Aeolian Sand Movement and Interacting with Vegetation: A GPU Based Simulation and Visualization Method

Ning WANG, Bao-Gang HU
LIAMA/NLPR, Institute of Automation
Chinese Academy of Sciences
100190 Beijing, China
nwang@liama.ia.ac.cn, hubg@nlpr.ia.ac.cn

Abstract

Simulation and visualization on aeolian sand movement and its interaction with vegetation are a challenging subject. In this work, we proposed a physically and procedurally based modeling and simulation method that can be used to synthesize sandy terrain with vegetation covers. For realizing a real-time simulation process, we implemented the method on the programming graphics processing unit (GPU). We tried to include significant mechanisms of sand transportation into the sand model, such as saltation and creep behaviors, while simplified the vegetation model and wind field model to make the simulation feasible and affordable. As a preliminary study, we conducted several numerical demonstrations on simple scenarios with or without vegetation. The simulation results confirmed that the proposed method is applicable for reaching visually realistic images in comparison with the real images from a desert field.

1. Introduction

Sand covered landscape is widely distributed on earth, especially in the arid environment. Its geometrical generation, rendering and interactive applications have been attracted researchers of computer graphics. Early works were mainly focused on soil or mud, which were granular material similar to sand. One pioneer work was made by Li and Moshell [1]. They used analytic methods based on soil properties and Newtonian physics to simulate soil slippage and soil manipulations. They also gave a numerical solution on linear time thus their method can be used in real-time simulation. In Chanclou's work [2], loose soil was modeled as a plastic network (or control network) with topology structure, and then a refinement network was interacted to the control network in order to simulate the small scale phenomena.

Compared with soil, sand has an incompact structure since the adhesive forces between sand grains are weak. Height map is a common technique used to represent sand bed. Sumner et al [3] simulated ground material such as sand, mud and snow, deformed by rigid object models. Their simulation was a simple physical based method only with five independent parameters. They also demonstrated their algorithms by showing footprints made by a runner as well as bicycle tire tracks. A more absorbing work is Onoue and Nishita's virtual sandbox [4], in which the ground can be deformed by concave objects using height span map. There are also other forms to represent sand. In the work by Bell et al [5], the sand grain is represented by a large collection of spherical objects, and the forces on the sand grain was accumulated by the forces on every sphere. In this way, faithful behavior of sand could be simulated through physical model. Another unique work was made by Zhu and Bridson [6], where they regarded the sand as a continuum and simulated sand animation with fluid dynamics method.

When the amount of sand grain is huge, dunes are formed. In computer graphics domain, Onoue and Nishita [7] for the first time rendered the desert scene with a two scales model representing the dune and wind-ripples respectively, and they added wind-ripples to dunes with bump-mapping technique. Beneš and Roa [8] extended the work of Onoue and Nishita by a simulation of sand interacting with cactus, but their extension was a purely ad hoc approach other than a physically based one.

There is another kind of work that mainly focuses on small sand grains (or dust) in the air. Chen et al [9] used particle systems and computational fluid dynamics (CFD) to simulate the dust behavior generated by traveling vehicle. Their analysis on the dynamics of dust particle inspired us a lot. Another related work is the animation of sandstorm made by Liu et al [10].

978-0-7695-3988-1/10 $26.00 © 2010 IEEE
DOI 10.1109/PMA.2009.14

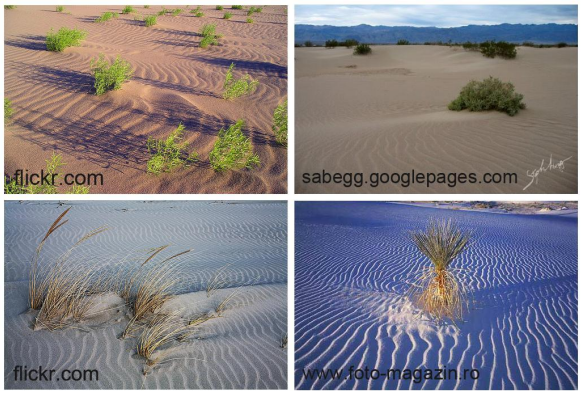


Figure 1. Sand ripple and vegetation [11].

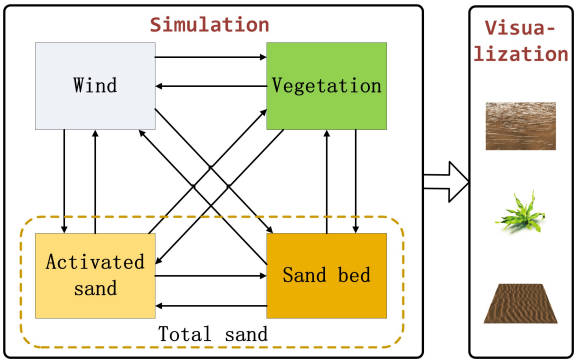


Figure 2. Schematic diagram of our model

They simulated sandstorm based on CFD and they implemented it on the GPU to accelerate the modeling process.

Sand originates from bedrock weathering, and the aeolian sand movement can be viewed as a kind of terrain deformation and erosion. The phenomena of weathering [12] and hydraulic erosion [13] [14] have been well simulated, but the aeolian erosion of sand bed is not. This is the main concern of this work. Since weathering and hydraulic erosion have very different mechanisms in nature compared with aeolian erosion, thus probing a new physically based method is in demand.

Figure 1 shows images of sand covered landscapes from different desert fields. Note that the sand ripples caused by aeolian sand transportation, and the shield and deposition effects behind the vegetation.

In this paper, we propose a physically and procedurally based simulation method for aeolian sand transportation and interacting with vegetation. Figure 2 is a schematic diagram of our proposed model, and a detailed description of submodels and relations can be seen in Table 1. Note that we do not consider the effect relating vegetation growth in the present model.

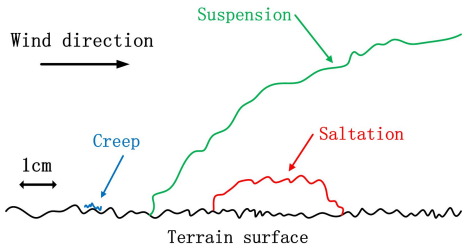


Figure 3. Three forms of aeolian sand grain movement, adopted and modified from [17].

Table 1. Interactions between submodels

Objects[1]	**Description of the Interaction**[2]
$W \Rightarrow V$	blowing and shaking the plant, altering the geometries of plant, *affecting its growth and developing etc.*
$V \Rightarrow W$	changing the wind profile, separating and dragging the wind flow.
$W \Rightarrow A$	entraining and accelerating the activated sand.
$A \Rightarrow W$	*dragging the wind flow.*
$W \Rightarrow B$	shearing the sand bed, activating the resting sand grain.
$B \Rightarrow W$	affecting the wind profile.
$B \Rightarrow A$	absorbing momentum when collided, changing the moving trajectory of the activated sand grain, mutual exchanging role with activated sand.
$A \Rightarrow B$	colliding with sand bed and shaping sand bed, mutual exchanging role with resting sand.
$B \Rightarrow V$	keeping the root system of vegetation, *supplying water and nutrition etc.*
$V \Rightarrow B$	keeping the sand, keeping the terrain.
$A \Rightarrow V$	colliding with vegetation, accumulating around vegetation.
$V \Rightarrow A$	obstructing sand grain, changing its flying trajectory both directly and indirectly.

[1] Wind (W), Vegetation (V), Activated sand (A), Sand bed (B). $W \Rightarrow V$ means W works on V.

[2] The simulated or partly simulated phenomenons in this paper are given in normal type, and the phenomenons neglected are given in italics.

Since the mechanics of two-phase flow (wind-sand in this paper) are too complicated for simulation, the dragging effect of sand to wind is neglected either. The rest of this paper is organized as follows: In section 2, the mechanics behaviors of sand grains are analyzed first, then we introduced two kinds of model used to represent vegetation of trunk shape and porous shape respectively. We also describe the simple wind field model we used at the last part of this section. In section 3 we implement our method on GPU and describe some techniques we used. Finally, simulations and demos are presented in section 4.

2. Model and Method

For better understanding this work, we will first introduce some basic concepts about the physics of blown sand. Interested readers may refer to [15] and

[16] for more details.

When the wind velocity reachs the threshold, sand grains on the sand surface will begin moving. The movement of sand grains can be basically sorted into three kinds: *creep, saltation and suspension* [15] [17], as shown in Figure 3. *Creep* is the larger sand grains (mainly between 0.25 and 2 mm) rolling and slipping on the sand surface. This kind of movement is mainly caused by the impact of saltating grains [15]. *Saltation* looks like a jumping motion that grains are taken up into the airstream and carried forward a few centimeters by wind. Generally, the average jumping height is about 10 cm [15]. When the saltation grain falls off it can rebound or splash other grains on the sand surface. Saltation mainly contains grain between 0.1 and 0.25 mm and it is the main form of sand grain transportation. According to Bagnold's view [15], more than 70 percent sand grain movement is saltation, and Mabbutt commented that saltation normally accounts for about 95 percent of the bulk transport of sand [16]. In the paper of Lyles [18] working on soil erosion, this number is 50-80 percent. Thus in the following text, we mainly focus on saltation movement. Some smaller fine grains, usually under 0.05 mm, once are carried into airstream will hard to deposit to the ground. This kind of long distance aeolian sand transportation is called *suspension*. Suspension is neglected in our model because it only takes little amount of total sand movement, and its long distance transportation and uniformly deposition don't influence local terrain shape.

2.1. Mechanical Behavior of Single Grain

For simplicity, in our simulation we assumed that the sand surface is composed of spherical grains with an uniform diameter d, and we supposed $d = 0.15mm$ since grain of this scale mainly transport in saltation mode [16].

2.1.1. Grain Entrainment. When wind blows across the sandy surface, part of its horizontal momentum is transferred into a kind of vertical force exerting on the grains of the surface. This force is called shear stress and its intensity is determined by the velocity gradient on the sand bed. Due to the difficulty of ascertaining the distribution of shear stress on a vegetation covered surface, we simply use wind velocity instead of shear stress. We found this approximation works well visually despite it is crude. Hereby we supposed that sand grains have a start-up velocity greater than zero when friction velocity u_* exceeds the wind threshold friction velocity u_{*t}, and the start-up velocity is proportional to the wind velocity on the reference height of sand bed.

2.1.2. Transportation in the air. In the air, the forces acting on a single saltation grain are the drag of air, the Saffman force, the Magnus force, the Basset force, the electrostatic force, the virtual mass force and the gravity force, but among them the most important forces are the gravity $\mathbf{F_g}$ and drag force $\mathbf{F_d}$ [19]. Hereby in our simulation, we only consider the gravity of the grain and the drag force caused by wind. In Cartesian coordinate, $\mathbf{F_g}$ and $\mathbf{F_d}$ can be computed as [19]:

$$\mathbf{F_g} = -\frac{1}{6}\pi d^3 \rho_s g = -mg \tag{1a}$$

$$\mathbf{F_d} = \frac{1}{8}\pi d^2 C_{ds} \rho_a \|\mathbf{\Delta V}\| \mathbf{\Delta V} = B\|\mathbf{\Delta V}\|\mathbf{\Delta V} \tag{1b}$$

where d is grain diameter and m is grain mass. C_{ds} is the drag coefficient of sand grain. ρ_a and ρ_s are air density and sand density. $\mathbf{\Delta V} = \mathbf{U} - \mathbf{V}$ is the relative velocity between wind and flying sand. The direction of $\mathbf{F_d}$ is the same with $\mathbf{\Delta V}$. B can be viewed as a constant in our simulation. A simpler formula of $\mathbf{F_d}$ is depicted by Herrmann [20]:

$$\mathbf{F_d} = \frac{1}{\sigma}\mathbf{\Delta V} \tag{2}$$

where σ is the characteristic response time of the moving sand to changes of the external wind velocity. Based on these hypothesis, the trajectory of the grain can be iteratively computed by the Newton's second law:

$$\mathbf{F_d} + \mathbf{F_g} = m\frac{\mathbf{V_{curr}} - \mathbf{V_{prev}}}{\Delta t} \tag{3a}$$

$$\mathbf{P_{curr}} = \mathbf{P_{prev}} + \mathbf{V_{curr}}\Delta t \tag{3b}$$

2.1.3. Deposition. When a saltation grain falls to the sand bed, if it has enough energy it can still rebounds or splashes other grains. We suppose there is only one grain can be splashed by the incident grain, and the splashed grain will get all the rebounding momentum from the collision system of two grains. So if the incident grain has collided and splashed another grain, it will becomes rest after the collision. This hypothesis is consistent with Bagnold's photography observation [15]. The collision detection technique used here is similar to the work of Kolb [21]. The incident particle velocity is decomposed into a normal component and a tangential component. Two damping parameters are used to simulate the friction force on the tangential direction and the momentum attenuation on the normal direction respectively. If the normal velocity or tangential velocity is smaller than a given threshold, it

should be set to zero to avoid velocity from slowing down close to zero.

2.1.4. Collapsing. If sand grains accumulate too much at one place, they will avalanche by their own weights. The steepest stable angle of a free sand surface is called its angle of repose, denote by symbol θ. The avalanche of sand can be seen as a kind of creep, but the cause for creep is not the wind but gravity. Since our simulation is on a two dimensional grid (or height map), this creep movement can be done parallely on GPU through selective filtering. If the amount of the sand on the current grid exceeds the angle of repose, we can set part of the exceeding grain flux to the neighborhood. This process can be formulated as:

$$\begin{aligned} h_{curr}(i,j) &= h_{prev}(i,j) \\ &\quad +G\Delta h\widetilde{step}(\tan\theta, T\Delta h) \qquad (4) \\ \Delta h &= \frac{2-\sqrt{2}}{4}\sum_{N_4} h_{prev}(i,j) \\ &\quad +\frac{\sqrt{2}-1}{4}\sum_{N_8-N_4} h_{prev}(i,j) \\ &\quad -h_{prev}(i,j) \qquad (5) \\ \widetilde{step}(a,b) &= \begin{cases} 1 & \text{when } b \geq a \\ 0 & \text{when } b < a \end{cases} \qquad (6) \end{aligned}$$

where $h_{curr}(i,j)$ and $h_{prev}(i,j)$ are the heights of grid (i,j) on current frame and previous frame respectively. N_4 and N_8 represent the four neighborhoods and eight neighborhoods of the current grid respectively. The weight $(2-\sqrt{2})/4$ and $(\sqrt{2}-1)/4$ are the reciprocal of distance between current grid and neighbor grid. G is a parameter related to gravity, and T is a grid scale factor. Scenes of sand grain movement are shown in Figure 4 and a detailed description can be seen in section 4.

2.2. Vegetation Model

Vegetation in nature always has a complex internal and external geometries, so it is not easy to get the exactly aerodynamics and circumjacent flow information such as drag coefficient and velocity distribution. Experimental measurement is a direct way, and sometimes for simply modeling, a value is normally assigned based on solid element of similar shape, for example cylinders or hemispheres. The complex geometry of vegetation is also intractable when relates to sand grains. To simplify the geometric characteristic of the vegetation, we consider the vegetation as a cylinder shaped object, as the method in [22]. This modeling method can greatly make things easy to go in the case of presence of vegetation.

2.2.1. Influences on wind velocity. Vegetation covers can drag the wind passed and reduce the wind's speed. If they flourish enough, they can also change the wind profile on the surface. A detailed description about wind field model will be given later. Here we first discuss the drag coefficient of a single plant.

There are some experiments to get the drag coefficient of a single plant, and the results are various. For example, Guan et al [23] got $C_{dv} = 1.08(1-\beta^{1.8})$ through wind-tunnel study, where β is the optical porosity of the plant. Gillies et al [24] measured three species of plants and the average C_{dv} are 0.42, 0.39, 0.34 respectively. Some annual plants were measured by Hagen [25] and the results are from 0.85 to 1.21. Based on these experiences we roughly assign 0.7 as the drag coefficient with optical porosity 0.5.

2.2.2. Influences on sand. Sand movement is obstructed by the vegetation, especially the vegetation has a trunk-like geometry. Sands are collide with the trunk and rebounded to ground, swept by the wind and finally formed a sand shield before the trunk. The sand shield can be simulated with sand collision and collapsing method mentioned above, whereas a boundary condition must be added to avoid the sand creep into the trunk. We will demonstrate windward accumulation before trunk in Figure 4(c) and find it visually realistic.

The method above is effective to the trunk-like plant, such as bole and cactus. When sands encounter porous plant, for example grasses and shrubs, they will not only deposit at the windward, but also at the leeward and in the domain of plant. Since sands deposit near uniformly in the plant domain, the visual effect is not conspicuous especially from the view of bird's eye. Moreover, considering the cost of collision detection on GPU, we can simply ignore the collision between sand and porous grass in most situation.

2.3. Wind Field Model

Ideal flows and turbulence are themselves complex, and this complexity is significantly increased for boundary-layer flows and turbulence around the vegetation [26]. This is because the terrain surface is not an ideal homogeneous flat surface, and the geometries and structures of vegetation are always complicated. Computational fluid dynamics is a numerical method used to analyze and solve problems involving fluid flows. This kind of method is very time-consuming because it needs solving Navier-Stokes equations on grid iteratively. To get a real-time 3D simulation and visualization, CFD based fluid simulation methods are

unaffordable in our application. Here a simple wind field model is adopted and we find it satisfies our need. In the following part, we supposed that the direction of wind is identical during the simulation.

According to the boundary-layer theory, the well-known logarithmic wind profile on a neutrally-stratified rough surface can be formulized as [22] [27]:

$$U(z) = \frac{u_*}{k} \ln(\frac{z - z_d}{z_0}) + \Phi \tag{7}$$

where z is the height above the substrate surface, and $U(z)$ is the mean velocity at the height z. $k = 0.4$ is the dimensionless Von Karman's constant. Φ is the thermal stability term, which can be set to zero in our application (adiabatic condition). z_d is zero-plane displacement height, it is an empirical parameter related to the elements on the surface such as vegetation. z_0 is aerodynamic roughness length, it means the height at which wind speed reaches zero. Considering the physical and mathematic reasons described in [22], and our motive is only visually correct, we further ignore z_d and set it to zero, thus z_0 is the only one to estimate.

A quantity often used for characterizing vegetation covered surface is frontal area index:

$$\lambda = \frac{b_v h_v}{D^2} \tag{8}$$

where h_v and b_v are the height and breadth of the plant respectively. D is the spacing between plants. If the plant on the sand surface is sparse enough, the influence of vegetation to the wind profile can be neglected. But when plants are uniformly or continuously distributed, their effects to z_0 must be taken into account. There are many ways to estimate z_0. A detailed list can be seen in [27] and [28]. We prefer to the expressions of Lettau [29] since it is reasonable when $\lambda < 0.15$ [28] and given in a simple form. It is defined as:

$$z_0 = 0.5 h \lambda \tag{9}$$

For a bare sandy surface, Bagnold suggest that $z_0 = d/30$ [15].

In our simulation, we also need to know the velocity distribution around vegetation to compute the drag force of wind to sand grain, and here we use the expression of wake flow. The velocity distribution of a wake behind a circular cylinder can be denoted as $U = U_\infty - \bar{U'}$ [30], where:

$$\bar{U'} = U_\infty \frac{\sqrt{10}}{18\gamma} \sqrt{\frac{C_D d}{x}} \left\{ 1 - \sqrt[3]{\frac{y}{b_w}} \right\}^2 \tag{10}$$

and U_∞ is a reference wind speed. In our simulation, it can be set to $U(z)$. x and y are coordinates with origin at the center of plant. $b_w = W(C_D dx)^{1/2}$ is the half-width of wake, γ and W are factors, and C_D is drag coefficient. Here we simply let $C_D = C_{dv}$. To get a more precise model, separation flow can be further considered, but we find its effect to bed shape isn't as obvious as the wake flow.

3. Implementation on GPU

Our application is implemented with OpenGL graphics API and Cg shading language. The sand grain is simulated with particle systems, and the sand surface is represented by a height map. We have followed the Kolb's GPU data structure for particle system [21]. The particle's position and velocity are packed on a RGBA texture respectively, and each texture is bound with a PBuffer object. Since we cannot read and write one buffer at the same time in a fragment program, double buffering technique (or ping-pong buffering) is used here. The height buffer which bound with height map is also done in the same way.

In our simulation, all particles are generated at the surface of the sand bed when entrained by the wind, and are died when falling to the bed and ultimately no more motion. The particle's velocity and trajectory can be computed iteratively on fragment program according to the description in section 2.1. The height map is updated according to the particle's accumulation. When a particle is allocated, the height map will decreased at the position of its birth. In a similar way, when a particle is died, the height map will increased at the position of its death. Hence the height map need updated two times for one particle. To avoid exchanging the double height buffer frequently, the increased or decreased height can be added to the height map using alpha blending.

During rendering, the position texture is packed into a vertex buffer object and render to the frame buffer directly. The height map is read in vertex program with vertex texture fetch and using vertex texture displacement [31] for sand bed rendering. We use billboard technique to render the vegetation on the surface since it has little burden to performance but visually realistic. Finally, per-pixel Phong lighting model is used in our terrain rendering.

4. Simulation and Visualization Results

We have applied our method to several terrain simulations on a grid of 256×256 (Figure 4 - 8). In our simulation, the activated sand grains are generated uniformly on the surface, and it is reasonable to suppose that the amount of activated sand grains is different according to wind conditions, since we have the experiences that strong wind usually brings more

Table 2. Parameters used in demos

	T_s[1]	u_*[2]	z_0[3]	N_a[4]	N_t[5]
Demo 1	400	0.2	0.0005	65536	4420282
Demo 2	400	0.3	0.0005	202500	12298585
Demo 3	400	0.2	0.0005	65536	4744080
Demo 4	400	0.2	0.4	65536	4419445

[1] Simulation time ($sec.$).
[2] Friction velocity (m/s).
[3] Aerodynamic roughness length (cm).
[4] The amount of activated particles per frame.
[5] Total particles generated in simulation.

sand grains transported with it. Here we use Bagnold's expression $q \propto Cu_*^3$ [15], where q is the total sand movement and C is a constant parameter. In all the simulations, the wind always blows from left to right, and in order to conserve the mass, sand flow streaming out the boundary will reenter from the other side. The simulation frame rate is different according to the particle number. For a simulation with 200 thousand particles, the frame rate is exceed 60 fps (NVIDIA 8800GTX, Intel Core 2 6600 2.4GHz), and a satisfiable result can be got within 200 seconds. Other parameters used in demo are list in Table 2.

Figure 4(a) is a snapshot from an angle of view nearing the sand bed. The particles are visualized into a series white lines since they have a high speed in the flow. The length of the line represent their speed. The short and tip-tilted ones are the particles just saltating to the air, either sheared by the flow or rebounding by itself. The longer and descending ones are the particles falling to the sand bed. They have a much faster speed since they stay longer in the air thus accelerated adequately. A holistic view is in Figure 4(b).

The windward accumulation is demonstrated through Figure 4(c). Note at the nearest two grasses and the bole of a tree. Grass on the left has a optical porosity equals 0.5, and for comparison, grass on the right is made no accumulation. Moreover, we can see the shape of windward accumulation is different according to plant species. Since windward accumulation isn't obvious from the view of bird's eye, it can be neglected in Demo 3 and Demo 4.

Figure 5 shows a sand ripples simulation with 65536 activated particles. We can see the developing process of sand ripples in which small ripples growing and incorporating into big ripples. Note that at the beginning the ripples are small, ruptured and have a crescent shape (Figure 5(b)). As time growth, these small curved ripples are graduated into straight structures and interdependently unite into line shape. This is in agreement with Bagnold's field observation and stability analysis [15]. Moreover, our sand ripples are non-symmetric ripples with branching structure, which are more realistic than symmetric ripples. Since particles are generated on the sand bed randomly, the simulation results are different every time, but they have similar shape if the parameters are the same. From the figure we can see our simulation method is reasonably good for representing sand ripple evolvement.

Bagnold also observed that the wavelength of sand ripple is increased with the wind speed [15]. This is demonstrated in Figure 6. All the conditions are similar to demo 1 except the wind is stronger. The wavelength of sand ripples in Figure 6 are longer compared with Figure 5, since the saltation grain has a longer jumping step in stronger wind.

The effect of vegetation covers is considered in Figure 7 and Figure 8. Grasses are packed into small tussock and laid on the sand bed. The drag coefficient of the tussock is 0.7 which has been depicted in section 2.2, and we simply use this drag coefficient to compute the flow around the vegetation as a trade-off. We can see at the leeward, sand shadow is formed as a result of the shield effect of the tussock. If the simulation time is longer enough, the sand shadow will tapers downwind, which is a characteristic in Figure 1. The sand activity in Figure 8 is further weakened not only because the shield effect but also due to the roughness length of the bed is changed, thus the aeolian sand activity by wind is strongly attenuated.

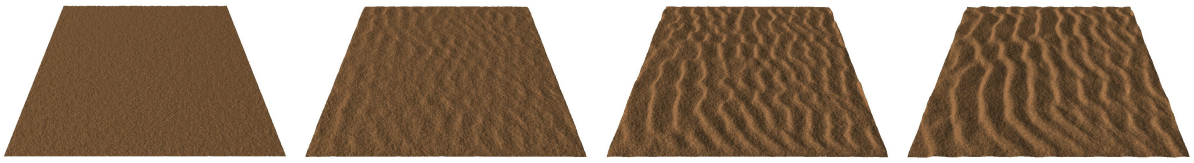
5. Final Remarks

The main contribution of this work is proposed a framework to simulate the aeolian sand movement on GPU, including the situation of existing vegetation covers. Since our sand movement simulation is physically based, it is easily coupled with other models, for example vegetation model in this paper. In our work, one difficult part is the accurate expression of wind field, especially in the presence of vegetation. Another part is the collision detection between vegetation and sand grain. Both of them are caused by a fact that the vegetation structure is too complicated. We did some hypothesis and simplifications to overcome the uncertain and unaffordable elements in model, for example using cylinder model to represent vegetation.

Using computational fluid dynamics to compute the wind field may be a further improvement of this work. With CFD method, we can simulate flow-vegetation interaction through boundary conditions. Moreover, plant growth model such as GreenLab [32] [33] can be incorporated with our model to enhance the significance and application of our model. Advancements in computational fluid dynamics, aerodynamics, aeolian sand physics, ecology and computer graphics may

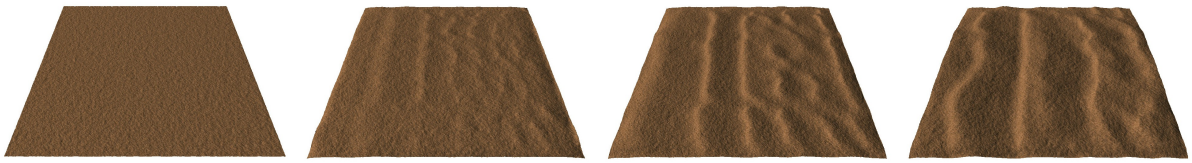
(a) A close view of aeolian sand flow (b) An overlook of aeolian sand flow (c) Comparison of windward accumulations.

Figure 4. Aeolian sand flow and windward accumulation



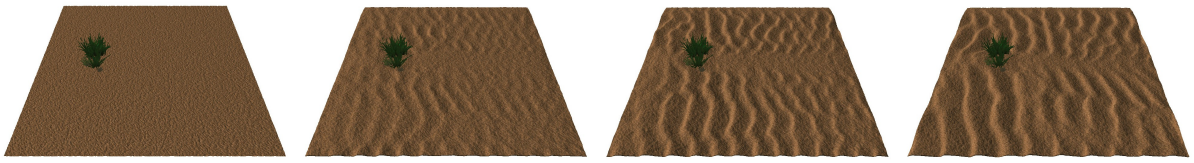
(a) T_s: 0 sec. (b) T_s: 100 sec. (c) T_s: 200 sec. (d) T_s: 400 sec.

Figure 5. Demo 1: Sand ripples in wind



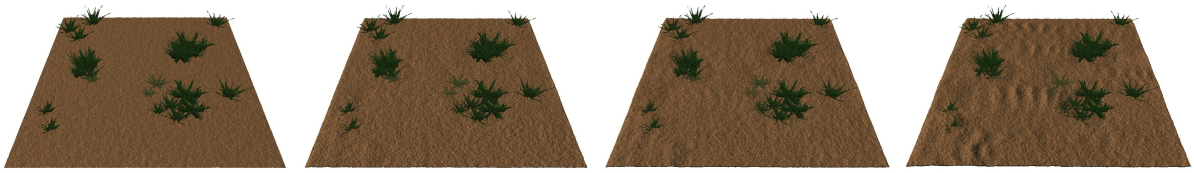
(a) T_s: 0 sec. (b) T_s: 100 sec. (c) T_s: 200 sec. (d) T_s: 400 sec.

Figure 6. Demo 2: Sand ripples in stronger wind



(a) T_s: 0 sec. (b) T_s: 100 sec. (c) T_s: 200 sec. (d) T_s: 400 sec.

Figure 7. Demo 3: Sand shadow in the leeward



(a) T_s: 0 sec. (b) T_s: 100 sec. (c) T_s: 200 sec. (d) T_s: 400 sec.

Figure 8. Demo 4: Aeolian sand activity attenuated by vegetation cover

boost our work. During the process of this work, we cannot help to feel that there are so many unknown and uncertain matters in nature and we still have a long way to go.

Acknowledgment

We would like to thank Dr. Xing Mei for helpful advises on GPU programming. We also thank the anonymous reviewers for their valuable comments. This work is supported in part by China National 863 Program (No. 2006AA01Z301) and MOST of China (No. 2007DFC10740).

References

[1] X. Li and J. M. Moshell, "Modeling soil: Realtime dynamic models for soil slippage and manipulation," in *Proceeding of SIGGRAPH 93*, 1993, pp. 361–368.

[2] B. Chanclou, A. Luciani, and A. H. Acroe, "Physical models of loose soils dynamically marked by a moving object," in *Computer Animation '96*, 1996, pp. 27–35.

[3] J. F. O. Robert Sumner and J. K. Hodgins, "Animating sand, mud, and snow," *Computer Graphics Forum*, vol. 18, no. 1, pp. 17–26, March 1999.

[4] K. Onoue and T. Nishita, "Virtual sandbox," in *Proceeding of Pacific Graphics'03*, 2003, pp. 252–259.

[5] N. Bell, Y. Yu, and P. J. Mucha, "Particle-based simulation of granular materials," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2005, pp. 77–86.

[6] Y. Zhu and R. Bridson, "Animating sand as a fluid," in *SIGGRAPH'05*, 2005, pp. 965–972.

[7] K. Onoue and T. Nishita, "A method for modeling and rendering dunes with wind-ripples," in *Proceeding of Pacific Graphics'00*, 2000, pp. 427–428.

[8] B. Beneš and T. Roa, "Simulating desert scenery," in *Proceeding of WSCG'04*, 2004, pp. 17–22.

[9] J. X. Chen, X. Fu, and E. J. Wegman, "Real-time simulation of dust behavior generated by a fast traveling vehicle," *ACM Transactions on Modeling and Computer Simulation*, vol. 9, no. 2, 1999.

[10] S. Liu, Z. Wang, Z. Gong, L. Huang, and Q. Peng, "Physically based animation of sandstorm," *Computer Animation and Virtual Worlds*, vol. 18, pp. 259–269, 2007.

[11] http://flickr.com, http://sabegg.googlepages.com, http://www.foto-magazin.ro.

[12] J. Dorsey, A. Edelman, H. W. Jensen, J. Legakis, and H. K. Pedersen, "Modeling and rendering of weathered stone," in *SIGGRAPH '99*, 1999, pp. 225–234.

[13] 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*, 2007.

[14] B. Neidhold, M. Wacker, and O. Deussen, "Interactive physically based fluid and erosion simulation," in *Eurographics Workshop on Natural Phenomena*, 2005, pp. 25–32.

[15] R. A. Bagnold, *The Physics of Blown Sand and Desert Dunes*. Methuen Co. Ltd, 1941.

[16] J. A. Mabbutt, *Desert Landforms*. The MIT Press, 1977.

[17] J. Ni and Z. Li, *2-Phase Flow of Blown Sand, Theory and Application (in Chinese)*. Chinese Science Press, 2006.

[18] L. Lyles, "Wind erosion: Processes and effect on soil productivity," *Transactions of the ASAE*, vol. 20, no. 5, pp. 880–884, 1977.

[19] C. Wu, M. Wang, and L. Wang, "Large-eddy simulation of formation of three-dimensional aeolian sand ripples in a turbulent field," *Science in China Series G: Physics, Mechanics and Astronomy*, pp. 945–960.

[20] H. J. Herrmann and G. Sauermann, "The shape of dunes," *Physica A: Statistical Mechanics and its Applications*, vol. 283, pp. 24–30, 2000.

[21] A. Kolb, L. Latta, and C. Rezk-Salama, "Hardware-based simulation and collision detection for large particle systems," in *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, 2004, pp. 123–131.

[22] Z. Dong, S. Gao, and D. W. Fryrear, "Drag coefficients, roughness length and zero-plane displacement height as disturbed by artificial standing vegetation," *Journal of Arid Environments*, vol. 49, pp. 485–505, 2001.

[23] D. Guan, Y. Zhang, and T. Zhu, "A wind-tunnel study of windbreak drag," *Agricultural and Forest Meteorology*, vol. 118, pp. 75–84, 2003.

[24] J. A. Gillies, W. G. Nickling, and J. King, "Drag coefficient and plant form response to wind speed in three plant species: Burning bush, colorado blue spruce and fountain grass," *Journal of Geophysical Research*, vol. 107, pp. 102–115, 2002.

[25] L. J. Hagen and E. L. Skidmore, "Windbreak drag as influenced by porosity," *Transactions of the ASAE*, vol. 14, no. 8, pp. 464–465, 1971.

[26] H. WANG and E. S. Takle, "Boundary-layer flow and turbulence near porous obstacles," *Boundary-Layer Meteorology*, vol. 74, pp. 73–88, 1995.

[27] L. Lyles and B. E. Allison, "Wind profile parameters and turbulence intensity over several roughness element geometries," *Transactions of the ASAE*, vol. 22, no. 2, pp. 334–343, 1979.

[28] Y. Shao and Y. Yang, "A scheme for drag partition over rough surfaces," *Atmospheric Environment*, vol. 39, pp. 7351–7361, 2005.

[29] H. Lettau, "Note on aerodynamic roughness parameter estimation on the basis of roughness element description," *Journal of Applied Meteorology*, vol. 8, pp. 828–832, 1969.

[30] P. K. Chang, *Separation of Flow*. Pergamon Press, 1970.

[31] Y. Kryachko, "Using vertex texture displacement for realistic water rendering," *GPU Gems 2*, pp. 283–294, 2004.

[32] P. de Reffye and B.-G.Hu, "Relevant qualitative and quantitative choices for building an efficient dynamic plant growth model: GreenLab case," in *Proceedings of PMA03*, 2003, pp. 87–107.

[33] B.-G. Hu, P. de Reffye, X. Zhao, H.-P. Yan, and M.-Z. Kang, "GreenLab: A new methodology towards plant functional-structural model, structural aspect," in *Proceedings of PMA03*, 2003, pp. 21–35.