Quasi-physical Simulation of Large-scale Dynamic Forest Scenes

Long Zhang Chengfang Song Qifeng Tan Wei Chen Qunsheng Peng

State Key Lab of CAD&CG, Zhejiang University, 310027, Hangzhou, China
{lzhang,songchengfang,tanqifeng,chenwei**,peng}@cad.zju.edu.cn

Abstract. This paper presents a quasi-physically based approach for interactively simulating large-scale dynamic forest scenes under different wind conditions. We introduce theories from the *wind engineering*, and model the natural wind field as a stationary stochastic process. To reduce the geometry complexities without sacrificing much image quality, we adopt a hybrid geometry/image representation scheme to faithfully model the appearance of trees. Some simplified mechanical rules are employed to compute the movement of such tree models. Three kinds of level of details concerning the scene geometry, the movement of trees and the wind field, are exploited to accelerate the simulation. For forest scenes with tens of thousands of animated trees, our implementation with programable graphics hardware achieves visually plausible results at interactive frame rates on consumer PC platforms.

1 Introduction

Over the past decade, simulation of natural phenomena has become an indispensable requirement for a wide variety of applications including environment assessment, video games and virtual reality. One of the hot research topics is the modeling and rendering of large-scale forest scenes. Although lots of approaches have been extensively studied, the rapidly growing demands on the scene complexity, the physical fidelity and the visual realism, still overwhelm the capabilities of current solutions.

The major difficulty of the simulation of large-scale dynamic forest scenes is of course due to the high complexity in both geometry and appearance. Few previous approaches can simultaneously meet the requirements of both interactive rendering speed and high image quality. In time-critical applications such as video games, one common way is to represent trees with billboards. This technique provides extremely high rendering speed yet poor image quality, and apparently, cannot support animations of trees. On the other hand, if the image quality is of primary importance, one may rely on detailed geometric models of trees. However, as millions of triangles are required to faithfully represent the shape details of one tree, accurate simulation of physical behavior of a forest under the wind seems to be a formidable task on today's PC.

** The corresponding author.

In this paper, we propose a quasi-physically based approach for interactive animation of large-scale forest scenes under adjustable wind conditions. Our contributions lie in several aspects. Firstly, we introduce a physical fidelity model of the wind field for realistic tree animations. Secondly, we provide simplifications for both the geometric model and physical rules, which greatly reduces the simulation cost. Lastly, we exploit novel LOD techniques which efficiently simplifies the computation involved in tree animations.

2 Related Work

There are a variety of well-established approaches for modeling trees. Lindenmayer and Prusinkiewicz[1] introduced the L-system to construct the geometric models of plants. Thereafter, Aono and Kunii[2] improved this method for 3D cases. The fractal approach by Oppenheimer[3] provides another solution for plant modeling. All these methods are based on the common assumption that complex plant geometry may be approximated with simple rules plus stochastic variation. Emphasizing the overall geometrical structure of tree, Weber and Penn[4] presented a parameterized procedural model which provides a more direct and intuitive control interface over the shape of trees.

Image-based rendering techniques have demonstrated advantage that the resultant models are independent of the geometric complexity. Recently, the Billboard Clouds technique was introduced to simplify the plant models by Bromberg[5]. Though they achieve high image quality, they cannot support dynamic simulations of large-scale forest under different wind conditions.

Much work has been conducted well on the dynamic simulation of trees. Ono[6] proposed to adopt Perlin noise to generate the turbulence and calculate the motion of tree based on mass-spring model. Sakaguchi *et al*[7] assumed branch segments as rigid sticks, calculated the rotation of each branch independently and integrate all movements for the motion of a whole tree. By representing the wind forces as $1/f^{\beta}$ noise, Ota *et al*[8] modeled branches as cantilever flat-springs and computed the rotations of leaves by using $1/f^{\beta}$ noise function directly. Coupling stochastic approaches and dynamics equation, Shinya and Fournier[9] synthesized realistic motion of trees, grass and snow under the influence of the wind field. Likewise, Stam[10] synthesized turbulence by filtering a white noise in the Fourier domain and solve the displacements directly. Giacomo *et al* [11] combined procedural approach and physically based method to animate a moderate sized forest. Recently developed commercial software SpeedTree[12] provides appealing results for real-time walk-through of large-scale forest scenes.

3 Construction of The Wind Field

We derive a wind field model from the wind engineering. The key component of our model is a stochastic process that faithfully mimics the stochastic properties of the wind field. This contributes to the visual and physical realism of the trees swaying in the wind.

Wind Velocity Vector: From the viewpoint of the mechanics, the velocity vector of a three-dimensional air flow is composed of three orthogonal components[13], namely, the longitudinal component $U(t)$ along the mean wind direction, the lateral component $V(t)$, and the vertical component $W(t)$. $U(t)$ is the sum of the mean component, denoted by $\bar{U}(z)$ and fluctuating component, while $V(t)$ and $W(t)$ have only fluctuating components. The wind velocity at point $\mathbf{Q}(x, y, z)$ can be represented as:

$$U(\mathbf{Q};t)=\bar{U}(z)+u(\mathbf{Q};t), \quad V(\mathbf{Q};t)=v(\mathbf{Q};t), \quad W(\mathbf{Q};t)=w(\mathbf{Q};t) \tag{1}$$

Cross-Power Spectrum Density Matrix: We model each fluctuating component of the velocity vector by a stationary Gaussian stochastic process. Their spatial-temporal properties in the frequency domain are represented by Cross-Power Spectral Density Matrix($CPSDM$):

$$\mathbf{S}_\epsilon(\omega) = \begin{bmatrix} s_{\epsilon_1\epsilon_1}(\omega) & s_{\epsilon_1\epsilon_2}(\omega) & \dots & s_{\epsilon_1\epsilon_n}(\omega) \\ s_{\epsilon_2\epsilon_1}(\omega) & s_{\epsilon_2\epsilon_2}(\omega) & \dots & s_{\epsilon_2\epsilon_n}(\omega) \\ \vdots & \vdots & \ddots & \vdots \\ s_{\epsilon_n\epsilon_1}(\omega) & s_{\epsilon_n\epsilon_2}(\omega) & \dots & s_{\epsilon_n\epsilon_n}(\omega) \end{bmatrix} \qquad (\epsilon = u, v, w) \tag{2}$$

where ω is the angular frequency, and $s_{\epsilon_j\epsilon_k}$ is the cross-power spectrum density:

$$s_{\epsilon_j\epsilon_k}(\omega)=\sqrt{s_{\epsilon_j\epsilon_j}(\omega)s_{\epsilon_k\epsilon_k}(\omega)}\, Coh(\mathbf{Q}_j, \mathbf{Q}_k; \omega) \tag{3}$$

Each $s_{\epsilon_j\epsilon_j}$ is normally represented by various formula which serve different applications[14]. After investigating all representations collected by Solari *et al*[15], we derive a new unified form: which serves for assessing and choosing the coefficients suitable to our case.

$$s_{\epsilon_j\epsilon_k}(\omega) = \frac{U_*^2 A_\epsilon \nu^\gamma}{(\omega/2\pi)[1 + B_\epsilon \nu^\alpha]^\beta} \tag{4}$$

where $\nu=\omega z/2\pi\bar{U}(z)$ is the Monin coordinate, U_* is the shear velocity, and A_ϵ, B_ϵ, α, β, γ are adjustable coefficients.

The coherence function (Coh) in Equation 3 is expressed as follows:

$$Coh(\mathbf{Q}_j, \mathbf{Q}_k; \omega)=\exp\left\{-\frac{\omega \sum_r C_{r\epsilon} \mid r_j - r_k \mid}{\pi\left[\bar{U}_j + \bar{U}_k\right]}\right\} \qquad (r=x, y, z) \tag{5}$$

Here, $C_{r\epsilon}$ is the exponential decay coefficient.

Evaluation of Wind Velocity: In most cases, $CPSDM$ is real symmetric and positive definite. Thus we decompose it using the Choleski method:

$$\mathbf{S}_\epsilon(\omega)=\mathbf{H}(\omega)\,\mathbf{H}^*(\omega)^T \tag{6}$$

where $\mathbf{H}(\omega)$ is a lower triangular matrix. Then the fluctuating component at time t can be expressed as:

$$\epsilon_i(\mathbf{Q}_i;t) = 2\sum_{j=1}^{N}\sum_{k=1}^{i}\Big[h_{ik}(\omega_j)G_j^{(k)}(t)\Big]\sqrt{\Delta\omega} \qquad (\epsilon = u, v, w)\ (i = 1, 2, ..., n) \quad (7)$$

where $\omega_j{=}j\Delta\omega$, $N\Delta\omega{=}\omega_u$ is the upper cut-off frequency. $G_j^{(k)}$ is a random value.

Finally, the wind force at $\mathbf{Q}(x, y, z)$ can be calculated as:

$$\mathbf{F}(\mathbf{Q};t) = \tfrac{1}{2}\rho\,\|\mathbf{T}\|\cdot\mathbf{T}, \quad \mathbf{T} = (U, V, W) \quad (8)$$

where ρ is the air density.

4 Quasi-Physically based Animation of Trees

Our quasi-physically based animation scheme is based on a hybrid geometry/image representation of trees.

4.1 The Hybrid Representation of Trees

Conceptually a tree consists of branches and leaves. Branches are represented by geometry primitives with associated textures. Leaves are clustered and represented as a list of billboards. Compared with traditional pure geometry based or pure image based representations, our hybrid tree representation is a good balance in terms of rendering quality and efficiency. Figure 1(a) illustrates an example with the proposed hybrid representation.

Models with the hybrid representation can be derived from existing tree models conveniently. In practice, we build a parametric modeling system to generate various tree models. The adopted parameters that control the shape of the branches are similar to those introduced by Weber and Penn[4]. Leaf clusters are generated around the branches. Each leaf cluster is tied to a branch at a hanging point. The distribution of leaf clusters are determined by global parameters. The textures for the leaf billboards can be taken from real captured images or created by artists.

4.2 Quasi-physically Based Animation

We approximate the movement of branches with a set of rigid transformations. Each branch is split into several *segment*s. Each segment is assumed to be rigid and can only rotate around the *joint* which links itself either to its parent branch or to its predecessor. Each joint defines a local frame in which the z axis directs to its child segment. We animate the branches by manipulating the local frames of each joint. The damping angular spring model is adopted to compute the

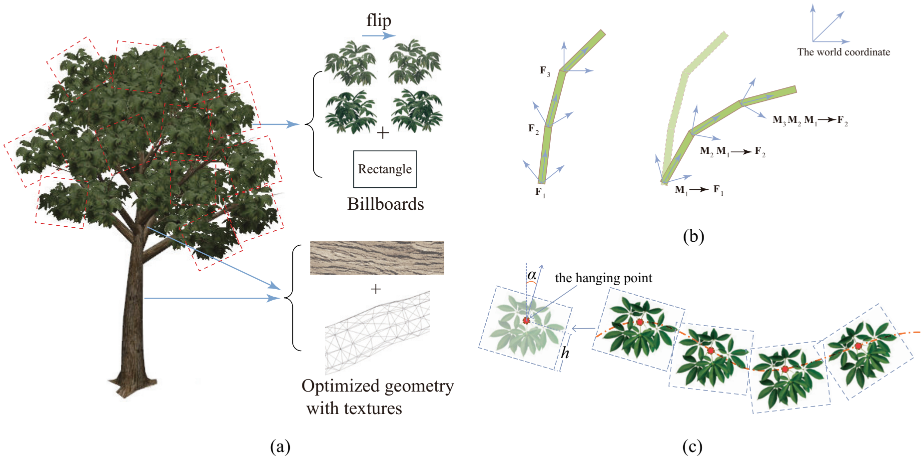

Fig. 1. Quasi-physically based animation of trees: (a) the hybrid representation of trees; (b) branch animation; (c) animation of leaf clusters.

transformation matrices. Figure 1(b) shows the distortion of a branch with three segments during the animation.

The movement of the leaves are primarily driven by that of the host branches. Furthermore, the leaves may vibrate due to the blowing of the wind. We simply model this type of motion as a periodically clockwise swaying, i.e., each leaf swings around its hanging point. Both the frequency and the swing are dependent on the wind strength, which are estimated with simple empirical formulas. The animation scheme is demonstrated in Fig.1(c).

4.3 Level of Details

We introduce three kinds of level of detail representations for the scene geometry, the movement of trees and the wind field. The appropriate level regarding each aspect is determined by multiple criteria, including the distance to the viewpoint, the visual importance and the priority of the image quality.

Geometry LOD We build the LODs for branches and leaf clusters in a separate way. Geometry simplification of branches is accomplished by dropping relatively smaller branches. The resultant variance of the overall shapes can hardly be noticed in practice since small branches are usually covered by the leaves and hence attract little attentions.

Geometry simplification for leaf clusters are performed by consolidating neighboring leaves, which yields a set of reduced leaves. In order to keep the fullness of the tree, we enlarge the remaining leaf clusters by an adjustable ratio. For the sake of performance we use the same leaf textures for lower level models. Smooth transition of consecutive LODs is obtained using an image-space blending technique.

Animation LOD Since the main cost to compute the tree movement lies in updating the local frames, we simplify the computation by deactivating related joints, whose local frames are not recomputed during the animation. Deactivating a joint means to omit the local distortions caused by its rotation.

We assign a priority to each joint, which is estimated by the total mass of the branches it controls. When an active joint is to be deactivated or vice versa, smooth transitions are performed by blending its transformation matrix with identity matrix.

Wind-Field LOD We generate the wind fields in a mipmap fashion. For trees far from the view point, we adopt the coarse version, with which lots of trees suffer from the same average wind force. This is amenable for instance-based simulation, which is popular in vegetation rendering. For all instances of one tree suffering from a same wind force, they share a same distortion. When zooming in or zooming out, smooth transitions are accomplished by interpolating corresponding transformation matrices.

5 Experimental Results

We have implemented our approach with OpenGL 1.5. All experiments were performed on a PC with AMD AthlonXP 3000+ CPU (1.8 GHZ), 512 MB RAM and NVidia 6600 GT video card.

5.1 Calculation of the Wind Velocity

In our experiments, we employed the *Blunt models* proposed in[14] to calculate the wind velocity. The coefficients for the wind power spectrum density are set to: $\alpha = 1, \beta = 5/3, \gamma = 1$, the other three coefficients for different fluctuating components are shown in Table 1. The values of exponential decay coefficients in coherency functions are listed in Table 2.

Table 1. Coefficients for power spectrum density of three components

A_u	B_u	A_v	B_v	A_w	B_w
252.625	60.62	53.76	20.16	5.125	4.92

Table 2. Exponential decay coefficients for coherency functions

C_{xu}	C_{yu}	C_{zu}	C_{xv}	C_{yv}	C_{zv}	C_{xw}	C_{yw}	C_{zw}
3.0	19.7	9.5	6.0	12.0	7.0	0.5	7.5	3.7

We divide the scenes into 100 × 100 cells and compute the wind velocity vector at each cell. The time for calculating the array of velocity vectors for one frame is $50ms$. It costs 144 MB memory space to store the data of one wind velocity field for 1000 frames. Normally, we need pre-compute the wind velocity fields for 1000∼2000 frames and save them as three dimensional arrays.

5.2 Rendering of Forest Scenes

We have tested four forest scenes (denoted by A, B, C and D)with different scene complexities. The snapshots for C and D are shown in Fig.2. Under given wind condition, these scenes can be displayed interactively with physically plausible behaviors. Table 3 lists the performance statistics for the four scenes. The second row reports the number of trees. The frame rates achieved by disabling one of the three LOD techniques are reported in the third, fourth and fifth rows. By integrating all these techniques, our adaptive rendering scheme achieves up to 1000% performance improvement, as shown in the last row.

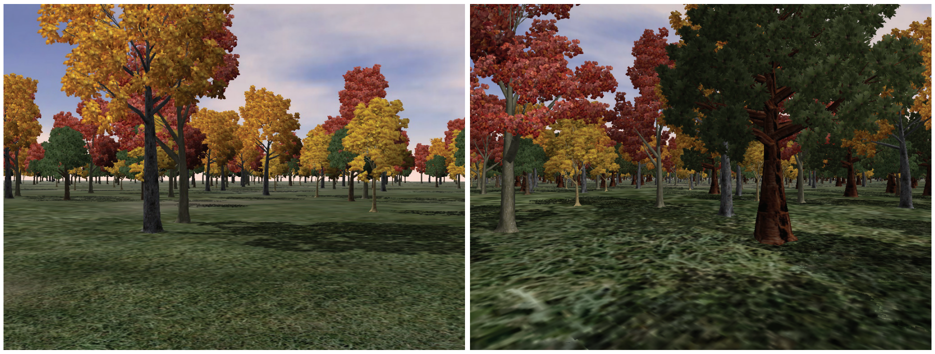

Fig. 2. The snapshots of one forest scene at the image resolution of 1024 × 768.

Table 3. The rendering performance in FPS. The image resolution is 1024 × 768.

Scene	A	B	C	D
#Tree	1000	3000	10000	50000
Without Geometry LOD	15.3	9.5	3.9	1.0
Without Animation LOD	14.9	11.6	7.4	2.1
Without Wind-Field LOD	7.8	3.4	1.1	0.2
With all LODs	20.0	14.5	8.3	2.2

6 Conclusions and Future Work

In this paper we present a quasi-physically based approach for animating large-scale forest scenes under various wind conditions. Our approach achieves interactive frame rates when simulating dynamic forest scenes with tens of thousands of trees.

In the future, we would like to take the mutual influence of trees into consideration, such as attenuation and reduction of wind forces. In addition, we want to study more accurate (efficient) animation schemes for close (distant) trees.

Acknowledgments

This work is partially supported by 973 program of China (No.2002CB312100), NSF of China for Innovative Research Groups (No.60021201), Zhejiang Provincial Natural Science Fund (No.Y105269), National Natural Science Fund of China (No.60303028) and Zhejiang Provincial Natural Science Special Fund for Youth Scientists' Cultivation (No.R603046).

References

1. P.Prusinkiewicz, A.Lindenmayer: The Algorithmic Beauty of Plants. Springer-Verlag, New York (1990)
2. M.Aono, T.Kunii: Botanical tree image generation. IEEE Computer Graphics and Application **4**(5) (1984) 10–34
3. P.E.Oppenheimer: Real time design and animation of fractal plants and tree. Computer Graphics (Proc. of ACM SIGGRAPH'1986) **20**(4) (1986) 55–64
4. J.Weber, J.Penn: Creation and rendering of realistic tree. In: Proceedings of ACM SIGGRAPH 1995, Los Angeles, CA (1995) 119–128
5. Behrendt, S., Colditz, C., Franzke, O., Kopf, J., Deussen, O.: Realistic real-time rendering of landscapes using billboard clouds. In: Proceedings of Eurographics 2005, Ireland (2005) 507–516
6. H.Ono: Practical experience in the physical animation and destruction of trees. In: Eurographics Workshop on Animation and Simulation 1997, Budapest, Hungary (1997) 149–159
7. T.Sakaguchi, J.Ohya: Modeling and animation of botanical tree for interactive virtual environment. In: Proceedings of ACM Symposium on Virtual Reality Software and Technology (ACM/VRST99). (1999) 139–146
8. Ota, S., Tamura, M., Fujimoto, T., Muraoka, K., Chiba, N.: A hybrid method for real-time animation of trees swaying in wind fields. The Visual Computer **20**(10) (2004) 613–623
9. M.Shinya, A.Fournier: Stochastic motion-motion under the influence of wind. In: Proceedings of Eurographics 1992. (1992) 119–128
10. J.Stam: Simulating the effects of turbulence on flexible structures. In: Proceedings of Eurographics 1997. (1997) 159–164
11. T.D.Giacomo, S.Capo, F.Faure: An interactive forest. In: Eurographics Workshop on Animation and Simulation 2001. (2001) 65–74
12. Interactive Data Visualization Inc.: SpeedTree product homepage. (2004)
13. E.Simiu, R.H.Scanlan: Wind Effects on Structures(2nd Edition). Wiley, New York (1985)
14. H.W.Tieleman: Universality of velocity spectra. Journal of Wind Engineering and Industrial Aerodynamics **56** (1995) 55–69
15. G.Solari, G.Piccardo: Probabilistic 3-d turbulence modeling for gust buffeting of structures. Probabilistic Engineering Mechanics **16** (2001) 73–86