ACM SIGGRAPH / Eurographics Symposium on Computer Animation (2020)
D. L. Michels (Editor)

Wind Erosion: Shape Modifications by Interactive Particle-based Erosion and Deposition

V. Krs[1], T. Hädrich[2], D. L. Michels[2], O. Deussen[4], S. Pirk[3], B. Benes[1]

[1]Purdue University, [2]KAUST, [3]Google AI, [4]University of Konstanz

Abstract

We present a novel user-assisted method for physics-inspired modeling of geomorphological features on polygonal meshes using material erosion and deposition as the driving mechanisms. Polygonal meshes defining an input scene are converted into a volumetric data structure that efficiently tracks the mass and boundary of the resulting morphological changes. We use Smoothed Particle Hydrodynamics to simulate fluids and to track eroded material. Eroded material is converted to material particles and naturally deposits in locations such as sinks and corners. Once deposited, we convert material particles back into the volumetric representation.

CCS Concepts

• ***Computing methodologies*** → *Shape modeling;*

1. Introduction

Defining and controlling natural shape morphology is still a challenging problem in computer graphics. Real-world objects undergo a large variety of morphological changes that can be modeled by humans, but this is often a tedious manual process. Nature has inspired many computer graphics areas, and many approaches exist that allow for user-controlled creation of real-world features. A good example is rendering, where physics-based algorithms are preferred over *ad hoc* approaches. However, geometric modeling still heavily relies on approaches, where the user defines an object shape by manual carving and adding material. This may be tedious, requires experienced users, and does not scale for large models or large quantity of elements.

We introduce wind erosion as a new method for physically-based modeling of geomorphological processes that uses erosion and deposition. Our focus is on the simplicity of use and interactivity. We present the following contributions: (1) the introduction of a dual SPH simulation for erosion and transportation of materials; (2) a novel erosion and deposition model that employs particles for fluids and eroded materials and a volumetric data structure for input meshes; (3) the simulation of multiple materials to approximate layered objects; (4) the introduction of means for interactively shaping and accentuating different objects.

2. Overview

An overview of our method is given in Figure 1. We start from a set of polygonal meshes representing input objects and their inner structure. Our system first converts the input mesh into a layered slab-based data structure introduced in [BF01]. This representation stores detailed information about the interface of the material and the outer environment. It also allows us to track the geomorphological changes typical for an erosion process. Each slab represents a volume with different material properties at the surface of the material, and we will call it *a material stack*. We simulate erosion using a SPH-based [Mon05] approach. Material particles are instantiated at the outmost material layers as erosion occurs, due to their interaction with other particles in the scene, and are transported. Material particles naturally deposit material in specific locations such as sinks and corners. Our system supports various materials such as rock, sandstone, soil, and different media such as wind and water. To speed up the calculation, we use an additional data structure to store voxels representing the outermost material layer of our model on the interface between the geometry and the fluid. We calculate the shear stress caused by the SPH simulation on the surface of the object. If the stress exceeds a material critical value, the material erodes. Erosion is simulated by removing the material from the affected voxel and converting it to material particles that are advected by the fluid. Gravity causes them to sink. If the material particles do not move for some time, they are deposited to the object and integrated into its material stack.

3. Stack Data Structure

Our data structure is an extension of a hybrid layered representation introduced by Benes and Forsbach [BF01]. This representation compresses materials as layers of run-length encoded voxels that we call *stacks*. The main advantage of this representation is its high compression, and the ability to represent overhangs and caves. Moreover, it can also efficiently track the mass and the boundary of

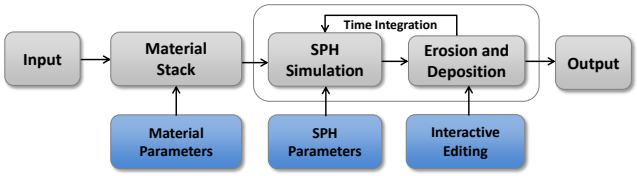

Figure 1: *Overview: given is a set of input meshes representing the volumetric object, a set of parameters for material properties, and the erosion simulation. The mesh is converted into a volumetric representation. Erosion is simulated by converting object material at the outermost layer of the object into material particles and by tracing and depositing those particles using SPH.*

morphological changes. Peytavie et al. [PGMG09] extended this representation with an implicit model that allows for constructing a smooth surface suitable for SPH collision detection.

The erosion and deposition processes occur on the boundary of the object. Therefore, we have further extended the previous work by explicitly representing the list of boundary voxels, *i.e., active voxels* that participate in the erosion and deposition process. When a new voxel is exposed, it is added to the list of active voxels, and when a voxel is deposited, the potentially covered voxels are tested for removal from the list.

The exposed voxel structure represents a volume on the surface. Note that the active voxels can also be inside the structure if there are caves present. Additionally, the active voxels data structure also contains surface vertices, *i.e.* the points where the implicit surface intersects the voxel, and the normal vectors in them. The boundary voxels need to be updated after erosion, when some voxels may disappear, and deposition, when new voxels may be created.

During the erosion and deposition simulation the SPH particles collide with the surface. In order to calculate collision response, we need to know the location of the surface and the local gradient. Therefore, we evaluate the isovalue f of the implicit model at a given point $\mathbf{p}$ as

$$f(\mathbf{p}) = \frac{1}{4\sigma^3}\left(\sum_{m\in M} V_m(\mathbf{p}) - 1\right), \tag{1}$$

where V_m is the volume of the material of type m inside a cube kernel, whose sides are 2σ. The set M includes material types except air.

Our algorithm calculates the material volume, traverses the active voxels and stacks in the neighborhood of the point, and sums volume contributions from layers in these stacks. If a slab is located on the surface, its volume contribution multiplies with the amount of material present in it. This allows for a smooth transition of the surface shape as the material erodes or deposits.

4. Particle Simulation

SPH solves the movement of the fluid (*i.e.* wind or water) by representing it as a set of independent particles that carry physical quantities, *e.g.* such as pressure and mass. We jointly simulate fluid and granular material using an SPH based approach, similar to Lenaerts and Dutré [LD09]. Figure 2 illustrates the interaction of the different particle types in our framework. Granular particles form a characteristic pile of material. Wind interacts with the granular particles and sets them in motion (left). Dust particles form a layer of material on top of the pile (middle) and react with the wind (right).

Figure 2: *Interaction of different particle types. Granular particles interacting with air (left); a layer of dust on top of a pile of a granular material (middle), and all three particle types interacting with each other (right).*

5. Erosion

Erosion occurs on the boundary of the object, and it is caused by an erosion agent such as wind or water. We capture the interaction between the fluid and the object by accumulating the fluid shear stress τ for each surface voxel. The occurrence of the stress triggers the erosion process that resolves a stack into granular and dust particles. These particles are released at the stack location and immediately interact with the particles of the fluid simulation.

5.1. Erosion Rate

The shear stress τ is calculated by using the power law [WCMT07]:

$$\tau = \theta^{0.5},$$

where θ is the shear rate expressed in terms of a velocity of the fluid $\theta = v_r/l$, where v_r is the relative velocity of the fluid and l is the distance over which the shear is applied [KBKv09]. The erosion rate ε depends on the shear stress and the critical shear stress τ_c [Par65]: $\varepsilon = \kappa(\tau - \tau_c)$, where κ is the proportionality constant. The amount of removed material in each active voxel is then calculated as

$$\frac{dm}{dt} = \varepsilon. \tag{2}$$

Note that we only need one material property τ_c that characterizes the material.

5.2. Particle Emission

The eroded material from Eqn. (2) is removed from the active voxel and converted to material particles that are advected by the fluid. The amount of material dm is converted to M particles. We assume each particle has a mass m_p so $M = dm/m_p$. The particles are emitted in disc-like clusters oriented in the reflected fluid's direction, assuming non-slip boundary. The position of the emitted particle is calculated based on the vertices on the intersection of the surface stack. The implicit surface, the velocity vector reflected over the averaged normal of the stack vertices, and a user-defined scalar value that controls how far from the surface the particles should be placed.

5.3. Surface Voxel Update

After the material removal from active voxels and particle emission, the amount of material in a voxel may reach zero. To track the morphological changes of the modeled object, we dynamically adapt the data structure. First, we search the neighborhood of the entirely eroded voxels and mark the stacks and height intervals to become exposed to air layers when these eroded voxels are removed. We then transfer the voxels, and subdivide all previously

marked stacks. The subdivision is a crucial step, as it makes sure that all the layers that are on the surface have unit height. Using these small layers on the boundary allows us to track changes in shape with greater detail and emit and deposit eroded particles with better precision. The subdivision takes a stack of layers and height intervals to subdivide as input. It first divides the stack into unit sized layers and then fuses the layers of the same material that were not marked for subdivision. For each newly exposed layer, an exposed voxel structure is created to track the amount of material.

6. Deposition

A complementary process to erosion is deposition that converts the freely moving material back to a solid volumetric object. The moving particles of material are advected by the fluid, as described in Section 4. If the movement of the fluid ceases, material particles sink to the ground and accumulate in characteristic locations.

6.1. Particle Conversion to Surface Voxels

We simulate the material deposition of the freely moving material by voxelizing the particles that do not move for a certain period of time. In this way, a new layer is deposited on the object surface. The newly deposited layer has material properties of an easily corrodible material. A fixed particle is not allowed to move freely, but it requires a certain amount of stress to be rereleased. Over time a resting particle requires increased stress to be released. This is similar to natural deposition behavior; volatile sediment particles are transported, sink, and slowly become more rigid.

6.2. Stack Update

Before a fixed particle can be deposited, we first check whether it has come in contact with the surface. We calculate the local normal vector of the contact point and check whether it points in the direction opposite to the gravity to avoid deposition on faces pointing down. If the angle between the normal and negative gravity direction is less than 45° the particle is marked as being ready for deposition.

During the deposition step, every surface's voxel neighborhood is checked for fixed particles that are ready for deposition. The amount of material represented by these particles is added to the corresponding stack, and the particles are removed from the fluid simulation. If the amount of material in an individual voxel reaches its capacity, we create a new voxel on its top and update its amount. This voxel is added to the active voxels. Finally, we update the corresponding stack and the list of surface particles. This provides us with a fine-grained level of control and lowers the number of required particles to model these phenomena.

7. Implementation and Results

We have implemented our system in C++. All experiments were conducted on a desktop computer with an Intel i7 processor at 3.4 GHz and 16 GB RAM. Results were rendered with an Nvidia Geforce 780 GPU in our graphics framework. We used OpenGL 4.2 for rendering and employed modern GPUs' compute capabilities based on CUDA for modeling.

Figure 3: *An example of sculpting by using wind. The dragon embedded in the block of stone reveals itself after being interactively eroded out by the user.*

The fast response of our system makes it possible to use for interactive and physics-based sculpting of models. Figure 3 illustrates these capabilities. The user starts with a block of material and interactively carves out the dragon model while the material deposits automatically. It is important to note that the user can also handle the deposited material, as the deposition is not immediate. This is similar to blowing away a sand pile.

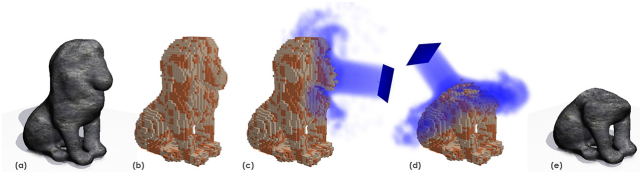

Figure 4: *A lion statue interactively shaped with our system. The input model (a) is converted into our volumetric representation (b). The user interacts with the system by controlling a wind emitter ((c)-(d)). Our system supports interactive rates allowing to immediately see the result of the modeling process (e).*

Another example of virtual sculpting by using wind is shown in Figure 4. The user interacts with the framework to shape a lion statue. The input mesh is converted to our data-structure (a) that allows us to efficiently track volumetric changes of the model (c) - (d). The system runs at interactive rates and thereby allows us to immediately see the result of the modeling process (e).

References

[BF01] BENES B., FORSBACH R.: Layered data representation for visual simulation of terrain erosion. In *Proceedings of the SCCG* (2001), pp. 80–87. 1

[KBKv09] KRIŠTOF P., BENES B., KŘIVÁNEK J., ŠŤAVA O.: Hydraulic erosion using smoothed particle hydrodynamics. *CGF 28*, 2 (2009). 2

[LD09] LENAERTS T., DUTRE P.: Mixing fluids and granular materials. *CGF 28*, 2 (2009), 213–218. 2

[Mon05] MONAGHAN J. J.: Smoothed particle hydrodynamics. *Reports on progress in physics 68*, 8 (2005), 1703. 1

[Par65] PARTHENIADES E.: Erosion and deposition of cohesive soils. *Journal of Hydraulics Division of the American Society of Agricultural Engineers* (1965), 105–139. 2

[PGMG09] PEYTAVIE A., GALIN E., MERILLOU S., GROSJEAN J.: Arches: a Framework for Modeling Complex Terrains. *CGF 28*, 2 (2009), 457–467. 2

[WCMT07] WOJTAN C., CARLSON M., MUCHA P. J., TURK G.: Animating corrosion and erosion. In *Proceedings of Eurographics Conference on Natural Phenomena* (2007), pp. 15–22. 2