# Wind Erosion Simulation on Meshes

Student: Hsin-Hsiao Lin

Advisor: Dr. Jung-Hong Chuang

Dr. Wen-Chieh Lin

Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

## ABSTRACT

In real world, most objects are imperfect and these imperfections are caused by weathering. Erosion phenomena have been discussed in many literatures whereas wind erosion phenomena is rarely studied. We present a scheme for wind erosion on meshes. We develop our simplified wind erosion simulation equation and use this equation in our simulation framework. In contrast to the previous works in simulating desert scene, wind direction in our method is in 3D space and we can control parameters in our equation to simulate the erosion effect under different soil material conditions.

# Acknowledgments

I would like to thank my advisors, Professor Jung-Hong Chuang, and Wen-Chieh Lin for their guidance, inspirations, and encouragement. I also want to thank Professor Sai-Keung Wong for his encouragement and many many advices made this paper more completely. I am grateful to Tan-Chi Ho for his comments and advices. Thanks to my colleagues during these two years: Yueh-Tse Chen, Chin-Hsian Chang, Kuang-Wei Fu, Ying-Tsung Li, Shang-Wen Wang, Kuei-Li Fang, Yi-Jheng Huang, Cheng-Min Liu, and Hong-Xuan Ji and Ta-En Chen and Jau-An Yang and Tsung-Shian Huang for their assistances and discussions. Specially thanks to Yi-Jheng Huang, who helped me solve a lot of the issues in using 3DS Max. I also want want to thank senior colleagues, Chih-Wen Chang, Yi-Chun Lin, Yu-Shuo Lio. It is pleasure to be with all of you in the pass two years. Lastly, I would like to thank my parents for their love and support.
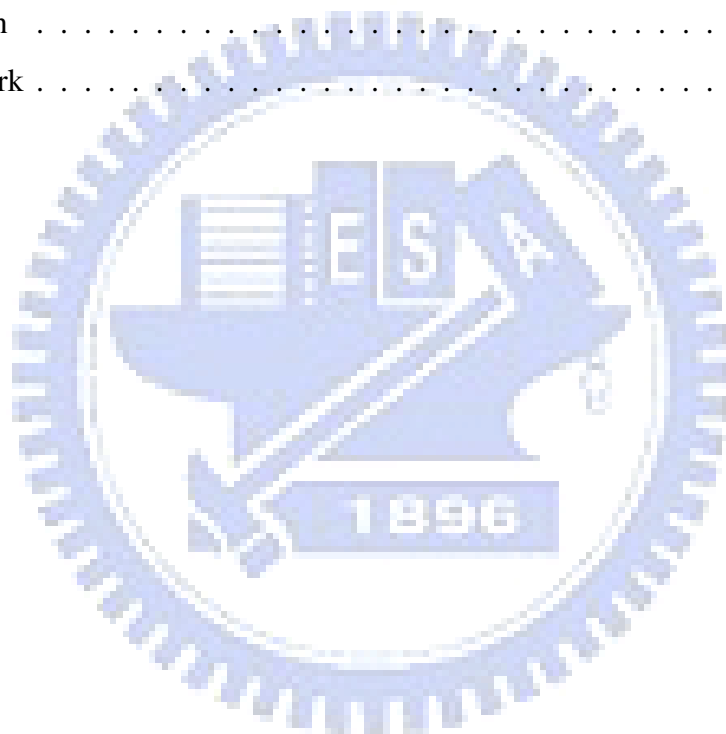
# Contents

# List of Figures

# Introduction

In real world, most objects are imperfect. Even a brand new product, there are still some flows or blemishes on it. These imperfections are caused by weathering, which refer to by people, animals or by the change of the intrinsic in physical way or chemical way. But in the virtual world, everything seems to be too clean and too smooth. In order to make virtual scenes realistic, artists have to do labor intensive works in texturing or modeling. But these works have to be done carefully to account for objects usage conditions. Researcher in computer graphics use weathering simulation to help this problem. But the great variety and complexity of involved scientific in every weathering phenomena makes this problem still challenging.

## 1.1 Thesis Overview

Many weathering phenomenas have been studied. Erosion phenomena changes the appearance of objects and smooths object sharp corners. Terrain erosion is a specific form of erosion process. Many works have been proposed in this field, taking into account specific terrain weathering processes[25][5]. These methods often represent terrain as height field, hence restrict the

structure of object. In contrast to height field representation, mesh can represent most structure of objects. Classified by external forces, there are several erosion processes: hydraulic erosion denotes the erosion caused by water and rainfall, thermal erosion denotes the erosion caused by the temperature changes in the day and in the night, and wind erosion denotes the erosion caused by wind force. Hydraulic erosion[3][2][23] and thermal erosion[6][7][4] have been discussed in many literatures, whereas wind erosion phenomena is rarely studied.

In this paper, we developed a new scheme to simulate the wind erosion phenomena on meshes. We based on an empirical equation called 'Wind Erosion eQuation', which have been used in soil science research for decades, developing our simplified wind erosion simulation equation. And then we introduce this equation to our simulation framework.

### 1.1.1   Contribution

Recently, a survey of aging and weathering phenomena in computer graphics[24] has mentioned several challenging tasks in the future. They are:

1 Introduction of new aging processes

2 Validation of the methods

3 Interactions with other fields and extensions to aging simulation

4 Predictability and control of results

5 Scaling factor and level of details

Our scheme fulfills the tasks [1] by choosing photo-realism and [4] by our redesigned wind erosion simulation equation.

Briefly, the contributions of this thesis can be summarized as :

• We redesign wind erosion equation which can be used in mesh representation

• We extend the wind erosion phenomena simulation from height field to 3D mesh objects

- We design a new framework to simulation mechanical based weathering effects, which mainly changes geometry appearance of objects

- Our simulation framework can be easily use graphics hardware to accelerate the simulation computation and shorten the simulation time

- Our simulation framework can be easily adapt the gpu based inverse displacement mapping technique to accelerate the rendering computation

## 1.1.2 Outline

The rest of the thesis is organized as follows : Chapter 2 gives the classify of natural phenomena simulation, literature review and previous works in wind erosion simulation. Chapter 3 presents the our simulation framework as well as our wind erosion simulation equation. Chapter 4 shows our experimental result from our approach. In the end, conclusions and future work are discussed in Chapter 5.

<div align="right">
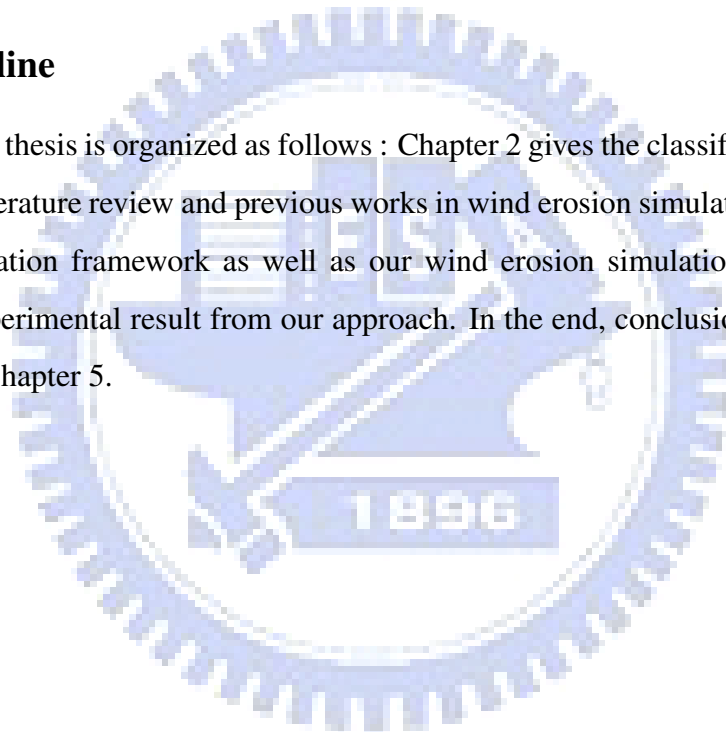
**CHAPTER 2**

</div>

---

# Background

---

This chapter introduce the related work of natural phenomena simulation. In the later sections, we will introduce a taxonomy of natural phenomena simulation at beginning. And then we introduce two previous works on wind erosion simulation. In the final section in this chapter, we discusses the background knowledge of mesh representation.

## 2.1   Natural Phenomena Taxonomy

The basic definition of weathering phenomena is simple. It consists in every process that makes objects decay. It is not possible to only consider changes in appearance over the time. These damages can be influenced by:

- External factors: forces from the environment condition, such as atmospheric conditions, usage, mechanical damages or erosion.

- Internal factors: intrinsic to materials themselves, specifically due to their manufacturing processes

4

We follow the classification proposed by Lu et al.[21] here. A reference surface, considered as perfectly clean, with a perfectly smooth geometry can be affected by a variety of different weathering processes composed of various attacks. These attacks can be classified by the type of process:

- Chemical processes: these attacks can occur both in the bulk material and onto its surface. They can also introduce specific structural damages, ex: destructive corrosion.

- Mechanical processes: geometry of original surface is affected by external factors and matter is removed. This can occur at every geometric scale. On a large scale, visible parts of the affected object are removed. For example, impacts at a geometric scale on a surface. At a smaller scale, fine details of the affected object are removed, such as scratches on a surface.

- Biological processes: these processes can result from external organic materials attacks, for example the growth of a plant, as well as to internal biological changing of materials themselves. In this latter case, it is hard to distinguish biological processes from chemical ones. Thus, here we classify each process that deals with organic materials as 'biological', and the others as 'chemical'.

## 2.2   Wind Erosion Simulation

In this section we introduce the wind erosion phenomena and then we discuses the previous works in desert scene simulation which are the most related work to wind erosion simulation.

### 2.2.1   Wind Erosion Phenomena

Wind erosion phenomena is classified mechanical process in the weathering process. The main effect of this weathering phenomena is the disappearing fine details of geometry on object surface, and make the hard rock into soft sands gradually. The wind erosion phenomena is composed of three elementary processes: saltation, creep and suspension. Fig.2.1 illustrate this tree kinds of process[1].

#### 2.2.1.1   Saltation

In saltation, the particles advance forward through a series of jumps. The wind lifts the first particle into the air to a height of around a meter or two. Those particles too heavy to remain suspended, drift downwind a distance of approximately four times the height they attained above the ground. When the particles return to earth, many will hit another particle, causing it to jump up and forward.

#### 2.2.1.2   Creep

When saltating particles strike particles too heavy to them, they may only nudge the larger grains along, a slow sliding and rolling movement known as creep. Creep usually requires stronger wind energy to the saltating particle to cause it to move the larger one.

#### 2.2.1.3   Suspension

The smallest dust particles are carried through the air by suspension. Suspension occurs when the dust is lofted into the air and held there by upward air currents strong enough to support the

weight of the particles. Typical surface wind speeds can suspend dust particles with diameters less than 0.2 millimeters and carry them short distances. Severe windstorms, however, can not only hold large particles aloft for some time, but may also push them to very high altitudes, which enhances their travel distance. Under strong winds, suspended dust particles may be lifted thousands of meters upward and drift thousands of kilometers downwind.
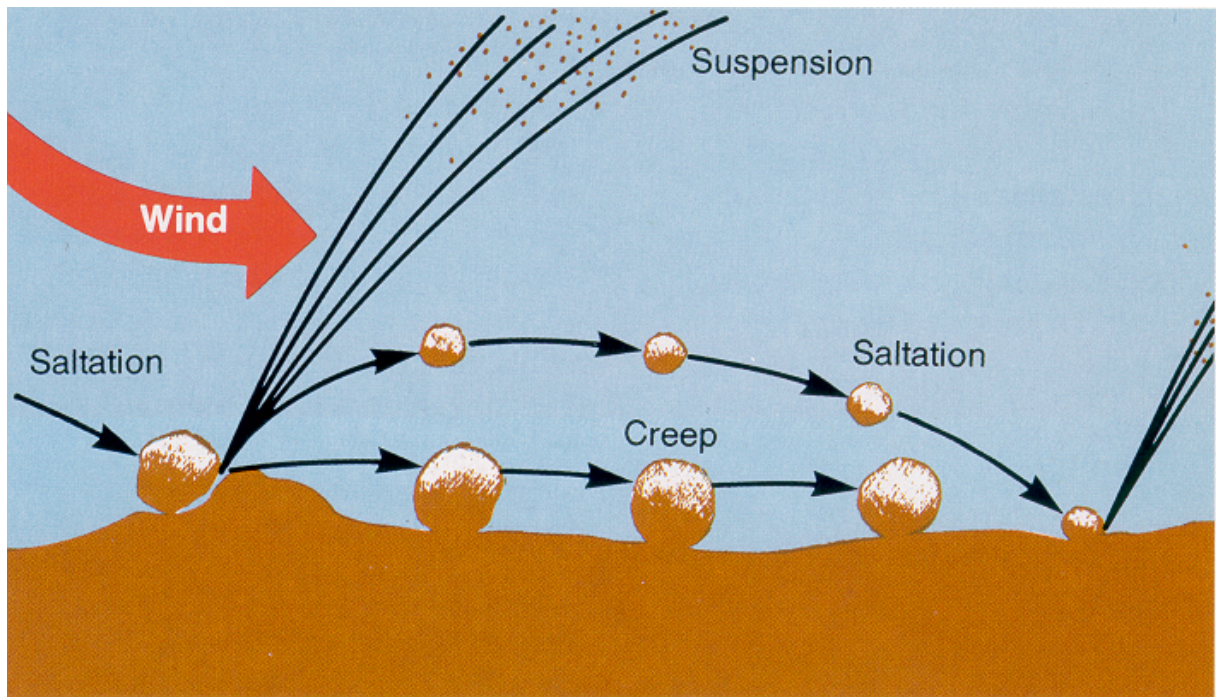


Figure 2.1: Sand transferred by wind force[1].

## 2.2.2  Desert scene simulation

Desert scene is one of the common natural landscape in the world. But the modeling and rendering desert scenes have not been studied until Koichi Onoue and Tomoyuki Nishita proposed their simulation method[27]. They proposed a simple model, which originally proposed by Nishimori et al[26], for formation dynamics of creep and saltation of sand. After calculating the height transformation in their test desert scene, they compute the corresponding bump map and use the bump map for rendering. In their modeling formation, they restrict the wind di-

rection to position x, and use two control parameters to control the transfer soil quantity hence form the wind-ripple and dune appearance. Although their method can create desert scene, their result is monotonic.

Later in 2004, Bedrich Benes and Toney Roa extend the algorithm of Onoue and Nishita by a simulation of sand interacting with objects[8]. Bedrich Benes and Toney Roa relief the wind direction into XY plane of their test desert scene and add obstacles in their test scene. The interaction with obstacles have two fundamental effects. The first is the accumulation of the material on the windward wide of the object. The second is the wind intensity decrease behind the object on the leewrad side, which stops the wind-ripples formation in the wind shadow. By adding these interactions well, this method can create more plausible desert environment, but the control of these parameter still are not intuition.

The formation of desert scene in physics contained many factors, but the major factor forming the desert shape is wind. We briefly discuss the previous works in simulating desert scene, but these methods still can be improved in modeling and in rendering. In contrast, wind direction in our method is in 3D space and our control parameters have more physically meaning, make the controlling of result more intuition.

## 2.3 Mesh Representations

The mesh representations are developed for various purposes. Catmull and Clark designed a subdivision scheme to generalize uniform B-Spline knots insertion to meshes of arbitrary topology [9]. Starting with a user-defined mesh of arbitrary topology, it refines the initial mesh by adding new vertices, edges and faces with each step of subdivision following a fixed set of subdivision rules. In the limit, a sequence of recursively refined polyhedral meshes will converge to a smooth surface.

Regular remeshing is the process where an irregular mesh is approximated by a mesh with (semi-)regular connectivity [14]. The simplicity of a regularly remeshed representation has many benefits. In particular it eliminates the indirection and storage of vertex indices and texture coordinates. This will allow graphics hardware to perform rendering more efficiently, by removing random memory accesses and thus improving memory access performance.

Gu et al. introduced the geometry images(GIM) which creates the most regular remeshed representation [18]. Their construction converts the surface into a topological disk using a network of cuts and parametrizes the resulting disk onto a square domain. Using this parametrization, the surface geometry is resampled onto the pixels of an image. As an added benefit, techniques such as image compression can be directly applied to the remesh.

Color mapping represents the color of surface features regardless of any 3D information. Bump and normal mapping manipulate the input normal of lighting computation to generalize the true lighting effects. Displacement mapping map a scalar height function to the mesh surface to represent surface details of meshes[13]. With exactly encoding the 3D information of mesh surface it provides self-occlusion, self-shadows and silhouette. Some surface features such as bricks, sculpture, textiles can be well approximated by the displacement mapping.

CHAPTER **3**

# Method

## 3.1 Approach Overview

We proposed a novel scheme to describe the wind erosion phenomena on arbitrary meshes. In our framework, a displaced mesh is needed. In the preprocessing stage, we converting the original mash into a simplified base mesh and it's corresponding displacement map. Then in runtime stage, we perform the wind erosion simulation and update the displacement map. Finally in the rendering stage, we render the eroded mesh by apply the updated displacement map to the base mesh.Fig 3.1 is our system flow.

In section 3.2, we describe the process of converting a mesh into a displaced mesh. In section 3.3, the simulation equation in our simulation framework is provided. In section 3.4, we discuss our simulation process.
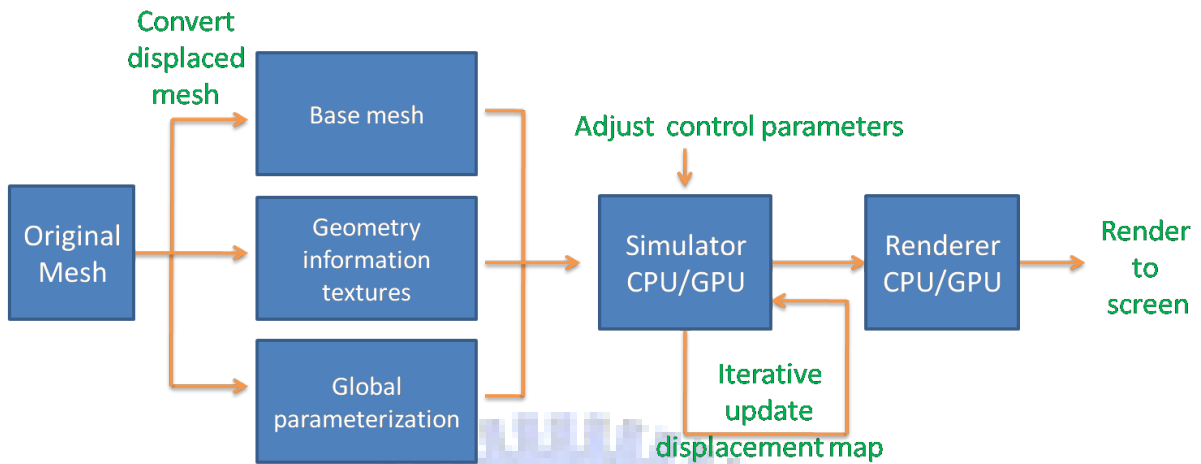
Figure 3.1: An overview of our framework.

## 3.2 Displaced Mesh Construction

In this section, we describe our displaced mesh representation framework, which is originally built by our senior[20]. Here we will describe the whole process.

The goal of this work is to construct the displaced mesh which is a coarse version of input original mesh. The surface details are converted to a displacement map and can be rendered using per-pixel ray-casting methods[29][15][16][28]. And this displaced mesh is used as our simulation domain.

### 3.2.1   Mesh Simplification

To capture the silhouette of original mesh in per-pixel ray casting methods for displacement mapping[15][16][28][29], we need to apply an offset along the outer direction to the surface of original mesh at first. This is done by using the concept of *Simplification Envelopes* developed by Cohen et al.[**?** ]. Then we simplify the offset surface based on the framework of *Progressive Meshes*[19] and uses QEM [17] as our simplification metric. To ensure that the simplified base mesh is valid for displacement mapping, we also introduce two constraints during simplification:

(1) The simplified surface can not intersect the surface of original mesh or the bounding shell. When checking an half-edge collapsing for this constraint, we virtually collapse the half-edge and trace the faces $F_i$ in the one-ring neighborhood of the start vertex of the half-edge. If $F_i$ is discarded after collapsing, ignore it, else check if it intersect the surface of original input mesh or the bounding shell.

(2) The function from simplified surface mapping to the surface of original mesh have to maintain a height-field structure. It means that there should be no height-field overlapping between the base mesh and the original mesh. Such height-field overlapping can be easily identified if there exits a point on original mesh surface whose normal dot the normal of base mesh surface equal to zero. This identifying method is proved to be correct by Collins and Hilton[12].

The first constraint is used to guarantee that the simplified base mesh is tightly in the bounding shell volume of original mesh. The second one ensures a one-to-one mapping between simplified base mesh and the original mesh.

### 3.2.2   Mesh Parameterization

After we obtain the base mesh in low-resolution, we parameterize it using the method proposed by Yoshizawa et al.[30], which is an area- preserving parameterization. We build the parameter-

ization of base mesh under the assumption that surface with larger area can catch more surface details of original mesh, so that we give it more samples by enlarge its parametric area.

### 3.2.3 Displacment Map Generation

With base mash and it's parameterization, we are going to build the displacement map at last. We sample rays from the base mesh surface along the direction of inverse interpolated vertex-normal, computing the distance and the original surface normal of the point at which the ray intersects the surface of original mesh. Then store the original normal vector with the displacement value as a 4-channels luminance to the pixel.

## 3.3 Wind Erosion Simulation Equations

In this section, we will show how to simulate such phenomena by our wind erosion equation. Before we show our wind erosion simulation equation, the Wind Erosion eQuation(WEQ) have to be introduced first. In the environment and soil science field, W. S. Chepil et. al[11] use wind tunnels and field studies to develop the first wind erosion prediction equation. The equation expressed in function form is:

$$E = f(I, K, C, L, V),\tag{3.1}$$

where $E$ is the potential average annual soil loss, $I$ is the soil erodibility index, $K$ is the soil ridge roughness factor, $C$ is the climate factor, $L$ is unsheltered distance across a field, and $V$ is the equivalent vegetative cover. This equation has been used for a long time to predict the soil loss caused by wind erosion in agricultural fields.

Observing the WEQ, we know this equation is designed for predicting the soil loss in geomorphology. Inspired by the WEQ, we design our wind erosion equation which suited for our modeling and rendering wind erosion phenomena on meshes. Our equation is based on these observations:

- The soil loss, $E$, is proportional to wind force($C$ factor and $L$ factor in WEQ)

- The soil loss, $E$, is propositional to objects appearance roughness($K$ factor in WEQ)

- The soil loss, $E$, is propositional to objects intrinsic composition($I$ factor in WEQ)

- The soil loss make mesh surface lost the fine details in height field

- The appearance roughness is actually the degree of surface variance, which we can refer to the curvature over mesh

To sum up, the lost of surface detail is propositional to the wind and the mesh's intrinsic composition combine with it's curvature. Hence we design our wind erosion simulation equation on mesh:

$$H(x) = \frac{Wt(x)}{R(x)}, \qquad (3.2)$$

where $x$ is the sample point on mesh which we are compute; $H(x)$ is the eroded height for this point; $Wt(x)$ is the wind force factor of $x$; $R(x)$ is a function reflect the relationships between eroded height and material of $x$.

And for simplex, here we assume the wind force has a constant effect over the entire mesh. For each sample point on mesh surface, wind force affect erosion quantity more on the upwind side of mesh, but less on the downwind side of mesh. Then we get the $Wt(x)$ equation:

$$Wt(x) = \vec{W_d} \cdot \vec{N_d}(x) + basicW, \qquad (3.3)$$

where $x$ is the sample point on mesh which we are compute; $\vec{W_d}$ is the wind direction in the simulation environment; $\vec{N_d}(x)$ is the average normal direction of the sample point $x$; $basicW$ is a constant denote the basic wind effect of erosion, and it is the first control parameter in our simulation equation.

As we mentioned material of each sample point $x$ in Equation3.2. Material is the property that shows under the variance of curvature, how much soil will lost on the sample point. For example, when sample point $x$ has higher curvature on mesh surface, it means that the mesh structure have less strength to support this sample and this sample would easily lost it's soil by wind force. For lower curvature sample point, it would be harder for wind to blow away the

soil of that sample. But how much soil lost is caused by this kind of material? If the material is stronger, no matter the curvature is high or low, soil lost of the sample is small. But when the material is weaker, a tiny curvature raise would cause high soil lost of the sample. $R(x)$ is designed to represent it. As a consequence, when the material changes, function $R(x)$ changes, too.

To define one $R(x)$, which means to define a new material to curvature mapping relationships, we need three more control parameters: $\mathcal{E}_{min}$, $\mathcal{E}_{medium}$, and $\mathcal{E}_{max}$. $\mathcal{E}_{min}$, $\mathcal{E}_{medium}$, $\mathcal{E}_{max}$ denotes respectively the minimum, the medium, and the maximum soil lost in one time step erosion simulation. In our setup, the minimum soil lost occurs when this material has only basic wind effect and its curvature is 0. The maximum soil lost occurs when this material direct faces the wind force and its curvature is 1. The medium soil lost occurs when curvature of this sample point is 0.5. Hence we get that $R(x)$ will pass three point: $(0, \frac{basicW}{\mathcal{E}_{min}})$ as $(x_0, y_0)$, $(0.5, \mathcal{E}_{medium})$ as $(x_1, y_1)$, and $(0, \frac{1+basicW}{\mathcal{E}_{max}})$ as $(x_2, y_2)$. By using the interpolating polynomial, we get $R(x)$:

$$R(x) = \frac{(x - x_1)(x - x_2)y_0}{(x_0 - x_1)(x_0 - x_2)} + \frac{(x - x_0)(x - x_2)y_1}{(x_1 - x_0)(x_1 - x_2)} + \frac{(x - x_0)(x - x_1)y_2}{(x_2 - x_0)(x_2 - x_1)}, \qquad (3.4)$$

where $x$ is the curvature of the sample point. $R(x)$ return the soil lost ratio of this kind of material under the curvature and wind condition of the sample point.Fig3.2 and Fig3.3 show different $R(x)$.

Note that we consider curvature is signed and the range is $[-1, 1]$. When curvature of sample point is below zero, this sample is on concave part of this mesh. We treat this as an exception, and will discuses in the later section. On the other hand, in order to make $R(x)$ be a monotonic decrease function, once $\mathcal{E}_{min}$ and $\mathcal{E}_{max}$ are given, $\mathcal{E}_{medium}$ should be bound in $[y0, y2]$, and $basicW$ should be bound in $(\frac{\mathcal{E}_{min}}{\mathcal{E}_{max} - \mathcal{E}_{min}}, \infty)$. Either $\mathcal{E}_{medium}$ or $basicW$ are not satisfied their bounding condition will be corrected in system. By well designed $R(x)$, we can satisfied the needs of $H(x)$, and a stable simulation system.

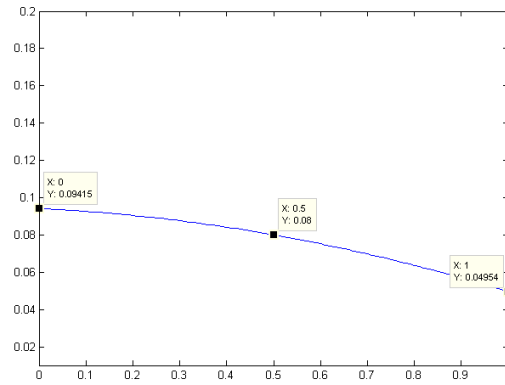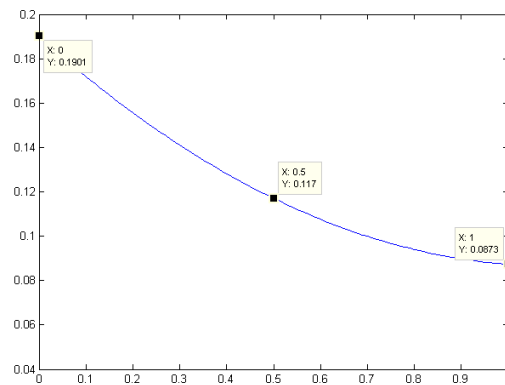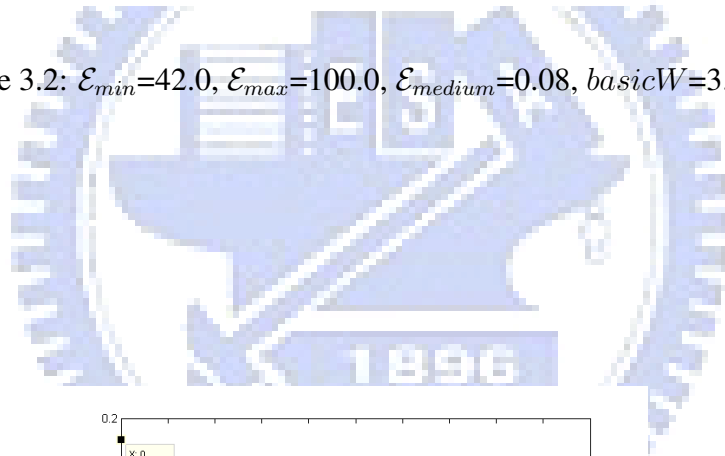Figure 3.2: $\mathcal{E}_{min}$=42.0, $\mathcal{E}_{max}$=100.0, $\mathcal{E}_{medium}$=0.08, $basicW$=3.95



Figure 3.3: $\mathcal{E}_{min}$=20.0, $\mathcal{E}_{max}$=55.0, $\mathcal{E}_{medium}$=0.117, $basicW$=3.80

## 3.4 Simulation

In this section, the simulation process is described. In the preprocessing stage, we get the displacement map through the global parameterization. The global parameterization is actually our simulation domain. And we use this global parameterization to get the needed mesh information. Recall the needed mesh information in our simulation equation3.3, additional two maps, normal map and curvature map, are also generated through the global parameterization. In GPU implementation, the position map is also needed.

As mentioned in 3.3, some control parameters should be given first. Besides, the geometry information in mesh should be updated if they are not ready for our simulation. So before the simulation started, we assign $\mathcal{E}_{min}$, $\mathcal{E}_{max}$, and $\vec{W_d}$, note that $\vec{W_d}$ to all sample point on mesh surface is the same for simplex. Since $\mathcal{E}_{medium}$ and $basicW$ can be computed in system, we have two ways to assign them. We can just use the computed $\mathcal{E}_{medium}$ and $basicW$, or by user assign. Here we choose the automated way. Then we update the average normal and mean curvature for each sample point. For simulation, we need another map(the erosion map) to keep the temporary erosion height of each sample point in one time step simulation. This map is generate through the global parameterization. After these processes, the initiation is complete.

We perform the following instructions in one simulation time step:

1. Based on the displacement map, normal map, curvature map and the control parameters, compute $H(x)$ for each sample point $x$

2. Write $H(x)$ to the erosion map

3. Perform low pass filter on erosion map

4. Update displacement map by adding erosion map

5. Update normal map and curvature map by displacement map and base mesh(CPU method) / position map(GPU method)

6. Render the simulation result by the displaced base mesh and the updated displacement map

and repeat these instructions until we have desired result.

If we detect the curvature of the sample point is negative in instruction 1, we give it a negative erosion height. The point in concave side of mesh will slowly growing height, just like slowly filled with dust. If we detect the sample point is in backwind side of mesh, which refer to the dot product of wind direction and normal of the sample point is below zero, we give it erosion height: $H(x) = basicW$. Only global wind affects the backwind side.

## 3.5   Rendering

Rendering in our simulation framework is really simple. In CPU implementation, we tessellate the displaced base mesh until the vertex number is the same as the displacement map texel number. Then we apply displacement map to tessellated mesh, update all the vertices. Rendering the tessellated mesh in opengl fixed pipeline. In GPU implementation, our framework is especially suit the per-pixel ray-casting methods[15][16][28][29]. Here we use the method proposed by Y.C. Chen and C.F. Chang[10], which reduce the silhouette inaccurate problem in the original relief texture mapping[15] method.

# CHAPTER 4

# Result

All experiments are performed on Intel core 2 due E6750 2.66GHz PC using NVIDIA Geforece 8600 GT graphics hardware. These result are all rendered using CPU. All models contain 16384 vertices. Rendering result all store into $800 \times 800$ resolution image for each time step.

We demonstrate our wind erosion simulation effect on different meshes. These meshes include tores(Fig.4.1), cggm cylinder(Fig.4.2) and relief cylinder(Fig.4.3). Fig.4.4 shows small wind force effects in the backwind side.
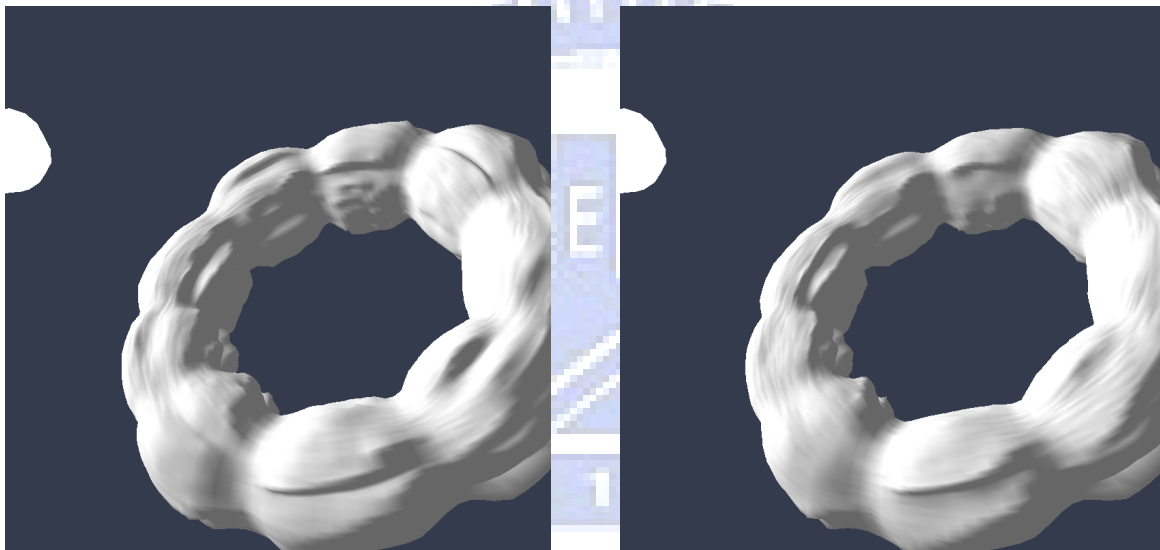
(a) Torus in time step 1                              (b) Torus in time step 10

Figure 4.1: Left side ball denote the wind source. Wind force smooth the left side boundary of torus.

(a) CGGM Cylinder in time step 1

(b) CGGM Cylinder in time step 5

(c) CGGM Cylinder in time step 12

(d) CGGM Cylinder in time step 20

Figure 4.2: Right-up side ball denote the wind source. Notice the CGGM appeared gradually because of different materials

(a) Relief Cylinder in time step 1              (b) Relief Cylinder in time step 10

Figure 4.3: The ball above the relief cylinder denote the wind source. Notice the erosion effect on top of the cylinder and on boundary of the relief



Figure 4.4: Left hand side view of 4.3(b)

**C H A P T E R** $5$

---

# Conclusion

---

In this chapter, we give brief summary of the thesis, and suggest some directions of future work.

## 5.1 Conclusion

We have proposed a novel scheme to simulate wind erosion phenomena on meshes. We use displaced mesh representation as our domain in simulation, and redesign the wind erosion equation to adapt our mesh representation. Previous works[27][8] strict the wind direction in X-Y plane while our system can simulate wind effect in 3D space. Our control parameters are intuitive for artists to add the weathering effect on mesh. Combining gpu-based inverse displacement mapping techniques, we can simulate and render result in real-time.

## 5.2 Future Work

We convert original mesh into displaced mesh in preprocessing stage of our method. This conversion process generates the base mesh which is outer shell of the original mesh and the global parameterization for our connectivity lookup. Then we can do our simulation process. Our displaced mesh representation has benefit in combining the inverse displacement mapping techniques for rendering. Thanks the advance of graphics hardware, we may not need the outer shell in the future. In the DirectX 11 API, the subdivision procedure can be accelerated by new graphics hardware. So we can simplify the original mesh without growing it first, this can shorten the preprocessing time. What even better is that we convert our original mesh into geometry images[18]. With geometry images representation we get the base mesh and the global parameterization at the same time. And we can to the subdivision process in texture with graphics hardware support, hence the whole preprocessing would be more simple and efficiently.

In our wind erosion simulation equation, the time variable does not considered. Because of lack of this time information, our result is more visual plausible way than the physically correct way. Adding the time variable into our equation would make our simulation equation more completely and can simulate the effects in real world time instead of just some time steps. The revisited version of this paper will leave to CGGM Lab, computer science department, National Chiao Tung University. Another work should be done is to examine the ground truth and compare it with our result.

# Bibliography

[1] Gone with the wind. `http://www.islandnet.com/~see/weather/elements/dustwind.htm`.

[2] N. H. Anh, A. Sourin, and P. Aswani. Physically based hydraulic erosion simulation on graphics processing unit. In *GRAPHITE '07: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, pages 257–264, New York, NY, USA, 2007. ACM.

[3] J. H. S. K. B. Bedich Bene, Vclav Tnsk. Hydraulic erosion: Research articles. *Computer Animation and Virtual Worlds*, 17(2):99–108, 2006.

[4] F. Belhadj. Terrain modeling: a constrained fractal model. In *AFRIGRAPH '07: Proceedings of the 5th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 197–204, New York, NY, USA, 2007. ACM.

[5] B. Bene and X. Arriaga. Table mountains by virtual erosion. In *Eurographics Workshop on Natural Phenomena*, pages 33–39, Dublin, Ireland, 2005. Eurographics Association.

[6] B. Bene and R. Forsbach. Parallel implementation of terrain erosion applied to the surface of mars. In *AFRIGRAPH '01: Proceedings of the 1st international conference on Computer graphics, virtual reality and visualisation*, pages 53–57, New York, NY, USA, 2001. ACM.

25

[7] B. Bene and R. Forsbach. Layered data representation for visual simulation of terrain erosion. In *SCCG '01: Proceedings of the 17th Spring conference on Computer graphics*, page 80, Washington, DC, USA, 2001. IEEE Computer Society.

[8] B. Bene and T. Roa. Simulating desert scenery. In *Winter School of Computer Graphics SHORT communication Papers Proceedings*, pages 17–22, 2004.

[9] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. In *Computer Aided Design*, 1978.

[10] Y.-C. Chen and C.-F. Chang. A prism-free method for silhouette rendering in inverse displacement mapping. In *Proceedings of Pacific Graphics '08*.

[11] W. S. Chepil and N. P. Woodruff. The physics of wind erosion and its control. *Advances in Agronomy*, 15:211–302, 1963.

[12] G. Collins and A. Hilton. Mesh decimation for displacement mapping. In *Proc. Eurographics*, 2002.

[13] R. L. Cook. Shade trees. In *Computer Graphics (Proceedings of SIGGRAPH 84)*, 1984.

[14] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proc. 22nd annual conference on Computer graphics and interactive techniques*, pages 173–182, 1995.

[15] P. F., O. M. M., and C. J. Realtime relief mapping on arbitrary polygonal surfaces. In *Proc. ACM Symposium on Interactive 3D Graphics and Games*, pages 155–162, 2005.

[16] P. F., O. M. M., and C. J. Relief mapping of non-height-field surface details. In *Proc. ACM Symposium on Interactive 3D Graphics and Games*, pages 55–62, 2006.

[17] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, aug 1997.

[18] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. In *Proc. ACM Transactions on Graphics*, pages 355–361, 2002.

[19] H. Hoppe. Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 99–108, aug 1996.

[20] Y.-C. Lin. Mesh representation with displacement mapping. Master's thesis, National Chiao Tung University, Hsinchu, Taiwan, ROC, 2006.

[21] J. Lu, A. S. Georghiades, A. Glaser, H. Wu, L.-Y. Wei, B. Guo, J. Dorsey, and H. Rushmeier. Context-aware textures. *ACM Transactions on Graphics*, 26(1):3, 2007.

[22] O. M. M. and P. F. An efficient representation for surface details. Technical Report 351, UFRGS, 2005.

[23] X. Mei, P. Decaudin, and B.-G. Hu. Fast hydraulic erosion simulation and visualization on gpu. In *PG '07: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 47–56, Washington, DC, USA, 2007. IEEE Computer Society.

[24] S. Mrillou and D. Ghazanfarpour. Technical section: A survey of aging and weathering phenomena in computer graphics. *Computers and Graphics*, 32(2):159–174, 2008.

[25] F. K. Musgrave, C. E. Kolb, and R. S. Mace. The synthesis and rendering of eroded fractal terrains. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 41–50, New York, NY, USA, 1989. ACM.

[26] H. Nishimori and N. Ouchi. Formation of ripple patterns and dunes by wind-blown sand. *Phys. Rev. Lett.*, 71(1):197–200, Jul 1993.

[27] K. Onoue and T. Nishita. A method for modeling and rendering dunes with wind-ripples. In *PG '00: Proceedings of the 8th Pacific Conference on Computer Graphics and Applications*, page 427, Washington, DC, USA, 2000. IEEE Computer Society.

[28] N. Tatarchuk. Dynamic parallax occlusion mapping with approximate soft shadows. In *Proc. ACM Symposium on Interactive 3D Graphics and Games*, pages 63–69, 2006.

[29] T. Welsh. Parallax mapping with offset limiting: A perpixel approximation of uneven surfaces. Technical report, Infiscape Corporation, 2004.

[30] S. Yoshizawa, A. Belyaev, and H.-P. Seidel. A fast and simple stretch-minimizing mesh parameterization. In *Proceedings of the Shape Modeling International 2004 (SMI'04)*, pages 200–208, 2004.