Dynamic Simulation of Grass Field Swaying in Wind

Hang Qiu *1,2
1 School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu, China
2 Sichuan Provincial Key Laboratory of Digital Media, Chengdu, China
Email: qiuhang@uestc.edu.cn

Leiting Chen1,2, Jim X. Chen3 and Yugang Liu1,2
1 School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu, China
2 Sichuan Provincial Key Laboratory of Digital Media, Chengdu, China
3 Department of Computer, George Mason University, Fairfax, USA
Email: richardchen@uestc.edu.cn, jchen@gmu.edu, liulional@uestc.edu.cn

***Abstract*—Grass is an essential element of natural scenes, which plays an important role in various fields of applications, such as virtual reality, computer games and special effects of movie. Unfortunately, it is still difficult to render and animate grass with interactive frame rates due to the huge number and wide covering range of grass blades. Realistic simulation of dynamic grass field turns to be one of the most challenging topics in computer graphics. In this paper, we propose a method for dynamic simulation of grass field swaying in wind. The representation of large-scale grassland relies on three different levels of detail that reduce the rendering cost and still allow high-fidelity rendering of grass close to the viewer. To simulate real-time waggle of grasses, some physically based methods and procedural approaches are put forward according to different levels of detail. Experiments demonstrate that our method not only can realistically render the animated grass scenes in wind, but also can support the variable wind field.**

***Index Terms*—natural scenes, grass field, levels of detail, billboard, dynamic simulation**

I. Introduction

Over the past decade, simulation of vegetation in natural scene has been a hotspot and one of the most difficult tasks in computer graphics. Grass is an important component of natural environment, which bears a great application value in many fields, including virtual reality, computer games and landscaping.

However, grass field is composed of millions even billions of grass blades and the full geometric representation is, due to both memory and calculation time constraints, far beyond the capabilities of current graphics hardware. Moreover, in recent years, people are no longer satisfied with the visual effect of static grass scenes, and there has been an increasing demand in interactive vegetation scenes. Nevertheless, due to the huge amount of grass blades and the complexity of physical interaction, dynamic simulation of grass field in real time remains a challenging work.

In this paper, a method for dynamic simulation of grass field under the influence of wind is proposed. We focus on simulating the essential behaviors of grass in wind and looking for the right compromise between realism and efficiency. The major contributions of our work can be listed as follows.

(1) To represent realistic grass blade shape, stiffness coefficient and the effect of gravity are considered, which determine the behavior of the blade close to the viewer.

(2) According to the features of grass blade in real world, a method to simulate the bending and recovery of grass in response to the wind is presented. This method adopts various dynamical equations and deformation strategies to calculate blade deformation. To achieve consistency animation between different LODs, data-sharing scheme is introduced.

II. Related Work

Extensive works made efforts to simulate realistic grass. The related technologies can roughly be classified into two following categories.

A. Rendering of Grass

Geometry-based rendering method is an effective way to represent grass blades precisely. As early as 1985, Reeves et al. [1] used particle system to create grass scene, but the rendering cost is too high to be suitable for real-time applications. Wang et al. [2] adopted 3D skeleton lines and transparent texture mapping to illuminate grass blades with rich details. To accelerate rendering process, they applied the LOD technique. The

Manuscript received April 19, 2011; accepted September 9, 2011.
This work is supported by National High Technology Research and Development Program of China (2007AA010407)

doi:10.4304/jsw.7.2.431-439

LOD representation of grass blades is accommodated automatically to the camera movement.

Compared with geometry-based rendering method, image-based rendering is faster and more approximative that becomes the most common approach for grass field. Reference [3-5] adopted semi-transparent textured billboards to describe grass scenes. Quadrilaterals covered by a semi-transparent 2D texture replace a large amount of geometries that reduces the complexity of scene. Nevertheless, the drawback of billboards is that the quality of appearance is limited, especially at viewpoint closes to the grass. Moreover, billboards do not completely avoid parallax errors. Instead of rendering polygonal billboards, Habel et al. [6] used front-to-back compositing of implicitly defined grass slices in a fragment shader, which avoids storage costs of geometric grass billboards.

Volumetric texture [7-9] is fit for the performance of complex objects with repetitive details, which makes it to be a feasible approach to render grass. In contrast to billboard approaches, it offers more parallax effect. Ray tracing algorithm can be used to implement volumetric textures, but it is a computationally expensive task. Many researchers adopted extended method to construct volumetric texture. Meyer et al. [10] used z-buffer techniques to render 3D geometry that is sliced into a series of thin layers. Reference [11, 12] adopted shell-based method to render grass.

In recent years, looking for the right compromise between realism and efficiency has lead researchers to consider the combination of rendering methods above mentioned. The main idea of the strategy is to use different LODs for representing grass, according to the distance from viewpoint to grass. Zhao et al. [13] used two levels of detail to illuminate grasses. Reference [14-16] relied on three levels of detail to represent grass. They all used geometry representation for nearby grass and a simple texture map for the faraway grass. Where they differ is the mid-level LOD. The other algorithms and more specific surveys can be found in [17].

B. Animation of Grass

Realism of grass depends greatly on whether it is animated or not. There are several works focused on the simulation of grass-wind, grass-object interaction.

Perbet et al. [14] presented procedural animation primitives that implement wind effects. However, this approach does not adhere to physical-based models, and the quality can be further improved. Pelzer [4] proposed three deformation methods for grass based on billboard, but they are limited to simulate grass-wind interaction. In [11], animation is achieved by translating the surface according to a wind vector. Nevertheless, this approach is too simple to support variable wind field. Wang et al. [2] adopted physically based method to calculate deformation of grass in wind, and achieved realistic visual effect. Nevertheless, as the computation based on Hooke's law, when wind force changes significantly, the animation sequence may not be very smooth.

For grass-object interaction, Guerraz et al. [15] presented a method to simulate interaction between grass and moving object. However, it is a procedural approach, which leads to the limitation of visual quality. Orthmann et al. [18] put forward a GPU-based approach to model responsive grass. Most recently, Chen et al. [5] proposed a procedural method to handle the grass interactions with objects. Nevertheless, the grass is only suited to low view, due to the adoption of billboard.

III. LOD Representation of Grass

Compared with trees and other vegetations, the geometry of grass is relatively simple. However, due to the complexity of species, and huge numbers of blades in nature scene, single rendering technique can not simultaneously meet the requirements of both interactive frame rate and high image quality. To find a good compromise, we adopt LOD representation method to construct large-scale grass field.

Similar to [14-16], we rely on three levels of detail, as shown in Fig.1. Point P indicates viewpoint. For grass blades near the viewer, geometry-based representation is adopted to depict grass with rich details. At mid-distance, billboard-based method is applied. For the faraway grasses, 2D texturing approach is used.

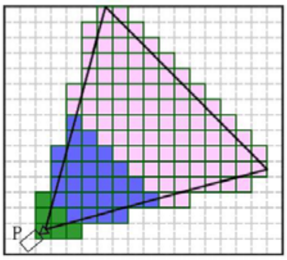


Figure 1. LOD Representation of Grasses

To achieve the effective management of the whole scene, block-based scheme is applied. As shown in Fig.2.

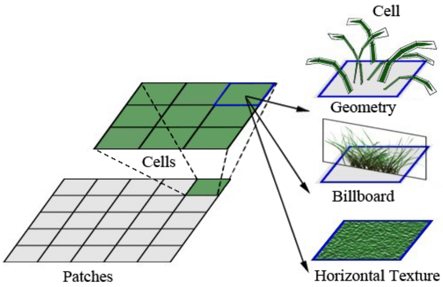


Figure 2. Management based on blocks

A. Geometry-based Representation of Grass

The grasses close to the viewer, which have perceptible exterior characteristics. To show the features of grass with rich details, geometrically modeled grass blades are applied.

As shown in Fig.3. A grass blade is split into several segments. Each segment is assumed to be rigid and can rotate around the joint which links itself to the adjacent segments. We use two-sided quadrilateral strips to approximate a single grass blade. Each quadrilateral corresponds to a segment. In real world, grass blades have full of venation detail, we perform alpha channel of the texture covering the quadrilateral strips to improve visual quality.

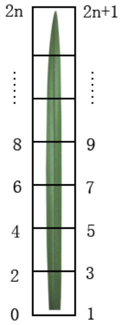


Figure 3. Grass model based on segments

The approach above-mentioned can achieve upright grass blade. However, naturally grass blades have bending shape under the influence of gravity. To depict this feature, reference [13, 16] adopted the trajectory of a particle system, which gives the coordinates of the blade reference points. In order to perform better visual quality, we present a novel method that takes into account stiffness coefficient and gravity factors.

Stiffness coefficient reflects the ability of grass blade to resist deformation. In nature, it is intuitive that the root of grass blade has higher stiffness than the tip, which means that it requires more force to affect the root of grass than the tip. Therefore, we define the stiffness coefficient of grass blade at position *i* as:

$$\mu_i^{base} = \mu^{base}(L) = \mu_{root}^{base} / (L + x) \qquad (1)$$

L depicts the distance from root to position *i*. μ_{root}^{base} is the stiffness coefficient of the root. *x* is compensation. The further the distance of a point from the grass root, the lower the stiffness. Given that stiffness varies throughout the grass blade, the force of gravity has interesting effects on the blade's shape. Fig.4 shows the varying stiffness coefficient and corresponding shape of grass blade under the influence of gravity.

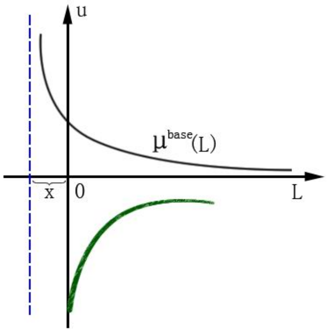


Figure 4. The effect of gravity over grass blade with varying stiffness coefficient

The gravity on each joint of segment can be depicted:

$$\vec{f}^{\,G} = \rho_{grass} \cdot l_{length} \cdot g \qquad (2)$$

Where, ρ_{grass} is the density of grass, l_{length} is the length of each segment, *g* is acceleration of gravity.

For each grass blade, the calculation of bending shape under the influence of gravity is based on the rotation. Specifically speaking, we give each grass blade a random initial inclination α , the final shape is achieved by rotating the segments, as shown in Fig.5.

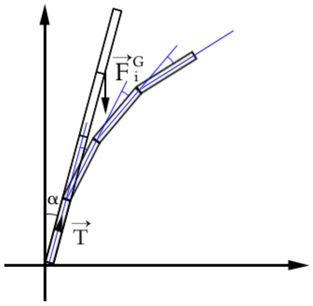


Figure 5. Calculation of bending under the influence of gravity

The gravity on joint *i* can be calculated as:

$$\vec{F}_i^G = (n-i) \cdot \vec{f}^{\,G} \qquad (3)$$

n is the total number of segments. $\vec{T}$ is the unit vector of upright grass blade. Therefore, the torque $\vec{M}_i^G$ due to gravity on joint *i* can be depicted as:

$$\vec{M}_i^G = [(n-i) \cdot l_{length} \cdot \vec{T}] \times \vec{F}_i^G \qquad (4)$$

According to equation (1), the stiffness coefficient of joint *i* is:

$$\mu_i^{base} = \mu^{base}(l_{length} \cdot i) = \mu_{root}^{base} / (l_{length} \cdot i + x) \qquad (5)$$

Since the adjacent segments are considered as rigid, and can rotate around the joint. When the angle between adjacent segments changes, there will be a recovery torque, which is proportional to the angle. Therefore, the bending shape of grass under the influence of gravity can be depicted by the angles between adjacent segments, and the bending angle of segment *i* can be approximately expressed as:

$$\theta_i^{base} = \vec{M}_i^G \cdot \mu_i^{base} \qquad (6)$$

To avoid the repetitive patterns, two approaches are applied.

(1) Representation of various species. Firstly, we set all the length of grass blades to a reference length l_{base}, which controls the average length of grass in the scene. Based on the reference length, we give a stochastic modifier Δl to each blade, where $\Delta l = \lambda \cdot N(0,1)$. λ is the range of absolute value of modifier. The actual length of each blade can be calculated as $l = l_{base} + \Delta l$. Through texture mapping with different grass textures, the rich venation details of various species can be attached.

(2) Representation of stochastic distribution. We define the amount of blades in each cell to control the density of grass field, and give random angle to each blade for representing the orientation of blade.

B. Billboard-based Representation of Grass

Based on the observation that grass at intermediate distance from the viewpoint always forms as clusters, the detail of each blade is imperceptible. Therefore, we treat a cluster of grass as one simulation element and use billboard to represent grass in this region. In each cell, there is a vertical quadrilateral billboard covered with semi-transparency grass pack texture. The length of each billboard is set to be longer than the boundary of the cell. Thus, numerous intersections occur between cells, preventing the creation of artifacts and offering more parallax effect.

To break the repetitive patterns, the actual height of each billboard is determined by the sum of reference height and stochastic modifier. The orientations of billboards are also generated randomly.

C. Horizontal Texture-based Representation of Grass

For grass that is far from the viewpoint, the detail features and dynamic effect are imperceptible. Therefore, we apply classical 2D texturing approach that maps grass-like texture onto the ground to represent grass in these regions.

D. Transition between LODs

Our representation of grass field combines three levels of detail, which makes the transition visible. To avoid popping artifacts appearing, seamless transition scheme is indispensable.

We construct transition region between two levels of detail, in which the adjacent representation methods are used simultaneously. The proportion of each representation is modulated by the weight functions.

In the first transition region, the weight functions for grass defined by billboard and geometry are as follows:

$$K_b(d) = (d - d_{g1})/(d_{g2} - d_{g1}) \tag{7}$$

$$K_g(d) = 1 - K_b(d) \tag{8}$$

d is the distance from viewpoint to the current grass. d_{g1} depicts the distance from the near boundary of transition region to viewpoint, and d_{g2} is the distance from the far boundary of transition region to viewpoint. They are adjustable parameters, which determine the range of transition region and the rendering quality. $K_b(d)$ is used to control the opacity of billboard, and $K_g(d)$ is adopted to modulate the number of geometry-based grass blades *N*. *N* can be calculated as follows:

$$N = \lfloor num \cdot K_g(d) \rfloor \tag{9}$$

num is the maximal number of grass blades in a cell when only the geometry-based representation is used.

Likewise, in the second transition region, the weight functions for grass defined by 2D texture and billboard are as follows:

$$K_t(d) = (d - d_{s1})/(d_{s2} - d_{s1}) \tag{10}$$

$$K'_b(d) = 1 - K_t(d) \tag{11}$$

d_{s1} and d_{s2} depict the distance from viewpoint to the near and far boundary of transition region. $K_t(d)$ and $K'_b(d)$ determine the opacity of 2D texture and billboard respectively.

IV. ESTABLISHMENT OF WIND FIELD

To simulate grass field swaying in wind, it is essential to establish the model of wind field. Wind is the flow of gases on a large scale, which can be modeled by Navier-Stokes differential equations or microscopic mechanism based on statistical physics. However, the computational complexity of these approaches can not be ignored.

We use two-dimensional vector field to construct wind field. The wind velocity and wind direction are depicted by the magnitude and direction of the vector respectively. The wind velocity is composed of two parts:

(1) Mean wind, which reflects the major wind force and direction, and belongs to steady wind.

(2) Turbulent wind, which denotes the airflow fluctuating.

The wind velocity can be expressed as:

$$v(x, y; t) = \bar{v}(x, y; t) + v'(x, y; t) \tag{12}$$

Where $\bar{v}$ is mean wind velocity, while v' is turbulent wind velocity. Fig.6 shows the process of establishment of wind field.

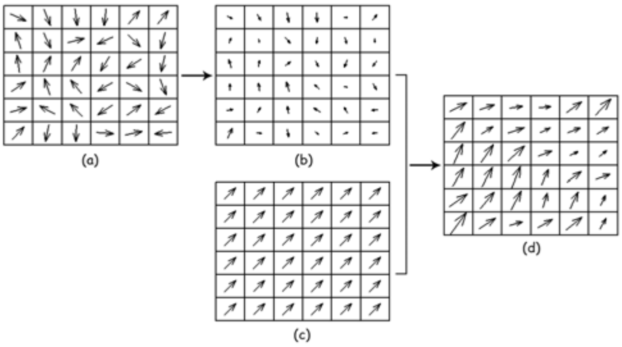


Figure 6. Grass model based on segments

As shown in Fig.6 (a). Firstly, we divide the region of wind field into a series of square units with the same size as cell. For each square unit, two-dimensional random vector is generated as the initial value of turbulent wind, which follows Gaussian distribution. To smooth the turbulent wind field, weighted average filter is adopted. For each initial turbulent wind vector $v'_0(x_i, y_i; t)$, we calculate the weighted average with the vectors around current unit. The equation can be expressed as:

$$v'(x_i,y_i;t)=\frac{\sum_{(x,y)\in B} v_0'(x,y;t)\cdot\frac{\lambda}{k\cdot d^2}+\lambda\cdot v_0'(x_i,y_i;t)}{\sum_{(x,y)\in B}\frac{\lambda}{k\cdot d^2}+\lambda} \qquad (13)$$

B is the range of filter that defined by user, it determines the smoothness of wind field. $\lambda / k\cdot d^2$ is used to calculate the weights of surrounding square units. λ is adjustable coefficient, which depicts the weight in current center square unit. d is the distance from current center square unit to other surrounding units. k is proportional coefficient that determines the proportion of the weights in center square unit and surrounding units.

In our implementation, B is defined to the eight units that surround the current unit, the value of λ is 4, and $k=2/a^2$, a is the edge length of unit. The definition of weights is shown in Fig.7.

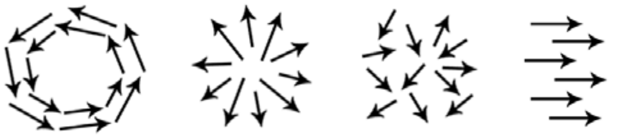
1	2	1
2	4	2
1	2	1

Figure 7. The definition of weights in unit

We use equation (13) and the weights above mentioned to achieve smooth turbulent wind field, as shown in Fig.6 (b).

Mean wind determines the major wind force and direction. Fig.6 (c) illuminates mean wind field. The final wind field is the combination of mean wind field and turbulent wind field. As shown in Fig.6 (d).

We have established four types of wind, including whirlwind, blast of wind, gentle breeze, and moderate breeze.



(a) Whirlwind (b) Blast of wind (c) Gentle breeze (d) Moderate breeze

Figure 8. Different types of wind

V. Animation of Grass in Wind

To simulate grass interacting with wind realistically, there are two issues must be considered. On the one hand, when wind occurs, grass should bend in the direction of wind. One the other hand, when wind stops, the bended grass will stop bending and oscillate back and forth before recovering to a rest pose.

To achieve the above effects, we have proposed different animation schemes respectively according to the different representations of grass. For grasses close to the viewpoint, physically based approach is used to illuminate the bending and recovery of individual grass blade. For grasses at intermediate distance from viewpoint, procedural approach is adopted. In addition, the smooth transition of animation between different representations is also considered. We do not provide any animation scheme to faraway grasses, because the animation is not visible.

A. Animation of Geometry-based Grass

As mentioned in Section III, the geometry-based grass blade is split into several segments. Each segment is assumed to be rigid and can rotate around the joint which links itself to the adjacent segments. The movement of grass can be depicted by the angles between segments, as shown in equation (6). We propose the following approaches to simulate grass bending and recovery respectively.

(1) Simulation of grass bending and twisting in wind

For a single segment of grass blade, according to the momentum law:

$$F_i^{wind}\cdot t = M_{wind}\cdot v \qquad (14)$$

Where F_i^{wind} is the wind force on the segment i, v is wind velocity, M_{wind} depicts the mass of wind.

$$M_{wind}=\rho_{wind}\cdot v\cdot t\cdot S \qquad (15)$$

ρ_{wind} is the wind density, S is the area of thrust surface of current segment, which can be evaluated as:

$$S = l_{length}\cdot l_{width}\cdot \sin\varphi \qquad (16)$$

$l_{length}\cdot l_{width}$ is the area of segment, φ is the angle between wind direction and current segment. According to the equilibrium condition of forces on segment i, the force that makes grass bending can be expressed as:

$$F_i^{bend}=F_i^{wind}=v^2\cdot\rho_{wind}\cdot l_{lengh}\cdot l_{width}\cdot\sin\varphi \qquad (17)$$

The corresponding torque of F_i^{bend} is:

$$\vec{M}_i^{bend}=F_i^{bend}\cdot l_{length} \qquad (18)$$

Similarly, the force that makes grass twisting can be calculated as:

$$F_i^{twist}=v^2\cdot\rho_{wind}\cdot l_{lengh}\cdot l_{width}\cdot\cos\varphi \qquad (19)$$

The corresponding torque of F_i^{twist} is:

$$\vec{M}_i^{twist}=F_i^{twist}\cdot l_{length}\cdot\sin(\theta_i(t)+\theta_i^{base}) \qquad (20)$$

Where θ_i^{base} is the angle between current segment and vertical orientation in the rest state, $\theta_i(t)$ is the angle of current twisting.

Obviously, we could use equations (6) (18) (20) to calculate the bending and twisting angles of each segment, and then perform the deformation of grass blade.

However, equation (6) only provides the angle in force equilibrium condition. When wind force and direction change significantly between adjacent frames, the animation sequence may not be very smooth.

To tackle with this problem, we present a new method that contains three steps. Firstly, calculate equilibrium angle using equation (6). Then, obtain the bending or twisting velocity. Finally, calculate real time bending or twisting angle. The main idea of our method is to keep the variation of angle between adjacent frames within an acceptable range.

Assumed that $\mu_i(t)$ is the current stiffness coefficient of joint *i*, $\mu_i(0)=\mu_i^{base}$. $\theta_i(t)$ is the current bending angle of segment *i*, $\theta_i(0)=0$.

Step1. The calculation of equilibrium angle

$$\overline{\theta}_i(t)=\vec{M}_i^{bend}(t)/\mu_i(t) \tag{21}$$

To reflect nonlinear feature of stiffness coefficient, real time stiffness coefficient is composed of two parts: basic stiffness coefficient and compensating coefficient.

$$\mu_i(t)=\mu_i^{base}+\mu_i^{extra}(t) \tag{22}$$

$\mu_i^{extra}(t)$ is the compensating coefficient, which is proportionate to bending angle of previous frame.

$$\mu_i^{extra}(t)=k\cdot\theta_i(t-\Delta t) \tag{23}$$

$\theta_i(t-\Delta t)$ is the bending angle of previous frame, k is proportional coefficient. We can then rewrite equation (21) as:

$$\overline{\theta}_i(t)=\vec{M}_i^{bend}(t)/[\mu_i^{base}+k\cdot\theta_i(t-\Delta t)] \tag{24}$$

Step2. The calculation of bending velocity

The angle variation range relative to previous frame is:

$$\Delta\theta_i(t)=\overline{\theta}_i(t)-\theta_i(t-\Delta t) \tag{25}$$

The bending velocity can be represented as follows:

$$V_i^{\theta}(t)=\frac{\Delta\theta_i(t)}{\Delta t}\times\frac{\vec{M}_i^{bend}(t)}{\gamma} \tag{26}$$

γ is adjustable coefficient.

Step3. The calculation of real time bending angle

$$\theta_i(t)=\theta_i(t-\Delta t)+V_i^{\theta}(t)\Delta t \tag{27}$$

We can use the similar process above mentioned to calculate twisting velocity and the real time twisting angle.

(2) Simulation of grass recovery

When wind stops, the grass will sway back and forth, especially in the upper part of the blade. To achieve this effect, we divide the grass blade into two parts automatically in real-time. The calculation of critical point is as follows:

$$n_{boundary}=n\cdot e^{-(\frac{1}{2n}\sum_{i=1}^{n}(|\theta_i(t)|+|\phi_i(t)|)-k)\lambda} \tag{28}$$

Where n is the number of segments, $\theta_i(t)$ is the bending angle of segment *i* relative to equilibrium position, $\phi_i(t)$ is the twisting angle of segment *i* relative to equilibrium position. k and λ are empirical coefficients.

For those segments, whose number is larger than $n_{boundary}$, vibration equations are adopted.

When wind fails, the external force is zero:

$$M_{grass}\ddot{\theta}_i(t)+C\dot{\theta}_i(t)+K\theta_i(t)=0 \tag{29}$$

$$M_{grass}\ddot{\phi}_i(t)+C\dot{\phi}_i(t)+K\phi_i(t)=0 \tag{30}$$

M_{grass} is the mass of single segment, K is the stiffness coefficient of joint *i*, $K=\mu_i(t)$. C is the damping coefficient, which can be calculated as:

$$C=\alpha M_{grass}+\beta K \tag{31}$$

For bending animation, at each time step, we apply equations (32) to solve linear vibration:

$$\begin{cases}\ddot{\theta}_i(t+\Delta t)=-(C\dot{\theta}_i(t)+K\theta_i(t))/M_{grass}\\ \dot{\theta}_i(t+\Delta t)=\dot{\theta}_i(t)+\Delta t\ddot{\theta}_i(t+\Delta t)\\ \theta_i(t+\Delta t)=\theta_i(t)+\Delta t\dot{\theta}_i(t+\Delta t)\end{cases} \tag{32}$$

The similar approach can be adopted to calculate twisting angle.

B. Animation of Billboard-based Grass

For billboard-based grass, precise simulation is unnecessary. Therefore, we adopt procedural approach by moving the upper part of billboard to implement animation effect. The only challenge that arises is to achieve consistency animation between different LODs.

In our implementation, each cell stores two types of data: bending and twisting angle. The grass in the same cell will share these data, no matter the representation is geometry-based or billboard-based. As shown in Fig.9. In the two-dimensional space, the calculation of two-dimensional deformation vector M is:

$$M=(L\cdot\sin\omega,-(L-L\cdot\cos\omega)) \tag{33}$$

L is the height of billboard, ω is the sum of bending angle of each segment in this cell.

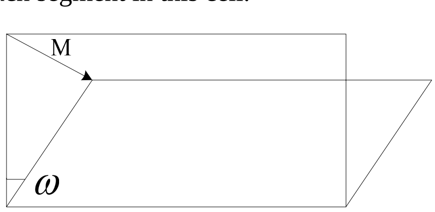


Figure 9. Animation of billboard

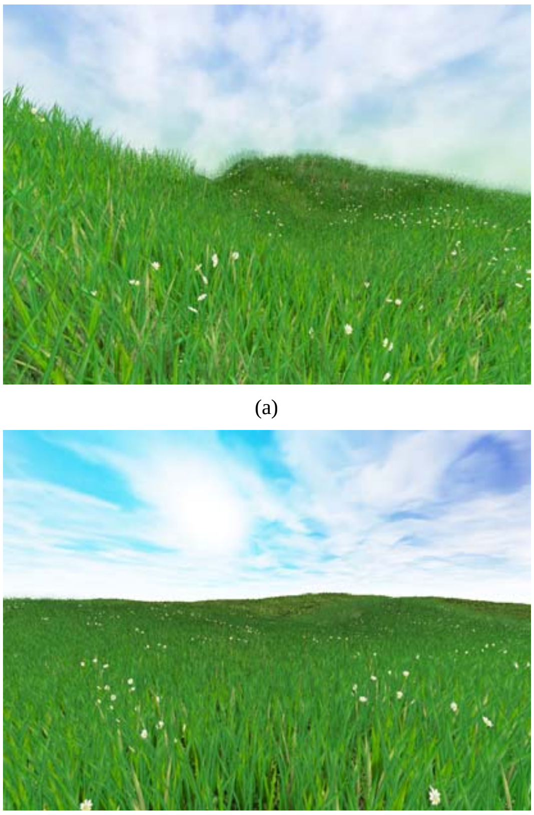
We extend above calculation process to three-dimensional by wind velocity vector V in current cell. So, the three-dimensional deformation vector M' can be expressed as:

$$\begin{aligned} M' = (&L \cdot \sin\omega \cdot Normal\ (V).x, \\ &L \cdot \cos\omega - L, \\ &L \cdot \sin\omega \cdot Normal\ (V).y) \end{aligned} \tag{34}$$

VI. Experiments and Discussions

We use OpenGL on C++ platform to implement the method presented in this paper, and carry out related experiments on PC. The computer configuration used for experiments is Windows XP operating system, Intel Pentium 2.6GHz CPU, 2G memory, and NIVDIA 9800GT graphics card.

The performances of large-scale static grass field using LOD representation are shown in Fig.10, which have little perceivable loss in image quality.



(a)

(b)

Figure 10. The snapshots of grass field

To testify the efficiency of LOD representation of grass field, we carried out a series of pressure tests. Firstly, we constructed seven grass fields, which can represent different amount of grasses from 500,000 to 2,000,000. The resolution of the terrain is 1024×1024 and the fly-through path is predefined. We did four groups experiments with different numbers of segments in geometry-based representation region. We sampled 5000 frames in each scene respectively, and achieved the average frame rate. Fig.11 shows the result.

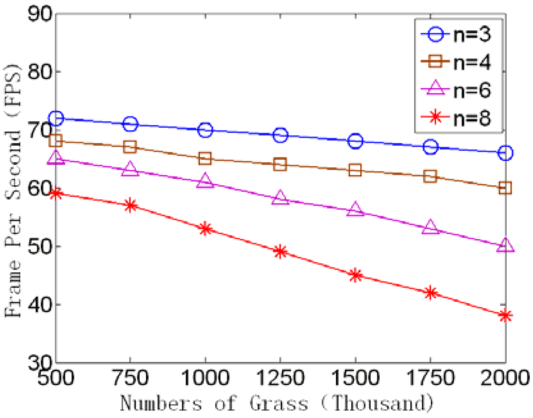
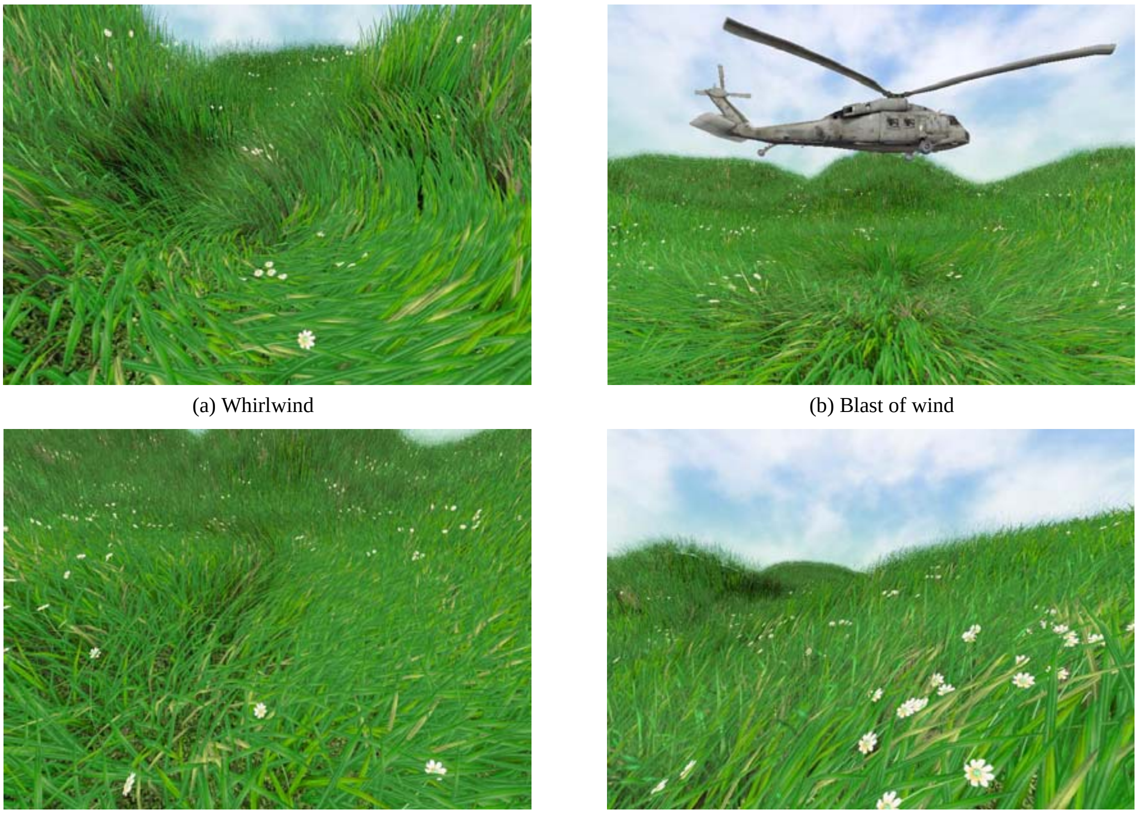


Figure 11. Rendering efficiency

As shown in Fig.11, with the increase of grass density, the rendering efficiency decreased. Moreover, the number of segments has considerable influence on rendering speed. In general, when the number of segments is between 4 and 6, it is sufficient to meet the requirement of visual quality. In our experiment, when the number of segments is up to 8, we still achieve 38fps for the representation of 2,000,000 grasses.

Furthermore, we set the grass density as a fixed value, i.e. each patch includes 5600 grasses, and constructed different scales of terrain. Due to the view frustum culling and LOD representation of grass field, the computational cost can be controlled effectively. With the increase of the terrain size, the reduction of rendering speed is unobvious. When the number of segments is 4, to represent about 6,000,000 grasses, the average rendering speed is still 57fps.

To validate the efficiency of dynamic simulation of grass field in wind, we established a scene that can represent about 500,000 grasses. Specifically speaking, the terrain with resolution of 1024×1024 is divided into 32×32 patches, each path contains 16×16 cells, and each cell can represent two grass blades. The rendering speed ranges from 16fps to 40fps. The speed varies in this range with the scale of wind field and the viewer position. We also tested the influence of variable wind field on animation quality. Result shows that no matter how the wind changes, we can still achieve smooth animation. Fig.12 illuminates the dynamic effect of grass in different types of wind.



(a) Whirlwind

(b) Blast of wind

(c) Gentle breeze

(d) Moderate breeze

Figure 12. Animation performance in different types of wind

Moreover, we compared our method with Wang et al. [2], Perbet et al. [14], Guerraz et al. [15] and Boulanger et al. [16]. The results are demonstrated in TABLEI.

As shown in TABLE I, reference [2] applied Hooke's law to calculate the deformation of grass blade in wind. However, when wind force changes significantly, the animation sequence may not be very smooth. In our approach, we deal with this problem by keeping the variation of angle between adjacent frames within an acceptable range. Although [14][15] adopted LODs to represent large-scale grass field, but the animation between different LODs lacked of consistency. We achieve consistency animation by sharing deformation data for various representation methods in the same cell. Furthermore, our method provides the realistic simulation of grass recovering process in response to the wind, which greatly enhances the reality of natural scene. The animation approach adopted by reference [16] is to translate each top vertex of the vertical slice by a time-dependent wind vector. It is a simple procedural animation, which leads to a limited quality of appearance.

TABLE I.
COMPARISON WITH OTHER METHODS

	Representation of grass	Dynamic simulation method	GPU accelerate	Consistency animation between LODs	Simulation of grass recovery	Smooth animation under variable wind field
Wang et al. [2]	Geometry-based	Physically based	√	√	×	×
Perbet et al. [14]	Three levels of detail	Procedural	×	×	×	×
Guerraz et al. [15]	Three levels of detail	Procedural	×	×	×	×
Boulanger et al.[16]	Three levels of detail	Procedural	√	√	×	√
Our method	Three levels of detail	Physical+Procedural	√	√	√	√

VII. CONCLUSION

In this paper, we have proposed a method for realistic simulation of grass field swaying in wind. To allow for a large range of grass field, LOD representation of grass is introduced. To represent geometry-based grass blades, which may include various species and shapes, stiffness coefficient and the effects of gravity are considered, which determine the behaviors of the blade close to the viewer. According to the different representation methods, physical and procedural animation schemes are presented respectively, which not only can implement the movement of grass in wind, but also display the recovering process naturally. In addition, the consistency animation between different LODs is achieved.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers for their careful reading of this paper. This work was supported in part by a grant from National High Technology Research and Development Program of China (2007AA010407).

REFERENCES

[1] W. Reeves, R. Blau, “Approximate and probabilistic algorithms for shading and rendering structured particle systems”, *Computer Graphics*, vol.19, no.3, pp.313-322, 1985.

[2] C. B. Wang, Z. Y. Wang, Q. Zhou, et al, “Dynamic modeling and rendering of grass wagging in wind”, *Computer Animation and Virtual Worlds*, vol.16, pp.377-389, 2005.

[3] D. Whatley, “GPU Gems 2”, Addison-Wesley, pp.7-25, 2005.

[4] K. Pelzer, “GPU Gems: Rendering countless blades of waving grass”, Addison-Wesley, pp. 107-121, 2004.

[5] K. Chen, H. Johan, “Real-time continuum grass”, *In Proc. VR’10*, Waltham, USA, 2010, pp.227-234.

[6] R. Habel, M. Wimmer, S. Jeschke, “Instant animated grass”, *Journal of WSCG*, vol.15, no.1, pp. 123-128, 2007.

[7] J. T. Kajiya, T. L. Kay, “Rendering fur with three dimensional texture”, *In Proc. Siggraph’89*, Boston, USA, pp.271-280, 1989.

[8] P. Decaudin, F. Neyret, “Rendering forest scenes in real-time”, *In Proc. Rendering Techniques’04*, Norrkoping, Sweden, pp.93-102, 2004.

[9] Y. Y. Chen, H. Lin, H. Q. Sun, et al, “Construction and realistic rendering of scenes with highly-complex plants”, *Chinese Journal of Computers*, vol.23, no.9, pp.917-925, 2000.

[10] A. Meyer, F. Neyret, “Interactive volumetric textures”, *Proc.Eurographic’98*, Lisbon, Portugal, pp.157-168, 1998.

[11] B. Bakay, W. Heidrich, “Real-time animated grass”, *In Proc. Eurographics’02 (short paper)*, Saarbrucken, Germany, 2002.

[12] S. Banisch, C. A. Wuthrich, “Making grass and fur move”, *Journal of WSCG*, vol.14, pp.25-32, 2006.

[13] X. K. Zhao, F. X. Li, S. Y. Zhan, “Real-time animating and rendering of large scale grass scenery on GPU”, *In Proc. ITCS’09*, Kiev, Ukraine, pp.601-604, 2009.

[14] F. Perbet, M. P. Cani, “Animating prairies in real-time”, *In Proc.I3D’01*, North Carolina, USA, pp.103-110, 2001.

[15] S. Guerraz, F. Perbet, D. Raulo, et al, “A procedural approach to animate interactive natural sceneries”, *In Proc.CASA’03*, New Jersey, USA, PP.73-78, 2003.

[16] K. Boulanger, S. N. Pattanaik, K. Bouatouch, “Rendering grass in real time with dynamic lighting”, *IEEE Computer Graphics and Applications*, vol.29, no.1, pp.32-41, 2009.

[17] H. Qiu, L. T. Chen, J. X. Chen, et al, “Survey on realistic simulation of grassland”, *Journal of Computer-Aided Design & Computer Graphics*, vol.22, no.9, pp.1628-1638, 2010.

[18] J. Orthmann, C. Rezk-Salama, A. Kolb, “GPU-based responsive grass”, *Journal of WSCG*, vol.17, pp.65-72, 2009.

Hang Qiu was born in 1978. He received his PhD. degree from University of Electronic Science and Technology of China in 2011. He is a member of China Computer Federation. Currently, he is working in the Provincial Key Laboratory of Digital Media and a lecturer in University of Electronic Science and Technology of China. His research interests include scientific visualization, computer graphics and virtual reality.

Leiting Chen was born in 1966. He is currently a professor and Ph.D. advisor of computer science Dept., University of Electronic Science and Technology of China (UESTC). He received his Ph.D. degree from UESTC in 2007. He is a professional member of China Computer Federation. His research interests include virtual reality, computer graphics.

Jim X. Chen was born in 1962. He is currently a professor and Ph.D. advisor of computer science Dept., George Mason University. He received his Ph.D. degree from University of Central Florida (UCF) in 1995. He is a senior member of IEEE and a professional member of ACM. He served as associate editor-in-chief of AIP/IEEE Computing in Science & Engineering (CiSE). His research interests include computer graphics, virtual reality.

Yugang Liu was born in 1982. He received his PhD. degree from University of Electronic Science and Technology of China in 2011. His research interests include image processing, medical image visualization.