

## Master of Engineering at the University of Tsukuba

# Simulating Arid Terrain With Aeolian Erosion

## 風食による乾燥地域の地形生成

CHEN ZUO

陳佐

Master's Program in Intelligent Interaction Technologies

知能機能システム専攻

Supervised by: HOSHINO Junich, NAKAUCHI Yasuhi and NOBUHARA Hajime (Division of Intelligent Interaction Technologies)

指導教員： 星野 准一, 中内 靖, 延原 肇 (知能機能工学域)

We address the problem of generating physically based arid terrain efficiently by presenting a unified framework for creating six kinds of typical arid terrain with aeolian erosion. Previous research mainly focuses on hydraulic erosion or some specific landforms by using ad hoc modeling techniques. Based on the physics of blown sand, the essential idea of our method is the following. We divided the whole simulation into three stages: Erosion, Movement and transportation/deposition. First, we combine the wind drag force with erosion model to realize aeolian erosion. Second, we add the wind parameters, buoyancy equation, thermal diffusion to position based fluids, allowing a saltation motion of sand flow. Besides, we handle the creep motion by moving the surplus solid particles to the neighbors if there are too many sand particles accumulate in one place. Finally, we simulate sand flow as open channel flow, associated with sediment transport between particles. Moreover, we propose a boundary filter to maintain the sediment concentration so that the sand flow particles could be reused and retain the process erosion all the time. The results include the final polygon mesh simulated from the volume model. Our implementation is based on GPU and simulates in an interactive framerate for scenes with up to 50,000 voxels.

keywords: terrain generation, aeolian erosion, physical based modeling

### 1 Introduction

The erosion process caused by wind is called aeolian erosion, which is the main erosion process in arid and semi-arid regions due to low precipitation and sparse vegetation in the long term. Through this procedure, diverse strange but appealing landforms take shapes gradually. Those eroded aeolian landforms such as mushroom rocks are highly suitable for expressing mood and constructing atmosphere so that they have appeared widely in various applications. They can be found in flight simulators<sup>1)</sup>, military training simulators<sup>2)</sup>, western films<sup>3)</sup>, documentary films<sup>4)</sup>, animated films<sup>5)</sup>, science fiction films<sup>6)</sup>, open world games<sup>7)</sup> and architecture design<sup>8)</sup>. Therefore, the process of aeolian erosion has a significant role in contents creation, especially when building up an arid environment.

An aeolian process has a variety of physical characteristics due to the motion of sand flow. The prevailing wind leads the shape change of rocks, which creates sharp edges and clear facets. The jumping motion of sand results in those arid terrains almost always contains overhangs or concavities. Also, for

those landforms with loose sand surface, selective erosion of wind takes more granular material than hard rocks. In actual development, those physical features are almost impossible to generate in traditional methods such as 2D heightmaps<sup>35,36)</sup>, Voronoi diagram algorithm<sup>37)</sup>. Besides, photogrammetry method is also extremely time-consuming and lack of diversity. Previous research dominantly focuses on hydraulic erosion and other ad hoc method as well. Therefore, with the purpose of generating arid terrain, physically based aeolian erosion method remains unexplored.

To solve this problem, we propose a unified aeolian erosion framework to generate six types of arid terrain. Based on physics of blown sand<sup>31)</sup>, we divided the process into three stages: erosion, movement and transportation/deposition. In the erosion stage, shear stress is replaced by a wind drag force, which accurately reflects the erosion of wind. Then in sand movement stage, we relate the prevailing wind velocity, direction, duration and vertical distribution to position based fluids, combined with buoyancy in result from the vertical distribution of temperature and thermal diffusion, allowing a more realistic saltation motion of sand flow.

Next, in the transportation/deposition stage, we simulate sand flow as open channel flow, associated with sediment transport. And we also propose a boundary filter to control the sediment concentration, allowing that the sand flow particles could be reused and maintain the erosion process all the time. The principal contributions of our research are:

- Combining wind drag force model into fluid simulation and erosion model.
- A sand movement model associating with wind velocity, direction, duration, and buoyancy.
- Boundary filtering of sediments and reuse of sand flow particles.
- A unified framework to simulate six kinds of typical arid terrain containing concavities.

## 2 Related Work

### 2.1 Overview

Terrain generation has been a significant topic in the domain of Computer Graphics (CG). Previous research mainly focuses on two directions. First is hydraulic erosion especially that caused by rivers and rainfall. Nevertheless, the erosion of wind for arid terrain remains unexplored. Second, there are some ad hoc modeling techniques, especially research about sand simulation approximates the sand movement. Although, those methods are efficient and straightforward, whereas lack of physical variations leads to the results following the same pattern. Besides, multi-fluid and multiphase fluid simulation leads to a high quality and high accuracy result while they cannot execute in an interactive framerate.

### 2.2 Terrain Generation with Hydraulic Erosion

As an essential topic, terrain generation has been researched for more than thirty years. The fundamental work<sup>38)</sup> proposed the fractal terrain which is a kind of random generated technique using mid-point displacement algorithm with a 2D heightmap. The disadvantage of fractal terrain is the deficiency of realistic characteristics like erosion.

After that, hydraulic erosion was proposed initially in 1988 by Kelley<sup>9)</sup>, who advanced a procedural method to generate watersheds. Hydraulic erosion became widely researched since then. These methods<sup>11)12)13)14)15)30)</sup> mainly focus on the valleys and mountains eroded by river network and rain. While terrain in arid climate cannot be generated through them, we still borrowed

some ideas such as splitting the whole process into multiple independent steps, choosing volume representation and layered structure to approximate realistic strata structure.

Then physically based method became popular, and Musgrave et al.<sup>10)</sup> nominated two crucial work. First is a hydraulic erosion model involves sediment transportation and deposition of rain. Second, They introduced the thermal weathering process which is a fast and straightforward way to deal with all sorts of material loose. The first method is further developed by Kristof et al.<sup>16)</sup>, which combined smoothed particle hydrodynamics method with terrain generation, and they proposed the advection equation in SPH and donor-acceptor advection scheme which solves sediment transportation quite efficiently. We extended this method with a new erosion model which is described in Section 4. Also, the second one cannot generate complex concavities, but it can still be used in our research as a part of sand flow simulation, and we explained it in Section 3.

Moreover, like other parallel erosion methods<sup>18)19)</sup>, our simulation runs on GPU as well, and the detailed architecture is explained in Section 6.

### 2.3 Terrain Generation with Ad Hoc Techniques

Additionally, there is some inspiring research about particular features of the terrain. Ito et al.<sup>20)</sup> used phenomenological techniques to simulate mountains full of joints. Benes et al.<sup>21)</sup> proposed a two-layer model that with different erosion effect to generate table mountains, which is a typical landform of the arid environment. Jones et al.<sup>22)23)</sup> performed a method combining cavernous weathering and spheroidal weathering with curvature estimation so that rocks can be worn out to mushroom rocks. Methods as mentioned earlier only focus on one or two specific kinds of landforms, and they are not physically based methods, so the diversities of the visual results are limited.

There is some research mainly focus on the simulation of sand. Onoue and Nishita<sup>24)</sup> described an efficient method to simulate sand dune and wind-ripple. Benes et al.<sup>25)</sup> proposed the model of material accumulation, which also works in the creation of wind-ripple. Moreover, another research<sup>26)</sup> put out a more complex physical model with plants for wind-ripple evolution. Those three studies focus one specific landform as well, but the influences caused by wind is

considered, which is very inspiring to our research and we discussed a more complex sand flow simulation in Section 3.

### 2.4 Multi-Fluid & Multiphase Fluid Simulation

Liu et al.<sup>27)</sup> combined the dust simulation with aerodynamics and synthesized the animation of the sandstorm. While it's not a terrain generation research, the sandstorm model is very inspiring to our study.

Besides, some multiphase fluid simulation studies<sup>28)</sup><sup>29)</sup> described how sand and water could be coupled together. Although the result is stable and excellent, the computational cost is too high for the real-time application. For more controllability, we simulate the process in an interactive framerate.

## 3 Sand Flow Simulation

### 3.1 Overview

In arid regions, lack of rain leads to typical features such as sparse vegetation, furthermore lack of soil moisture and a large amount of unconsolidated sediments. Then They become important agents to wind activity, and those sand grains carried by prevailing wind shape the land surface as well through continuing impact and wearing out. Therefore, the complexity of sand flow motion is that it has characteristics of both air and sand grains.

### 3.2 Sand Movement

When the wind velocity exceeds a critical value, the sand grains on the ground start to move and even fly with the wind. Based on different wind velocity, different diameters and mass of sand grains, the motion of sand flow varies in a diversified way. It can be categorized into three typical patterns referring to Bagnold's book<sup>31)</sup>: creep, suspension and saltation, as shown in Figure 1.

First, the large sand grains are too heavy to be lifted by the wind, so they can only slide or roll along the land surface, which is called creep. Creeping sand grains account for about 25% of the total sand particles in transport. With both high mass and high momentum, even at a low speed, those sand grains can bring about more fly sand grains and change the sand surface of the ground by the impact.

Second, the small-sized sand grains are basically dirt and dust, which are so light that the wind can easily raise them. Suspension occurs when the wind blows them

into the air then they cannot deposit on the ground for a long time. Instead, they just drift horizontally following the wind. Those grains are around less than 5%, so the eroded effect of them is negligible.

Third, Saltation is the major fraction of sand movement which occupies more than half of total sand quantity. On account of drag force or collision, sand grains are thrown into the air, and their momentum continuously increases when following the air flow. After a while, the energy attenuates, and those sand grains are pulled downward by gravity correspondingly. Then the grains may hit other sand grains to trigger new saltation or creep, which could lead to significant damage to the terrain surface.

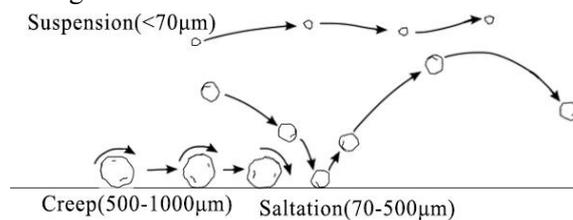


Figure 1 Sand Movement

### 3.3 Fluid Simulation

In this paper, we consider the sand flow as fluid and simulate it by position based fluids<sup>39)</sup>. The saltation could be realized combining with buoyancy equation. When sand grains are piled too much, gravity makes them collapse, which could be seen as a sort of creep<sup>32)</sup>.

#### 3.3.1 Navier-Stokes Equation

Navier-Stokes equation plays a key role in fluid simulation. To simulate the sand flow, here we choose the Lagrangian approach to approximate the Navier-Stokes equation

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v} \cdot \nabla) \mathbf{v} - \frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \Delta \mathbf{v} + \mathbf{f}, \quad (1)$$

with respect to the incompressibility constraint

$$\nabla \cdot \mathbf{v} = 0. \quad (2)$$

$\mathbf{v}$  is the velocity vector field,  $t$  the time step,  $p$  the pressure scalar field,  $\rho$  the mass density scalar field,  $\mathbf{f}$  the external forces and  $\mu$  the viscosity.

The advantage of Lagrangian approaches is the particles are not necessarily full of the simulation domain which can reduce overhead.

#### 3.3.2 Smoothed Particle Hydrodynamics

In order to approximate the Navier-Stokes equation (1) and its incompressibility constraint (2), we select the

meshfree method SPH to handle it. The method solves regular fluid parameters like density and pressure by interpolating among discrete particle positions, which speed up the calculation process significantly. By using the following formula, any scalar physical quantity  $A$  at position  $\mathbf{r}$  could be interpolated from the neighboring particles:

$$A_s(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h), \quad (3)$$

where  $m_j$  is the particle mass and  $\rho_j$  is the fluid density. The term  $W$  denotes a smoothing kernel around the particle  $\mathbf{r}$  with radius  $h$ . When  $h \geq 2\Delta s$  the simulation is able to maintain stability, where  $\Delta s$  denotes particle spacing. When dealing with different quantity, there are different kinds of kernel functions.

In SPH method, the simulation would become unstable when particles do not have enough neighbors. The general way to address this problem is to take small time steps to avoid density estimate inaccuracy. Otherwise, the quantity of particles has to be improved sufficiently to avoid the unstable situation, at a high computational cost.

### 3.3.3 Position Based Fluids

By introducing the Position Based Dynamics (PBD) framework, the incompressible flow simulation can be unconditionally stable and robust. PBD method solves the fluid dynamics system with Gauss-Seidel iteration. Instead of updating position after force solving, it solves position first then derives momentum changes implicitly, which avoids the instabilities related to the explicit methods. And the fluid simulation algorithm within the PBD framework is called position based fluids.

In PBD, dynamics system is solved by constraint projection. Given point position  $\mathbf{p}$ , the constraint function is written as  $C(\mathbf{p} + \Delta\mathbf{p}) = 0$ , which can be approximated by

$$C(\mathbf{p} + \Delta\mathbf{p}) \approx C(\mathbf{p}) + \nabla_{\mathbf{p}} C(\mathbf{p}) \cdot \Delta\mathbf{p} = 0. \quad (4)$$

Position correction  $\Delta\mathbf{p}$  is what we want to find to update positions. Restricting  $\Delta\mathbf{p}$  to be at the same direction with  $\nabla_{\mathbf{p}} C(\mathbf{p})$  then

$$\Delta\mathbf{p} = \lambda \nabla_{\mathbf{p}} C(\mathbf{p}). \quad (5)$$

Associating with kernel function of SPH, the position update can be represented as

$$\Delta\mathbf{p}_i = \frac{1}{\rho_0} \sum_j (\lambda_i + \lambda_j) \nabla W(\mathbf{p}_i - \mathbf{p}_j, h), \quad (6)$$

$$\lambda_i = -\frac{C_i(\mathbf{p}_1 \cdots \mathbf{p}_n)}{\sum_k |\nabla_{\mathbf{p}_k} C_i|^2 + \varepsilon}.$$

where  $\varepsilon$  is a customized relaxation parameter,  $\lambda$  the scaling factor and  $\nabla W$  denote the gradient kernel function.

### 3.3.4 Buoyancy

Since the high transmittance<sup>40)</sup> of solar radiant energy passing through the air, the ground is heated more quickly than air. The difference in temperature among different height become an important factor of buoyancy. Hence, we introduce the buoyancy and thermal diffusion to calculate the temperature change.

The buoyancy equation is as follows,  $\beta$  is the buoyancy coefficient,  $T_{ambient}$  the ambient temperature.

$$\mathbf{F}_{buoyancy} = \beta(T - T_{ambient})\mathbf{y}, \quad (7)$$

Besides, air near the ground is heated more rapidly than air midair. We set a heat zone which is able to heat particles when they close to the ground. And the particles keep losing temperature after they rise from the ground. By this method, we can also approximate the jumping motion of sand saltation.

Next, the uneven temperature distribution would cause thermal diffusion. With the SPH's kernel function, it can be written as follows:

$$\frac{DT}{Dt} = a \nabla^2 T = -a \sum_j m_j \frac{T_j + T_i}{\rho_j} \nabla^2 W_{viscosity}(\mathbf{r}, h). \quad (8)$$

$a = \lambda / c_p \rho$  is the thermal diffusivity.

### 3.3.5 Vorticity Confinement

Turbulent motion of wind is a vital factor of aeolian erosion. Additional damping added by PBD weakens the effect, which can be fixed by introducing the vorticity confinement. At first, it requires calculating the vorticity at each position of particles like this:

$$\omega_i = \nabla \times \mathbf{v} = \sum_j \mathbf{v}_{ij} \times \nabla_{\mathbf{p}_j} W(\mathbf{p}_i - \mathbf{p}_j, h), \quad (9)$$

where  $\mathbf{v}_{ij} = \mathbf{v}_j - \mathbf{v}_i$ . Then the corrective force can be obtained by

$$\mathbf{f}_i^{vorticity} = \varepsilon(\mathbf{N} \times \omega_i), \quad (10)$$

with the vector  $\mathbf{N} = \frac{\eta}{|\eta|}$ , where  $\eta = \nabla|\omega|_i$ .

### 3.3.6 Wind Velocity

In addition to the fluid simulation, we add the horizontal wind velocity to the original velocity to simulate the motion brought by global climate. In the planetary



Figure 2 Six Types of Arid Terrain

boundary layer, vertical distribution of horizontal mean wind speeds is in accordance with the log wind profile which accounts for surface roughness and atmospheric stability.

$$\mathbf{v} = k(\mathbf{v} + \mathbf{v}_{wind}) \cdot \lg\left(\frac{y}{y_0}\right), \quad (11)$$

where  $k$  is the wind velocity coefficient, and we set it to 10. The  $y_0$  is surface roughness, and here is 0.2.

### 3.3.7 Boundary Filter

Simulating sand flow over a large-scale terrain with PBF method appears unfeasible on current computers. To account for this problem, the sand flow particles are constrained in a limited area and simulated as recycled open channel flow. When sand flow particles cross the boundary, they are reused on the other side.

To avoid that the erosion process stopped due to saturated sand flow particles, we set a boundary filter, which is used for moving particles from one side to the opposite side and at the same time clear the sediments of  $q\%$  sand flow particles. As shown in Figure 3, here we set  $q$  to 60. The yellow particles are particles carrying sediments.

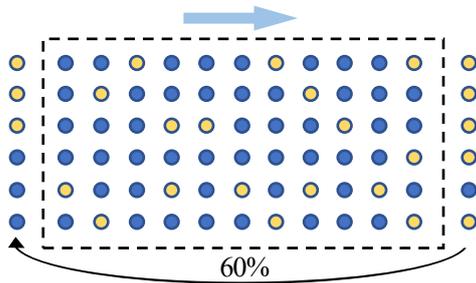


Figure 3 Boundary Filter

## 4 Aeolian Erosion

### 4.1 Overview

For sand grains, the wind-driven process is precisely the process of lift, transport, and deposit. Also, that process could be seen as erosion for the land surface. While the variations of landforms caused by water are generally

more notable than sand flow, the persistent wind driven geological process takes primary roles in an arid and semi-arid climate, and it's what we called aeolian erosion.

Aeolian erosion contains two sorts of the process: abrasion and deflation. The sand grains, lifted into the air during the saltation or dragged by wind in the creep, collide with exposed rock or sand surface and keep wearing down the surface. Dreikanter is a typical landform as a result of the abrasion process. In the process of wind deflation, those ground surface with a great number of loose sand grains or soil particles is easily eroded by lift force of the wind. The deflation hollow is a representative landform of the deflation process.

### 4.2 Drag Force

When carrying out a force analysis on a sand particle in saltation, the forces acting on it are the drag force of air, the Magnus force, the Saffman force and the gravity force[36]. But the most important forces are drag and gravity.

$$F_g = -mg = -\frac{1}{6}\pi D^3 \rho_{sand} g, \quad (12)$$

$$F_d = \frac{1}{2} C_d A \rho_{air} v_r \mathbf{v}_r, \quad (13)$$

where  $\mathbf{v}_r$  is the relative velocity between the sand grains and air particles.  $D$  is the diameter of the spherical sand particle.  $A = \frac{\pi}{4} D^2$  is the area of the cross-section of the sand grain. Moreover,  $C_d$  is the coefficient of drag which can be represented by a function of  $Re_r$ . According to the empirical formula of White[39]:

$$C_d = \frac{24}{Re_r} + \frac{6}{1 + \sqrt{Re_r}} + 0.4, \quad (14)$$

where the Reynolds number  $Re_r = \frac{v_r D}{\nu}$ , and  $\nu$  is the kinematic viscosity of air, here we set  $\nu = 1.46 \times$

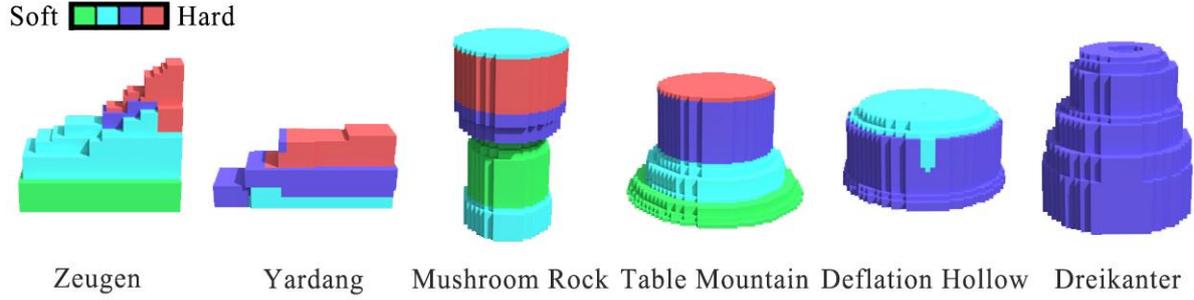


Figure 4 Six Types of Terrain Models

$10^{-5}m^2/s^{33}$ .

### 4.3 Erosion Model

When sand flow hit the landform, we consider the force as shear stress

$$\tau_d = F_d/A = \frac{1}{2}C_d\rho_{air}v_r\mathbf{v}_r, \quad (15)$$

then the shear stress erosion equation can be used, and the erosion mass is as follows:

$$S_e = \sum_j K_s(\tau - \tau_d). \quad (16)$$

The corresponding air particles conserve the eroded sediments. If the initial mass of the solid particle has been eroded completely, it would be removed when being written into volume texture.

### 4.4 Transportation

We see sand flow as sediments suspended in the air, so it moves following the velocity of the wind. This transport of sand particle is usually represented by advective quantity  $M_{adv} = -\mathbf{v}_s \cdot \nabla S$ .  $S$  is the sediment mass,  $\mathbf{v}_s$  the sediment settling velocity. According to White<sup>34</sup>, they proposed a donor-acceptor advection scheme which implements material advection in SPH method.

$$M_{adv} = -\sum_j \begin{cases} m_j \frac{S_j}{\rho_j} (\mathbf{v}_s \cdot \hat{\mathbf{r}}_{ij}) F(|\mathbf{r}_{ij}|, h), & \mathbf{v}_s \cdot \mathbf{r}_{ij} \geq 0 \\ m_i \frac{S_i}{\rho_i} (\mathbf{v}_s \cdot \hat{\mathbf{r}}_{ij}) F(|\mathbf{r}_{ij}|, h), & \mathbf{v}_s \cdot \mathbf{r}_{ij} < 0 \end{cases} \quad (17)$$

where  $\nabla W(\mathbf{r}_{ij}, h) = \hat{\mathbf{r}}_{ij} F(|\mathbf{r}_{ij}|, h)$ ,

$$\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j,$$

$$\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/|\mathbf{r}_{ij}|$$

and  $F$  is the derivative of the kernel function taken with respect to  $|\mathbf{r}_{ij}|$ .

The equation shows when the relative position is in the same direction as the advection vector  $\mathbf{v}_s$ , donor particle pass on the sediment to acceptor particle. And the settling velocity for sand particles is as follows:

$$\mathbf{v}_s = \frac{2}{9} r_s^2 \frac{\rho_s - \rho_a}{\mu} \mathbf{g} f(S), \quad (18)$$

where  $\rho_s$  is the sand density,  $\rho_a$  the air density,  $r_s$  the radius of the sand particle,  $\mu$  the viscosity of the air. If the sediment concentration is higher than the advection rate is lower, which is represented by the hindering settling function.

$$f(S) = \begin{cases} 1 - \left(\frac{S}{S_{max}}\right)^e, & S \leq S_{max} \\ 0, & S > S_{max} \end{cases} \quad (19)$$

where  $S$  is the solid volume fraction at acceptor particle and  $S_{max}$  is the maximum concentration in the sand flow particle. And we set the exponent  $e$  to 4.

### 4.5 Deposition

When the drag force is not sufficient to maintain the sand concentration in transportation. Then part of sand grains lost speed and fall down to the ground. It's what we called deposition. And in the simulation, if a sand flow particle gets close enough to existing solid particles or boundary particles, it deposits on them then the new solid particle generates. This equation can represent the process:

$$S_d = \sum_j \frac{\rho_s}{\rho_j} (M_{adv} - M_c), \quad (20)$$

where  $M_c$  is the critical sediment mass for deposition, and  $\rho_s$  is the density of sediment material.

The newly deposited strata of solid particles are the softest layer in our simulation, which makes them very quickly to be eroded again.

### 4.6 Creep

Compared to the fluid simulation, creep is closer to a rigid-body motion. In our research, we did not divide the particles into different-sized groups because the multi-fluid simulation is with high overhead. Referring to the research by Wang<sup>33</sup>, we consider this situation as a kind of creep, that sand grains accumulate too much at one

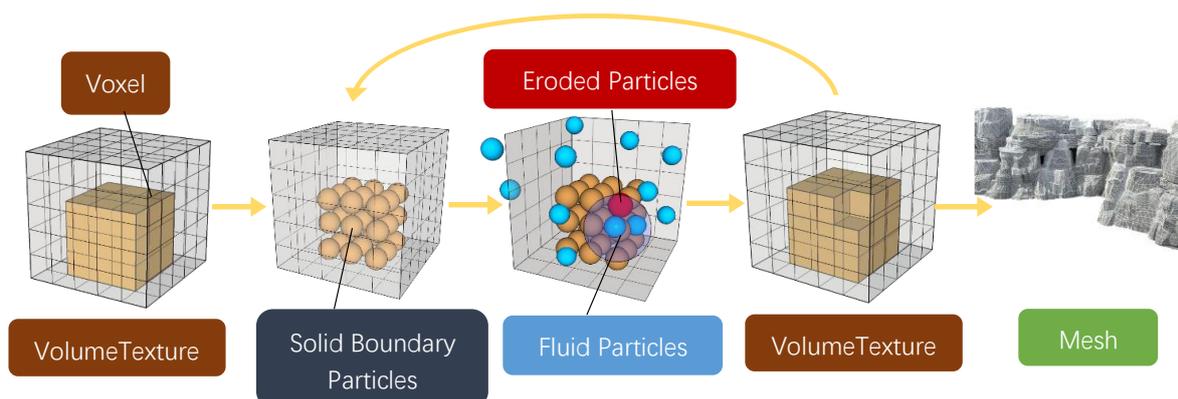


Figure 5 Architecture Overview

position and the slope angle get over the talus angle then it collapses and shape changes. We simulate the process following this equation:

$$y_i = \begin{cases} y_i + \alpha(y_j - y_i - T_a) & y_j - y_i > T_a \\ y_i & y_j - y_i \leq T_a \end{cases} \quad (21)$$

## 5 Arid Terrain

### 5.1 Overview

As shown in Figure 2, we focus on six types of arid terrain in this research: zeugen, yardang, mushroom rock, table mountain, deflation hollow and dreikanter, which are all prominently shaped by aeolian erosion. There do exist many other landforms in an arid environment such as canyon, alluvial fan, sand dune, but they are either associated with one or more non-aeolian process like tectonics, glacial and alluvial forces, or occurs on a tiny scale that cannot be treated within the same framework.

Zeugens look like ridges in deserts, which has a layer of resistant rock underlain by a layer of weak rock. Wind abrasion process turns desert surface into ridges and furrows. With time the furrows are gradually widened and the hard rocks remain and become zeugens. As the abrasion continue to act on them, they get lower and consequently they are undercut and worn away.

The appearance of yardangs is similar to zeugens, but they are smaller. Yardangs are ridges formed on vertical strata while zeugens are on horizontal strata. The resistant rocks are developed and lie parallel to the direction of the prevailing wind. And these landforms are usually undercut on their windward sides.

A Mushroom rock is also called rock pedestal, hoodoo. Its tower-like shaped features formed from horizontal bands of differing hardness rocks and abrasive action of wind. Continuing erosion results in the softer parts being worn away at a more rapid rate

resulting in the formation of mushroom shapes. We model it with a changing direction wind and horizontal strata.

Table mountain is also called mesa. It's a kind of flat-topped tableland with one or more steep sides, common in the Colorado Plateau regions of the United States. The harder rock in a table mountain act as flat protective caps for the underlying rocks. The sediments eroded is accumulated around the bottom.

Deflation hollows are where loose sand grains are blown away by wind then the land surface sinks to a level in which soil particles are too heavy to be lifted or underground water exist. That armored surface stabilizes the sand and prevents the surface being lowered further.

Dreikanter is a type of ventifact that bears the abrasion in the long term so that characteristically it forms a pyramidal appearance with sharp edges and mostly three wind-abraded facets. In the early stage, the shape is flattened and polished in the direction of the prevailing wind. It changes the mass distribution, resulting in the rock turns another surface toward the wind. The process continues, new faces generated.

### 5.2 Terrain Model

To realize overhangs and concave terrain, we choose volume texture as input data structure so that the strata information can be conserved as voxels. Moreover, the characteristics of wind are considered in this model as well.

As shown in Figure 4, A zeugen is modeled with horizontal strata of different hardness, and the wind keeps changing. Next, a yardang is modeled with a fixed direction wind and vertical strata. Then we model the mushroom rock with a changing direction wind and horizontal strata with a hard top layer. And the model of



Figure 6 An Overview of Various Aeolian Eroded Landforms



Figure 7 The Simulation Results of Six Sorts of Landforms

a deflation hollow includes changing direction wind and shallow horizontal strata, the center of which is a soft layer. As for the dreikanter, instead of rolling it based on rigid body physics, we model it with three fixed directions consistent wind and high resistant hardness.

## 6 Implementation Details

### 6.1 Architecture Overview

As shown in figure 5, the whole process is executed on GPU so that it can simulate in a real-time framerate. And the pseudocode is showed in the appendix. In the beginning, the strata data is edited in MagicaVoxel, which is an opensource 3D volume modeling software. Then the initial terrain is imported as a volume texture, where every voxel is defined as a 32bit integer with different resistance to erosion. Layer 0 is the layer of sand, which is also most easily eroded.

Then it is transformed into solid particles so that the terrain can interact with fluid particles. At this time, sand flow is simulated by a fluid simulation method and erode the solid particles procedurally. In order to speed up the nearest neighbor search in fluid simulation, we

choose the bitonic sorting algorithm to sort hash grid of all the particles. Bitonic sorting is a parallel algorithm particularly suitable for GPU particles, whose complexity is  $O(n \log^2(n))$  with a delay of  $O(\log^2(n))$ .

During the aeolian erosion, completely eroded solid particles are marked so that they are removed when transformed back to the volume texture. A user can pause the process in any time to finally generate the polygon mesh from the volume texture.

We choose the Marching Cubes method to build the meshes. In this method, every voxel in volume texture represents a cube corner either being inside or outside the volume. The value of each 8 voxels decides the cube type, which is a case index with 256 possible values. Then the newly generated vertices lie on the 12 edges of the cube. By a lookup table, the triangle indexes can be found. And this simulation is based on the volume texture so the vertex is exactly in the middle of the edge. Unity3D helps us recalculate the vertex normal, which significantly smooths the whole surface.



Figure 8 Rendering Result of Zeugen



Figure 11 Rendering Result of Table Mountain



Figure 9 Rendering Result of Yardang



Figure 12 Rendering Result of Deflation Hollow



Figure 10 Rendering Result of Mushroom Rock



Figure 13 Rendering Result of Dreikanter

## 6.2 Specification

Our simulation ran on an NVIDIA GTX 980M, Intel Core i7-4790K 4.00GHz, and each stage of the solver is fully parallelizable. It's implemented with compute shader in Unity3D. The input volume data is edited in Magicavoxel. Then we transferred the .vox file to volume texture. We generated 10,508 boundary

particles, 7,320 sand flow particles, 35,109 solid particles, and the frame rate is around 38 fps. We hashed more than 50,000 particles into a 3D grid then do the bitonic sorting algorithm every iteration. The grid we got can be used in the nearest neighbor searching. Finally, we choose marching cubes to generate polygon meshes.

## 7 Results

The program restricted the simulation region to the size of  $100 \times 50 \times 100$  containing 500,000 voxels in the volume texture.

Figure 6 shows an overview of various aeolian eroded landforms.

Figure 7 shows the simulation results of all Six sorts of landforms.

Figure 8 shows the rendering result of a zeugen.

Figure 9 shows the rendering result of a yarding. The images demonstrate the significant visual difference between vertical strata and horizontal strata.

Figure 10 contains the rendering of a mushroom rock whose shape is eroded unevenly and the sharp edges and clear facets demonstrate the features of wind.

Figure 11 is the rendering result of a table mountain, which present a visual appearance with eroded top and deposited bottom.

Figure 12 presents the simulation effect on the loose sand surface which caused a deflation hollow.

Figure 13 shows the simulation of fixed direction persistent wind abrasion is able to create a dreikanter.

## 8 Conclusion

We have presented a unified aeolian erosion method that can handle Six kinds of arid terrain with different strata settings. With fluid simulation and wind parameters, the typical features like sharp edges and clear facets caused by prevailing wind can be generated by our approach. Besides, with the buoyancy equation and thermal diffusion, it is possible to simulate the saltation of sand, which is very helpful to create vertical variations of concave rocks. Combined with the volume structure of landforms, it can be proved that our approach significantly simplifies the generation of arid terrain. By choosing the position based fluids algorithm, we can also simulate in an interactive framerate, which improves controllability as well.

## 9 Limitations and Future Work

Although the deposition of sand has been accomplished in present simulation, this research mainly focuses on landforms eroded by abrasion and deflation process so the deposited landforms like sand dunes cannot be handled. Besides, in the arid environment there exist lots of landforms that related to other complex geological or climate factors. Therefore, in the future, we hope to find a unified and concise way to deal with those landforms.

Another limitation of the present framework is the simulation scale. The sand ripples generated within the wind field affected by existed obstacles haven't been implemented. It's because present resolution cannot support that kind of tiny-sized landform, but it can be created combined with a traditional 2D heightmap.

## 10 Acknowledgments

I would like to thank my professor Junichi Hoshino who gave me the opportunity to challenge this topic, and I came to know about so many new things during this process.

Secondly, I want to thanks OB Daiki Satoi, Hiroaki Iida and my friend Jianxiong Hu, Yufan Zhou. They gave me much valuable advice, which helped me overcome some difficulties from research methodology to specific programming practice.

## 11 Appendix

### Algorithm 1 Simulation Loop

---

```

1: for all voxels  $i$  do
2:   solidParticles.Add( $i$ )           ▷ volume texture to solid particles
3:   find neighboring voxels  $j$ 
4:   if  $i_{height} - j_{height} > \tan(\text{talusangle})$  then
5:     solidParticles.Add( $i_{height+1}$ )           ▷ apply creep
6:   end if
7: end for
8: for all fluid particles  $i$  do
9:   calculate  $f_{buoyancy}$ 
10:   $v_i \leftarrow v_i + v_{wind} + \Delta t f_{ext}(x_i)$            ▷ apply forces
11:   $x_i^* \leftarrow x_i + \Delta t v_i$            ▷ predict postions
12: end for
13: for all fluid particles  $i$  do
14:   find neighboring fluid voxels  $j$ 
15:   calculate  $\lambda_i$ 
16:   calculate  $\Delta p_i$ 
17:   $x_i^* \leftarrow x_i + \Delta p_i$            ▷ update postions
18:  apply vorticity confinement
19: end for
20: for all fluid particles  $i$  do
21:   find neighboring solid voxels  $j$ 
22:   calculate  $f_{drag}$ 
23:   calculate eroded sediments  $S_e$            ▷ apply aeolian erosion
24:   calculate transported sediments  $M_{adv}$  ▷ apply sediment transportation
25:   calculate deposited sediments  $S_d$        ▷ apply sediment deposition
26: end for
27: for all fluid particles  $i$  do
28:   find neighboring solid voxels  $j$ 
29:   calculate temperate  $T$            ▷ solve thermal diffusivity
30: end for
31: for all solid particles  $i$  do
32:   remove eroded particles
33:   $\text{voxel} \leftarrow i$            ▷ solid particles to volume texture
34: end for
35: if generate polygon mesh then
36:   apply marching cubes
37:   return 0
38: end if

```

---

## 12 References

- 1) IPACS, "Aerofly FS 2 - USA Utah on Steam," 2017.  
[https://store.steampowered.com/app/611210/Aerofly\\_FS\\_2\\_USA\\_Utah/](https://store.steampowered.com/app/611210/Aerofly_FS_2_USA_Utah/).
- 2) MetaVR, "VRSG Visuals in JTAC Simulation," 2017.

- [http://www.metavr.com/downloads/MetaVR\\_JT\\_AC-Brochure.pdf](http://www.metavr.com/downloads/MetaVR_JT_AC-Brochure.pdf).
- 3) Sergio Leone, "A Fistful of Dollars," Wikipedia. 1964.
  - 4) National Geographic, The Majesty of Monument Valley | National Geographic. 2010.
  - 5) Pixar Animation Studios, "Boundin'," Wikipedia. 2003.
  - 6) Rockstar Games, "Red Dead Redemption," Wikipedia. 2010.
  - 7) Ridley Scott, "The Martian," Wikipedia. 2015.
  - 8) M. Larsson, "Dune: Arenaceous Anti-Desertification Architecture," in *Macro-engineering Seawater in Unique Environments: Arid Lowlands and Water Bodies Rehabilitation*, V. Badescu and R. B. Cathcart, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 431–463.
  - 9) A. D. Kelley, M. C. Malin, and G. M. Nielson, "Terrain Simulation Using a Model of Stream Erosion," *ComputerGraphics*, vol. 22, no. 4, p. 6, Aug. 1988.
  - 10) F. K. Musgrave, C. E. Kolb, and R. S. Mace, *The Synthesis and Rendering of Eroded Fractal Terrains*. 1989.
  - 11) K. Nagashima, "Computer generation of eroded valley and mountain terrains," *The Visual Computer*, vol. 13, no. 9–10, pp. 456–464, Jan. 1998.
  - 12) N. Chiba, K. Muraoka, and K. Fujita, "An erosion model based on velocity fields for the visual simulation of mountain scenery," *Journal of Visualization and Computer Animation*, vol. 9, pp. 185–194, 1998.
  - 13) B. Beneš and R. Forsbach, "Visual Simulation of Hydraulic Erosion," p. 8, 2002.
  - 14) B. Beneš and R. Forsbach, "Layered Data Representation for Visual Simulation of Terrain Erosion," in *Proceedings of the 17th Spring Conference on Computer Graphics*, Washington, DC, USA, 2001, pp. 80–.
  - 15) B. Beneš, V. Těšínský, J. Hornýš, and S. K. Bhatia, "Hydraulic erosion," *Computer Animation and Virtual Worlds*, vol. 17, no. 2, pp. 99–108, May 2006.
  - 16) P. Křištof, B. Beneš, J. Krivánek, and O. Štáva, "Hydraulic erosion using smoothed particle hydrodynamics," in *Computer Graphics Forum*, 2009, vol. 28, pp. 219–228.
  - 17) C. Wojtan, M. Carlson, P. J. Mucha, and G. Turk, "Animating Corrosion and Erosion," in *NPH*, 2007, pp. 15–22.
  - 18) X. Mei, P. Decaudin, and B.-G. Hu, "Fast Hydraulic Erosion Simulation and Visualization on GPU," in *15th Pacific Conference on Computer Graphics and Applications (PG'07)*, Maui, HI, USA, 2007, pp. 47–56.
  - 19) J. Vanek, B. Beneš, A. Herout, and O. Stava, "Large-Scale Physics-Based Terrain Editing Using Adaptive Tiles on the GPU," *IEEE Computer Graphics and Applications*, vol. 31, no. 6, pp. 35–44, Nov. 2011.
  - 20) T. Ito, T. Fujimoto, K. Muraoka, and N. Chiba, "Modeling rocky scenery taking into account joints," in *Computer Graphics International*, 2003. *Proceedings*, 2003, pp. 244–247.
  - 21) B. Beneš and X. Arriaga, "Table mountains by virtual erosion," in *Proceedings of the First Eurographics conference on Natural Phenomena*, 2005, pp. 033–039.
  - 22) M. D. Jones, M. Farley, J. Butler, and M. Beardall, "Directable weathering of concave rock using curvature estimation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 1, pp. 81–94, 2010.
  - 23) L. A. Tychonievich and M. D. Jones, "Delaunay deformable mesh for the weathering and erosion of 3D terrain," *The Visual Computer*, vol. 26, no. 12, pp. 1485–1495, Dec. 2010.
  - 24) K. Onoue and T. Nishita, "A method for modeling and rendering dunes with wind-ripples," in *Computer Graphics and Applications*, 2000. *Proceedings. The Eighth Pacific Conference on*, 2000, pp. 427–428.
  - 25) B. Beneš and T. Roa, "Simulating desert scenery," 2004.
  - 26) N. Wang and B.-G. Hu, "Real-Time Simulation of Aeolian Sand Movement and Sand Ripple Evolution: A Method Based on the Physics of Blown Sand," *Journal of Computer Science and Technology*, vol. 27, no. 1, pp. 135–146, Jan. 2012.
  - 27) S. Liu, Z. Wang, Z. Gong, L. Huang, and Q. Peng, "Physically based animation of sandstorm," *Computer Animation and Virtual Worlds*, vol. 18, no. 4–5, pp. 259–269, Sep. 2007.

- 28) J. J. Monaghan and A. Kocharyan, “SPH simulation of multi-phase flow,” *Computer Physics Communications*, vol. 87, no. 1, pp. 225–235, May 1995.
- 29) A. P. Tampubolon et al., “Multi-species simulation of porous sand and water mixtures,” *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–11, Jul. 2017.
- 30) J.-D. G enevaux,  . Galin, E. Gu erin, A. Peytavie, and B. Bene s, “Terrain generation using procedural models based on hydrology,” *ACM Transactions on Graphics*, vol. 32, no. 4, p. 1, Jul. 2013.
- 31) H. Tsoar, “Bagnold, R.A. 1941: The physics of blown sand and desert dunes. London: Methuen,” *Progress in Physical Geography - PROG PHYS GEOG*, vol. 18, pp. 91–96, Mar. 1994.
- 32) N. Wang and B.-G. Hu, “Aeolian Sand Movement and Interacting with Vegetation: A GPU Based Simulation and Visualization Method,” in *2009 Third International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications*, Beijing, China, 2009, pp. 401–408.
- 33) C. Wu, M. Wang, and L. Wang, “Large-eddy simulation of formation of three-dimensional aeolian sand ripples in a turbulent field,” *Sci. China Ser. G-Phys. Mech. Astron.*, vol. 51, no. 8, pp. 945–960, Aug. 2008.
- 34) F. M. White, *Viscous Fluid Flow*, 3 edition. New York, NY: McGraw-Hill Education, 2005.
- 35) “World Machine: The Premier 3D Terrain Generator.” <https://www.world-machine.com/>.
- 36) “Landscape Outdoor Terrain.” <https://docs.unrealengine.com/en-us/Engine/Landscape>.
- 37) “Rock Generator 2 2018 | ScriptSpot.” <http://www.scriptspot.com/3ds-max/scripts/rock-generator-2-2018>.
- 38) B.B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman, San Francisco, 1982.
- 39) M. Macklin and M. M uller, “Position based fluids,” *ACM Transactions on Graphics*, vol. 32, no. 4, p. 1, Jul. 2013.
- 40) HUMBOLDT STATE GEOSPATIAL ONLINE, “Atmospheric Absorption & Transmission.” <http://gsp.humboldt.edu/OLM/Courses/>

GSP\_216\_Online/lesson2-1/atmosphere.html.

CHEN ZUO



Master's Program in Intelligent Interaction Technologies, University of Tsukuba.