey and Chen such as staining and colluvium deposition.

2.3. Terrain generation

Prior work on terrain generation through both erosion and weathering focuses primarily on concave surfaces modeled using height maps. This work ranges from purely procedural approaches based on fractals [PH93, GM01] to entirely physically-based models [RPP93, CMF98, IFMC03, BA05, NWD05, BTHB06] and one which combines ideas from both [MKM89]. Concave terrain features have received less attention but were addressed in [IFMC03] and [BA05]. Our model includes both randomized procedural and physically-based elements. We work on the scale of 10's of meters rather than 10's of kilometers as is done in most prior terrain generation work (except Beneš and Ito as noted below).

Gamito and Musgrave present a purely procedural approach to concave landscape generation. Gamito and Musgrave generate concave surfaces by deforming an elevation grid using a vector field [GM01]. The function which describes the deformation must be differentiable. The results for ocean waves are compelling, partly because ocean waves have smooth shapes which can be described by differentiable functions, but the results for exposed rock in canyons is less so. The aim of their work is to create visually plausible results while ignoring the actual processes that create the landscape. We also strive for visual accuracy, but use physical processes to guide algorithm design.

More recent work has included the simulation of geological processes on voxel grids. These works result in realistic terrain features in the 10's of meters range and admit concave surfaces. Beneš et al. present a terrain model based on voxel grids and an algorithm for simulating hydraulic erosion using Navier-Stokes [BTHB06]. Ito et al. simulate rock fall using gravity and rock surfaces containing joints [IFMC03]. We approach the problem using weathering rather then erosion or gravity as the primary geomorphological force and our model is also based on voxel grids. A complete tool for concave terrain would include all three processes.

3. How Goblins Form

Figure 2 contains a gallery of images from Goblin Valley which illustrate the size and shape of typical goblin formations for reference. In this section, we describe a process by which goblins may form. The following process is put forth by Milligan in [Mil03]. A definitive description of goblin formation has not yet appeared in the literature.

According to Milligan, the process that forms goblins has four major steps: deposition and lithification, fracturing, weathering, and transportation.

First, material of varying properties is deposited and is lithified, or transformed into rock. In the area of Goblin Valley, alternating beds of sandstone, siltstone, and shale have been deposited. These rock types vary in their ability to resist weathering. Together, these layers compose the geologic formation called Entrada sandstone. The interbedding at Goblin Valley suggests that the entire Entrada layer at Goblin Valley was formed while this area was a tidal flat between a large lake to the north and a sand dune desert to the south.

Next, joints (i.e. cracks) form in the rock. Joints weaken the rock and increase its susceptibility to various types of

(a) (b) (c) (d)

Figure 2: *A gallery of photographs from Goblin Valley. (a) View over the valley. Note people in bottom right corner for scale. (b) Goblin with two sandstone beds. (c) Goblin with sandstone cap and shale and siltstone base and (d) toppled sandstone cap which exposes the less resistant base to additional weathering.*

weathering. The joints in Goblin Valley frequently intersect, and divide the landscape into sections of rock that will erode to eventually become goblins.

After fracturing into joints, the dominant force by which goblins are formed is spheroidal weathering. Weathering weakens the rock. Parts of the rock erode from the surface after they have become sufficiently weak. As the rock erodes, those parts of the rock with sharp edges and corners tend to erode faster than flat surfaces. Mathematically, this can be modeled using the ratio of surface area to volume for a given portion of rock. The weathering rate, r, is proportional to the exposed surface area, a, divided by the volume v.

$$r \propto a/v.$$

To simplify the model, assume that rock is divided into indivisible cubes with unit volume which weather and erode atomically. Under this assumption, the erosion rate r is proportional to the number of exposed faces

$$r \propto \text{number of exposed faces.} \tag{1}$$

The ratio of surface area to volume at point p on a surface s is proportional to the curvature of s at p when s is convex at p and inversely proportional to the curvature at p otherwise.

Not only do different shapes of the same material weather at different rates, but material in different bedding plane layers weather at different rates as well. In the case of goblins, siltstone and shale beds erode faster than the sandstone beds. The sandstone forms the goblin caps, while the shale and siltstone form the pedestals on which goblins rest. In addition to variations in resistance between rock layers, less extreme variations exist within each layer. Such variations within layers may account for unusual rock shapes which are created in Goblin Valley. Milligan proposes that variations in the cementation of the sand grains which form the sandstone affect their resistance to erosion.

Finally, after material erodes away from the goblins, it is removed by falling to the ground, being blown away by wind, or carried away by water.

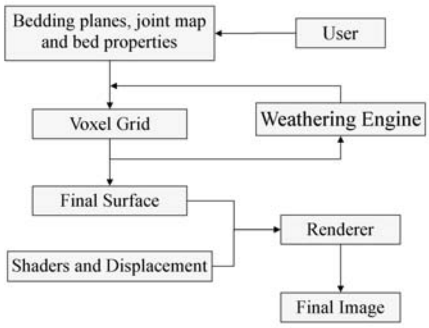


Figure 3: *Architecture for a weathering system which creates sandstone terrain features.*

4. Our Model

In this section, we present our weathering model including a description of the input provided by the user. The next section covers important implementation details.

Figure 3 summarizes our model of goblin creation through spheroidal weathering. The user provides the initial vertical joint map and bedding plane layer information including layer heights and resistance to weathering. A simple user interface for specifying the layer information is shown in Figure 4. The sliders on the left set the top-most height of each layer.

The rock begins as a column with vertical straight sides as described in the joint map. The horizontal profile (as seen looking at the rock from above) of the column is specified

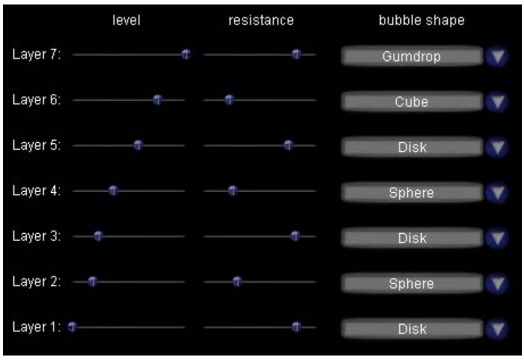


Figure 4: *User interface for specifying bedding plane layer parameters such as size, resistance to weathering and bubble shape.*

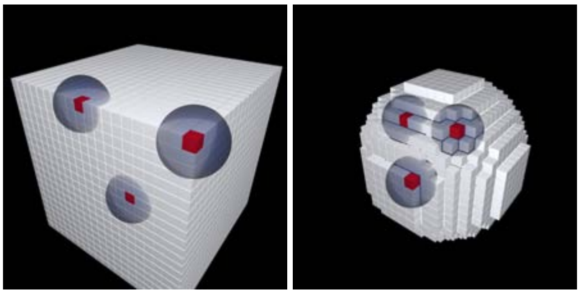


Figure 5: *Spheroidal weathering is simulated by centering a bubble on exposed voxels and computing the ratio of air to rock within that bubble. Voxels on corners have a higher ratio and weather more quickly.*

using a grayscale image in which black pixels correspond to areas with no rock.

Bedding plane layers horizontally divide the initial column into regions. The resistance to weathering in each region is set by the user as are other simulation parameters such as the bubble shape and size. Bubbles of different shapes and sizes result in different weathering effects. In nature, these effects correspond to non-uniform weathering due to factors like prevailing winds and the direction in which rain falls.

The bedding plane layer information is incorporated into a grid of axis-aligned voxels which are each the same size.

The simulation begins by initializing the decimation rate for each voxel on the surface of the rock. The decimation rate is initialized to the product of the voxel's resistance to weathering and the ratio of air to rock for the bubble centered at that voxel. Figure 5 illustrates the use of bubbles in simulating spheroidal weathering. Bubbles centered on corner voxels have a higher ratio of air to rock while bubbles centered on face voxels have a lower ratio of air to rock. The difference in ratios causes corner voxels to erode more quickly than face voxels. An example of a cube weathering into a sphere using this simulation is shown later in Section 6.

When the level of decimation for a voxel reaches zero, the voxel is simply removed. When a voxel is removed, neighboring voxels are notified so that neighboring surface voxels can increase their decimation rates as needed. Recomputing decimation rates directly using bubbles is expensive. We store the amount of increase per deleted voxel for each bubble shape and size in a lookup table to avoid recomputing decimation rates using bubbles.

The simulation proceeds until the user is satisfied with the goblin shape. When the user is satisfied with the shape, the user clicks an export button to invoke the marching cubes algorithm [LC87] to extract a surface. The surface can then be imported into a third party rendering tool for smoothing and texturing.

5. Implementation

We have implemented a terrain feature generation tool based on the weathering algorithm using C# and DirectX.

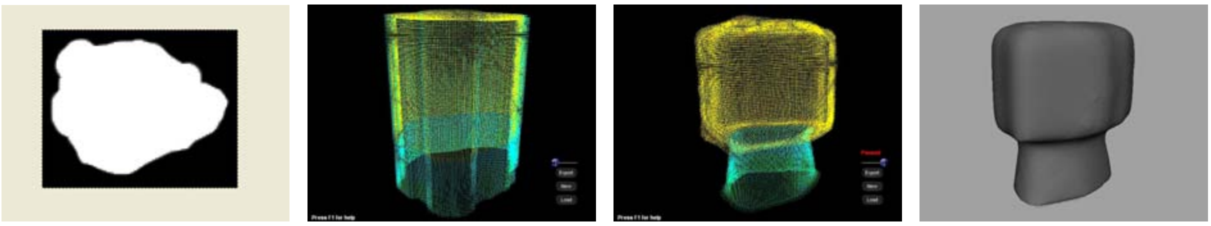
The user provides the initial rock shape, layer information and bubble parameters. The user watches the rock erode in a 3D viewer. Voxel hues in the viewer are a function of the voxel resistance to weathering with red the most resistant. When the user is satisfied with the shape, clicking export invokes the marching cubes algorithm to extract a surface which can be imported into a third party rendering tool for texturing. Figure 6 highlights the step by step process used to create a goblin.

To save memory, only active voxels are explicitly stored in memory. An active voxel is a voxel which is close enough to the surface to experience weathering. We store the voxel grid as an array of references to voxel objects which is indexed by x, y and z location. Inactive voxels all contain a reference to the same generic object.

When a voxel becomes active, a new voxel object is created and initialized with the appropriate resistance to weathering and weathering rate. The resistance for the new voxel object is based on the layer which contains the voxel and the value of a Perlin noise function [Per85] for the voxel location.

The weathering rate is computed by counting empty voxels within the appropriate bubble type. The bubble type is also determined from the enclosing layer.

The rock shape is specified using a gray scale image in which black pixels correspond to empty space. Layer information includes the thickness and resistance to weathering for each layer. The bubble parameters describe the size and shape of the bubble used to estimate curvature in each layer. We have designed a simple interface, part of which is shown



(a) (b) (c) (d)

Figure 6: *Step-by-step process of weathering simulation: (a) gray scale image used to specify initial column shape. (b) Before erosion. (c) Before exporting. (d) Smoothed in rendering program.*

in Figure 4, for specifying layer height, resistance and the bubble parameters.

An alternative approach is event-based simulation in which voxels are removed immediately as they become fully decimated. The event-based approach is more sensitive to noise in the input and runs about two times more slowly than the synchronized two-step algorithm. Of course, the synchronized two-step algorithm can be reduced to the event-based approach by reducing the time quantum between steps to match the shortest time span between erosion events. We employ a compromise which forces some synchronization to improve running time but which allows noise to appear in the final results.

For this paper, we used Maya 7.0 from Autodesk Inc. to smooth and texture the rock shapes. For completeness, we include an overview of how the goblin shapes were smoothed and textured. The surface was smoothing using the smooth vertices function. The surface texture was created by combining different frequencies of fractal motion patterns which have been compressed or stretched along a primary axis using Pixar's Renderman Artistic Tools. The model was texture-mapped instead of bump-mapped to produce a more realistic profile. The color patterns are produced in a similar manner by combining fractal motion frequencies. The layering effect in both texture and color is created by applying different shaders based on model height. All images were rendered using Pixar's Renderman Pro Server.

6. Results

In this section, we present results which demonstrate that the resulting shapes resemble natural goblins and highlight the speed at which terrain shapes can be generated.

Figure 7 contains two simulation results which objectively validate the accuracy of the weathering algorithm. Spheroidal weathering of a cube shaped rock of homogeneous material should result in a sphere-shaped rock. Figure 7 (a) shows the initial cube and (b) shows the resulting sphere-shaped rock. The cube was placed above the ground plane so that weathering would act equally on all sides. The sphere-shaped rock in (b) is actually smaller than the cube (but enlarged in the rendering to show surface texture) because we allow weathering to occur in voxels contained in the face of the cube and the weathering simulation uses an approximation of the actual curvature.

Bubbles of different sizes and shapes result in different weathered shapes. Figure 7(c) shows the shape which results from a gumdrop-shaped bubble. The gumdrop-shaped bubble has a round top and a flat bottom–much like the shape in (c). Differently shaped bubbles result in different weathering results because they skew the curvature estimate used to calculate the weathering rate. In general, a cube weathers into approximately the same shape as the bubble used in simulation.

Figure 8 contains images of two different goblin shapes created using the weathering simulation. Each goblin is shown with two different textures and without any texture to highlight the shape. The pair of connected goblins on the top row were created in a single simulation in which a thin wall connecting the goblins erodes into a thin rock band. The generation of the goblin on the bottom row is shown in real time in the video which accompanies this paper.

The total simulation time to generate each goblin shape in Figure 8 is less than 2 minutes on a 2.4 GHz Intel Core2 processor with 2 GB of RAM.

User time to generate the initial rock shape, layer and bubble settings for the shapes in Figure 8 is less than one hour in each case. The goblin shape in Figure 1 required about 3 hours of user time to generate the final input because the objective was to generate a specific shape rather than a generic plausible shape.

The running time of each simulation cycle is sensitive to the number of active voxels and the bubble size. The average simulation cycle time was measures on a 1M voxel cube, of which less than 110,000 are active at any given time, and bubbles ranging in radius from 1 to 8. The simulation cycle time is less than 0.1 second for 40,000 or fewer active voxels for all bubble radii and is less than 1 second for 110,000 or fewer voxels when the bubble radius is less than 5.

For very large models, computation time is more critical

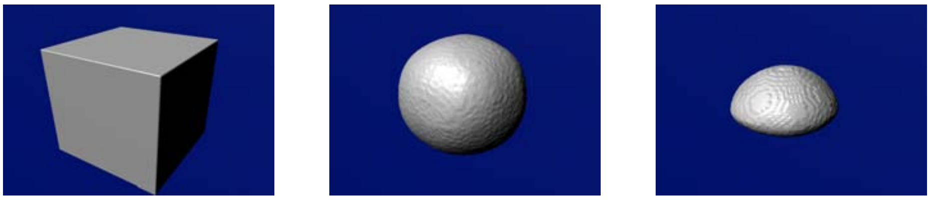


Figure 7: *Simulation results in which a cube (a) weathers into a sphere (b) and a cube weathers into a different shape (c) using a differently shaped bubble.*

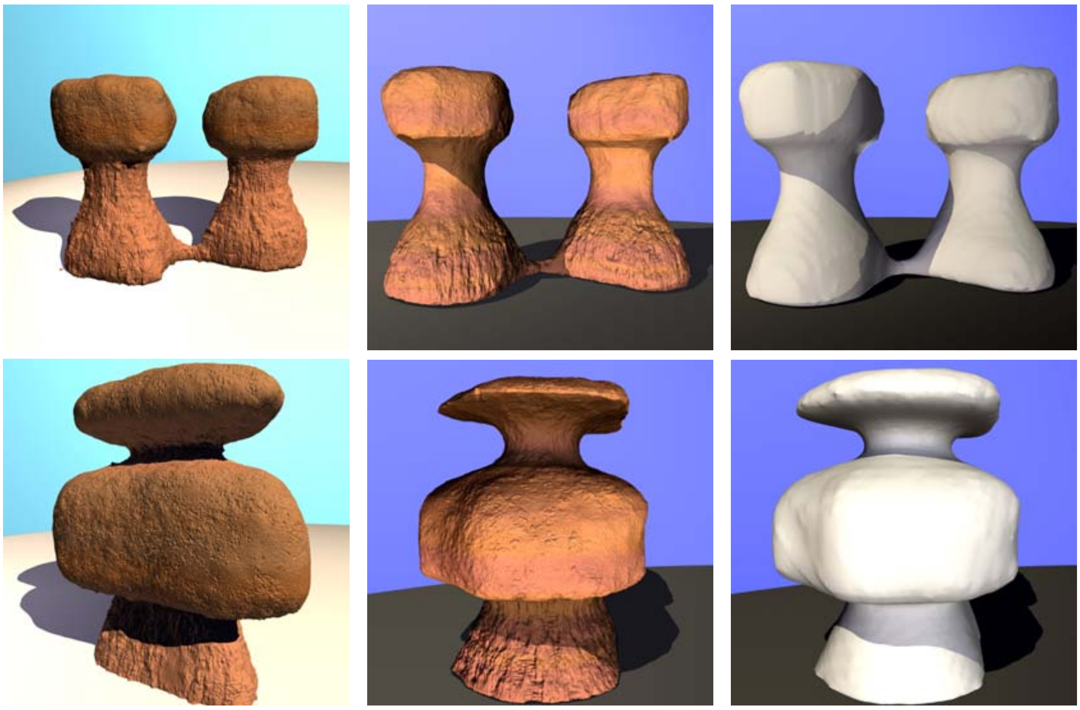


Figure 8: *Goblin shapes, with and without surface texture, created using a simulation of spheroidal weathering.*

than memory. A model containing 10M voxels with up to 840,000 active at any given time requires 793 MB of memory at peak memory use and an average of 4.1 seconds per simulation cycle with 700,000 or more active voxels and a bubble radius of 4. Weathering the 10M voxel column to nothing requires 12 minutes 33 seconds of computation time including 7 minutes to initialize data structures.

7. Conclusions and Future Work

We have presented a simulation of spheroidal weathering which can be used to produce goblin-like shapes from relatively simple input at near interactive speeds for most simulation settings. When properly textured, the shapes bear a reasonable resemblance to natural goblins.

This work is a step toward the automatic generation of concave landscape features. Rock formations with concave features can play an important part in setting the mood or feeling of a scene in film. The tool presented here, when mature, will give directors of computer generated animation more flexibility in using terrain in visual storytelling. A mature tool would include several geological processes such as hydraulic erosion, cavernous weathering and processes which involve gravity.

Surfaces extracted with marching cubes, as done here (except Figure 7), appear more angular and faceted than weathered rock. Smoothing algorithms were used to further refine the geometry. Adding more voxels to the simulation would also result in smoother surfaces.

Extending the algorithm to more efficient data structures and a multi-threaded implementation would increase the number of voxels that can be efficiently handled in simulation. Three dimensional hash tables or representations based on run-length encoding could be used to reduce memory consumption. The simulation is easily parallelized into threads because the weathering rate at each voxel depends on the location of a handful of neighbors.

References

[BA05] BENEŠ B., ARRIAGA X.: Table mountains by virtual erosion. In *EWNP'05: Proceedings of the 1st Eurographics Workshop on Natural Phenomena* (Aire-la-Ville, Switzerland, 2005), the Eurographics Association, pp. 33–39. 3

[BTHB06] BENEŠ B., TĚŠÍNSKÝ V., HORNYŠ J., BHATIA S.: Hydraulic erosion. *Computer Animation and Virtual Worlds 17*, 2 (2006), 99–108. 2, 3

[CMF98] CHIBA N., MURAOKA K., FUJITA K.: An erosion model based on velocity fields for the visual simulation of mountain scenery. *Journal of Visualization and Computer Animation 9*, 4 (1998), 185–194. 3

[CXW*05] CHEN Y., XIA L., WONG T.-T., TONG X., BAO H., GUO B., SHUM H.-Y.: Visual simulation of weathering by γ-ton tracing. In *SIGGRAPH '05: Proceedings of the 32nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2005), ACM Press, pp. 1127–1133. 3

[DAZ06] DAZ PRODUCTIONS: Bryce 6.0, 2006. 1

[DEJ*99] DORSEY J., EDELMAN A., JENSEN H., LEGAKIS J., PEDERSEN H.: Modeling and rendering of weathered stone. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer Graphics and Interactive techniques* (New York, NY, USA, 1999), ACM Press, pp. 225–234. 3

[DL05] DEUSSEN O., LINTERMANN B.: *Digital Design of Nature: Computer Generated Plants and Organics*. Springer, 2005, pp. 113–124. 2

[eoS06] E-ON SOFTWARE: Vue-6 infinite, 2006. 1

[GC01] GOBRON S., CHIBA N.: Crack pattern simulation based on 3d surface cellular automaton. *The Visual Computer 17*, 5 (2001), 287–309. 2

[GM01] GAMITO M., MUSGRAVE F. K.: Procedural landscapes with overhangs. In *Proceedings of the 10th Portuguese Computer Graphics Meeting* (2001), pp. 33–42. 1, 3

[IFMC03] ITO T., FUJIMOTO T., MORAOKA K., CHIBA N.: Modling rocky scenery taking into account joints. In *Proceedings of Computer Graphics International 2003* (July 2003), pp. 244– 247. 2, 3

[LC87] LORENSEN W., CLINE H.: Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1987), ACM Press, pp. 163–169. 5

[Mil03] MILLIGAN M.: Geology of Goblin Valley State Park, Utah. In *Geology of Utah's Parks and Monuments*. Utah Geological Association and Bryce Canyon Natural History Association, 2003, pp. 421–432. 2, 3

[MKM89] MUSGRAVE F. K., KOLB C. E., MACE R. S.: The synthesis and rendering of eroded fractal terrains. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1989), ACM Press, pp. 41–50. 2, 3

[NWD05] NEDIHOLD B., WACKER M., DEUSSEN O.: Interactive physically based fluid and erosion simulation. In *EWNP'05: Proceedings of the 1st Eurographics Workshop on Natural Phenomena* (Aire-la-Ville, Switzerland, 2005), the Eurographics Association, pp. 25–32. 3

[PCG92] PIMIENTA P., CARTER W., GARBOCZI E.: Cellular automaton algorithm for surface mass transport due to curvature gradients: Simulations of sintering. *Computational Material Science 1* (1992), 63–77. 2

[Per85] PERLIN K.: An image synthesizer. *Computer Graphics 19*, 3 (1985), 287–296. 5

[PH93] PRUSINKIEWICZ P., HAMMEL M.: A fractal model of mountains with rivers. In *Proceedings of Graphics Interface '93* (Toronto, Ontario, Canada, 1993), pp. 174–180. 3

[RPP93] ROUDIER P., PEROCHE B., PERRIN M.: Landscapes Synthesis Achieved through Erosion and Deposition Process Simulation. *Computer Graphics Forum 12*, 3 (1993), 375–383. 3

[SPH87] SARRACINO R., PRASAD G., HOOHLO M.: A mathematical model of spheroidal weathering. *Mathematical Geology 19* (1987), 269–289. 2

[Vic84] VICSEK T.: Pattern formation in diffusion-limited aggregation. *Physical Review Letters 53*, 24 (Dec 1984), 2281–2284. 2

[VM03] VARADHAN H., MUELLER K.: Volumetric ablation rendering. In *VG '03: Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics* (2003), ACM Press, pp. 53–60. 2

[Whi06] WHITNEY H.: The Mormons, Part I, WGBH/Boston, 2006. 1