

# Physics-based algorithm to simulate tree dynamics under wind load

Lifeng Xu<sup>1</sup>, Zhongzhu Yang<sup>1</sup>, Weilong Ding<sup>1</sup>, Gerhard Buck-Sorlin<sup>2\*</sup>

(1. College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310026, China;

2. IRHS, INRAE, AGROCAMPUS OUEST, Universit d'Angers, SFR 4207 QUASAV, 42 rue Georges Morel, 49071 Beaucouz cedex, France)

**Abstract:** Rapid development in the different computer science fields during the recent decades has facilitated the creation of new applications in the area of dynamic simulation of plant development. Among these new applications, simulation of trees swaying in the wind is of great importance, as those computer graphics related areas, e.g., computer games, tree cultivation and forest management simulations, help a lot in revealing the mechanisms of tree dynamics under wind load. However, it is a big challenge to balance the effect of visualization in real time and calculation efficiency for any simulation algorithm. A physics-based algorithm to simulate tree dynamics under wind load was proposed in this study. A mechanistic model simulating the bending of a cantilever beam was used within the algorithm to simulate deformation of stems, and the algorithm was integrated with a landscape model in which different types of trees were constructed with an L-system-based formalism. Simulation results show that realistic dynamic effects can be achieved with reasonably high computational efficiency.

**Keywords:** physics-based algorithm, tree model, L-system, Cantilever beam

**DOI:** 10.25165/j.ijabe.20201302.4967

**Citation:** Xu L F, Yang Z Z, Ding W L, Buck-Sorlin G. Physics-based algorithm to simulate tree dynamics under wind load. Int J Agric & Biol Eng, 2020; 13(2): 26–32.

## 1 Introduction

Trees are an essential component of the natural environment, and virtual tree models have been widely used in many aspects such as digital games, movies, urban modeling, plant biology, forestry, and agriculture. Given such complex structures, and the large amounts of data coming along with them, it is difficult to construct a model with instant rendering and interaction between trees and the environment. With the continuous development of virtual reality theory and technology, more and more researchers are focusing on the dynamic simulation of trees in the wind<sup>[1]</sup>. As a non-linear flexible system, it is relatively difficult to simulate the deformation of a tree in the wind accurately and quickly, thus making it a challenging topic in computer graphics. Previous methods<sup>[2]</sup> can only be used within relatively simple tree models, in which the control nodes and shape fitting patches are relatively small, mostly counted in hundreds of control nodes and thousands of shape fitting patches. Obviously, the use of such a simple model to generate the natural sway of complex trees is difficult; the dynamic deformation of the calculation accuracy and imaging of the realism will be less precise.

Virtual plant technology uses computers to simulate the growth process of plants, and represents the morphological structure of plants with a 3D model. Virtual plants can help us to research plants with a new perspective. It can not only reveal the complex morphological structure and life characteristics of plants but also promote the development of computer modeling and animation

theory. Virtual plant modeling approaches include two major categories, i.e. graphical-based approaches<sup>[5]</sup> and image-based approaches<sup>[6]</sup>. Aono and Kuniti<sup>[7]</sup> made a detailed estimate of the distribution of the stem and the leaves on the parent by using the tree as biological object model, and developed an interactive synthetic tree operating system called A-system. Oppenheimer<sup>[8]</sup> used fractals to construct a tree model, adjusting the parameters to change the geometric and topological shapes of the simulated object, considering the random effects of the environment to avoid the self-similarity of the fractal, resulting in greater diversity and realism. In order to solve the problems inherent to plant image processing, Shlyakher et al.<sup>[9]</sup> used image segmentation and image-based three-dimensional reconstruction techniques to produce the skeleton (trunk and branch) of the plant and L systems<sup>[10]</sup> to generate the 3D model of the plant. Based on image rendering technology (IBR, Image-Based Rendering), the rapid visual modeling of plant images was well realized. However, the reconstruction of skeletons mainly depended on the approximate 3D shape of the plant, and the result was different from the actual skeletons of the plants. Okabe et al.<sup>[11,12]</sup> designed and implemented a 3D model system using hand-drawn sketches and instance-based editing operations that could be used to generate 3D tree graphics quickly and easily based on the 2D tree sketches, which were also manually drawn by the user.

A number of authors have been working on the simulation of swaying of trees. Stam et al.<sup>[13]</sup> obtained the turbulence effect by filtering the white noise in the frequency domain space, and then used the finite-element model analysis method to calculate the natural frequency of the branches, to obtain the vibration effect of the branches under wind load. This method simulated various types of trees, but using the white noise filtering simulation model made it difficult to represent the natural wind characteristics faithfully. Shin et al.<sup>[14]</sup> simulated the wind field by using the 1/ff noise function. The spring model was used to simulate branch motion, while the leaves were regarded as rigid bodies. The motion was the vector sum of translation and rotation. With this

**Received date:** 2019-02-03 **Accepted date:** 2020-03-02

**Biographies:** Lifeng Xu, PhD, research interests: virtual plant modeling. Email: lfxu@zjut.edu.cn; Zhongzhu Yang, Master Student, research interests: plant modeling and virtual simulation, Email: zhongzhu\_yang@126.com; Weilong Ding, PhD, research interests: virtual plant modeling, Email: wlding@zjut.edu.cn.

\***Corresponding author:** Gerhard Buck-Sorlin, PhD, research interests: plant modeling. IRHS, INRAE, AGROCAMPUS OUEST, Universit d'Angers, SFR 4207 QUASAV, 42 rue Georges Morel, 49071 Beaucouz cedex, France. Tel: +33(0)241225548, Email: gerhard.buck-sorlin@agrocampus-ouest.fr.

method, the wind field was simulated relatively faithfully, but there was not enough space for the spring model to move with the rigid body, rendering the simulated effect far from elaborated. Akagi et al.<sup>[15]</sup> developed a wind model using the Navier-Stokes equation, and used the “boundary condition map” method to analyze the occlusion of the foliage. Hu et al.<sup>[16]</sup> presented an interactive approach for the modelling and animation of outdoor trees from photographs and videos. The fast Fourier transform was used in that study to estimate physical parameters of branches and leaves, which were then applied to a simplified physics-based model to generate animations of trees. In the simulation of a large-scale scene, Fan et al.<sup>[17]</sup> used empirical equations from meteorology instead of physical equations to accelerate the computing process. Different kinds of trees could be simulated by changing coefficients in the equations. Ancelin et al.<sup>[18]</sup> introduced a model of wind damage using the Transfer Matrix Method of beams. This model could be used to explore how damage is related to wind speed. For the uncertainty about the appropriate exponent and reliability of the conventional form, Cullen<sup>[19]</sup> reported on modeling of aerodynamic drag equation forms and concluded that conventional form could be appropriate for trees. Gaffrey et al.<sup>[20,21]</sup> developed a complete 3D model to simulate the elastic mechanical behavior of trees under wind and gravity. The stress and strain distribution at the surface of stem could be estimated using this model. Sellier et al.<sup>[22]</sup> presented a dynamic model to analyze the mechanical response of trees in turbulence. The 3D description of tree architecture was integrated in this finite-element model.

In this study, we proposed a new physics-based approach for calculating deformation of trees subject to different ranges of wind loads. Furthermore, a method to develop a 3D scene within the simulation was introduced, as well as a modeling method for different types of trees. The deformation algorithms proposed were implemented and realized together with the modeling approach in a landscaping context for validation.

## 2 Materials and methods

### 2.1 Simulation of the scene

The 3D virtual scene contains three modules: a sky model, a terrain model, and a radiation model.

For the sky model, SkyBox<sup>[23]</sup> was used to construct the background module of the 3D virtual scene. Representing the 3D scenes as a cube, SkyBox does not have the ability to bend texture coordinates. Thus the standard UV coordinate mapping method was applied to handle the corresponding texture, together with Cubic texture technology which was applied to simulate the details of the sky. Unlike normal textures, the six sides (i.e. Back, Front, Up, Down, Left, and Right) of the skybox can be tiled into the entire space using cube texture technology. The skybox texture is generally produced by a panorama, making the whole scene look continuous and realistic.

A rectangle was used to represent the ground to improve fusion with the Terrain model into the background. Then texture-mapping technology was used to map the ground texture image to the Terrain model to enhance simulation of the entire 3D scene.

The virtual sun was simulated as a directional light source. The direction of the incident light was determined by the location of the sun, i.e. its altitude and azimuth<sup>[24]</sup>.

### 2.2 Plant modeling

The L-system formalism was used to construct the tree model.

A ternary formula can be used to represent the syntax of an L-system<sup>[25]</sup>, as follows:

$$G = \langle V, \omega, P \rangle$$

where,  $V$  is the alphabet;  $\omega$  is the axiom or start word, i.e. a symbol or string from  $V^+$  (the set of all not empty words that can be generated with the alphabet) which represents the initial state of the tree model;  $P$  represents the set of growth or production rules of trees, as in the following example:

$$\begin{aligned} V &: \{A, B, C\} \\ \omega &: A \\ P_1 &: A \rightarrow BC \\ P_2 &: B \rightarrow AC \\ P_3 &: C \rightarrow B \end{aligned}$$

There are three symbols in the alphabet  $V$ . The object  $A$  is a start word. Three rules ( $P_1$ ,  $P_2$  and  $P_3$ ) are defined in this L-system rule set:  $A$  will be replaced by  $BC$ ,  $B$  be replaced by  $AC$ , and  $C$  be replaced by  $B$ . The rule set will be applied repeatedly and in parallel to a series of strings ( $\omega - \sigma_1 - \sigma_2 - \dots - \sigma_n$ ), which can be further interpreted with Turtle geometry<sup>[26]</sup>, thereby rendering a developmental series of a virtual tree.

#### 2.2.1 Surface modeling

To enhance the realism of the tree model, Bezier surfaces<sup>[27]</sup> were used to simulate the trunk surface. The Bezier surface was defined as:

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n P_{ij} B_{i,m}(u) B_{j,n}(v) \quad u, v \in [0, 1] \quad (1)$$

$$B_{i,m}(u) = C_m^i u^i (1-u)^{m-i} \quad (2)$$

$$B_{j,n}(v) = C_n^j v^j (1-v)^{n-j} \quad (3)$$

where,  $P_{ij}$  is the space point range;  $u$  and  $v$  are the parameters of the Bezier surface. Equations (2) and (3) are the Bernstein basis functions. The space grid (also called feature grid) is formed by connecting the two adjacent points in  $P_{ij}$ . The matrix of the Bezier surface is calculated with Equation (4), as follows:

$$P(u, v) = \begin{bmatrix} B_{0,n}(u) & B_{1,n}(u) & \dots & B_{m,n}(u) \end{bmatrix} \begin{bmatrix} P_{00} & P_{01} & \dots & P_{0m} \\ P_{10} & P_{11} & \dots & P_{1m} \\ \dots & \dots & \dots & \dots \\ P_{n0} & P_{n1} & \dots & P_{nm} \end{bmatrix} \begin{bmatrix} B_{0,m}(v) \\ B_{1,m}(v) \\ \dots \\ B_{n,m}(v) \end{bmatrix} \quad (4)$$

#### 2.2.2 Topology modeling

Ranks were used to specify the branches on an individual tree model (as shown in Figure 1). A frustum of a cone was used as the basic unit for tree branches. The branch unit model was defined with the following structure,

$$B(r_t, r_b, l, r, o) \quad (5)$$

where,  $r_t$  represents the top radius of the branch;  $r_b$  is the bottom radius of the branch;  $l$  indicates the length of the branch;  $o$  denotes the sequence of the branch;  $r$  indicates the position of the branches with the same order.

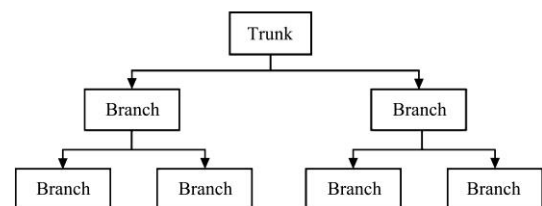


Figure 1 Structure of branches with different ranks designed within the topological models

Since the growth direction of each branch is not necessarily the same from one branch to the next, parameters to control growth

directions for the branches were needed. Three turtle commands, named **RU(a1)**, **RH(a2)**, **RL(a3)** applying geometry interpretation based on Turtle geometry<sup>[28]</sup>, were used to accomplish this task.

### 2.2.3 Branch modeling

The tree model established with L-systems has a modular structure that is recursively generated from simple structures according to the growth and development rules of the tree in question. Taking advantage of this, calculation and realization of deformation for the branches under wind load can be made with low cost by an algorithm representing the relationship between wind forces and branch bending. In this algorithm, a single branch was divided into multiple segments that were connected with several joints. Each segment was considered as a rigid body. Therefore, the bending of the branches can be obtained by calculating the deflection angle of the nodes. A schematic diagram of the branch model is shown in Figure 2.

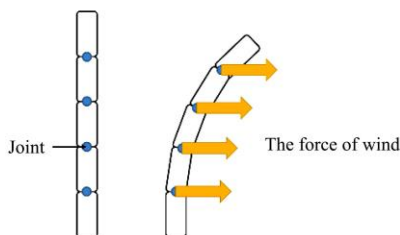


Figure 2 Schematic diagram showing the structure of a branch and its reaction to the wind force in the model. Points refer to joints connecting different segments, while arrows represent the wind force, which has an impact on the joint to bend the branch

### 2.3 Wind

The wind field was simulated as a uniform flow field. Movement of deformable solid with the effect of fluid loads could exert an effect on the uniform flow field to change the distribution of fluid loads. This phenomenon was named fluid-structure interaction<sup>[29]</sup>. Interactions between wind load and trees were considered as a similar phenomenon. Thus, a similar approach<sup>[30]</sup> could be used to simulate the effect on trees from different wind directions and speeds. Two modules, branch and leaf, were designed in this study, both were considered within the calculation of interaction between tree modules and wind load.

The virtual 3D space was divided into many voxels to calculate the influence of tree modules on the wind field. Assuming the entering wind velocity is  $V$ , then it would decay to  $V_i$  after having travelled through  $i$  continuous voxels (Figure 3).

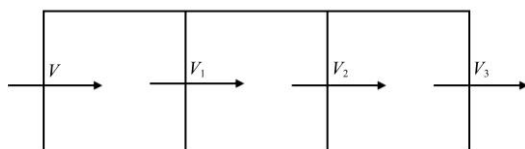
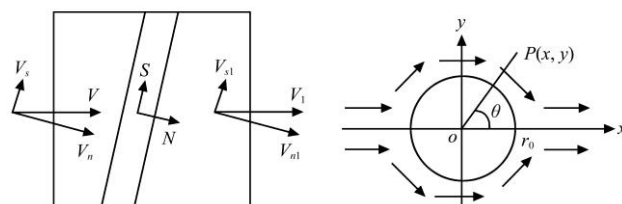


Figure 3 Diagram of squares each represents a voxel. The wind velocity  $V$  decays to  $V_1$  after having crossed the first voxel. As before,  $V_1$  decays to  $V_2$  after having crossed the second voxel

#### 2.3.1 Occlusion of branches

Only one single branch segment was considered per voxel (Figure 4a). The particles of the fluid around a branch, which latter was regarded as a cylinder, were disturbed. The input wind velocity  $V$  of this voxel was decomposed into two components ( $V_s$  being parallel to the direction of that branch,  $V_n$  perpendicular to the direction of that branch). Since  $V_s$  was not obstructed by the branches, the component  $V_{s1}$  of the output wind velocity in this direction would be equal to the component  $V_s$  of the input wind

velocity in this direction. The output wind velocity  $V_1$  could be obtained after calculating the changes of  $V_n$ .



a. Quadrangle in the middle of the voxel as a projection of a branch segment, on which  $S$  represents the direction of the branch segment and  $N$  perpendicular to  $S$   
 b. Circle as the cross section of the stem within the voxel, with the arrows representing the wind velocities around the circle

Figure 4 Diagram of a voxel with branch segment and wind load upon the stem

Calculation of  $V_n$  can be transformed into calculation of a cylinder with circular cross-section (Figure 4b). Ideal fluid around the cylinder is composed of uniform flow and doublet flow. The complex potentials of the uniform flow and doublet flow ( $W_1$  and  $W_2$ , respectively) could be calculated with Equations (6) and (7):

$$\left. \begin{aligned} W_1 &= V_n r (\cos \theta + i \sin \theta) \\ W_2 &= \frac{M}{2\pi r} (\cos \theta + i \sin \theta) \end{aligned} \right\} (r \geq r_0) \quad (6)$$

$$M = 2\pi r_0^2 V_n \quad (7)$$

where,  $M$  is the dipole moment, and  $r$  represents the distance between  $P(x, y)$  and the origin  $O$ . The radius of the cylinder is  $r_0$ .  $\theta$  is the angle between  $OP$  and the  $X$  axis. According to the principle of superposition of potential flows, the complex potential  $W$  of a new flow formed by the superposition of uniform flow and doublet flow could be obtained as follows:

$$W = W_1 + W_2 \quad (8)$$

Then the velocity potential function  $\Phi$  and the stream function  $\psi$  can be calculated as:

$$\left\{ \begin{aligned} \Phi &= V_1 x + \frac{M}{2\pi} \frac{x}{x^2 + y^2} \\ \psi &= V_1 y + \frac{M}{2\pi} \frac{y}{x^2 + y^2} \end{aligned} \right. \quad (9)$$

The velocity components at position  $P$  in the flow field are:

$$\left\{ \begin{aligned} V_x &= \frac{\partial \Phi}{\partial x} \\ V_y &= \frac{\partial \Phi}{\partial y} \end{aligned} \right. \quad (10)$$

$V_{n1}$  is obtained from Equation (10) and then combined with  $V_{s1}$  to get the output wind velocity  $V_1$ .

#### 2.3.2 Occlusion of leaves

Compared to the rigidity of the branches, the rigidity of leaves is low and the variation in the wind is rather complicated. In this study, a simplified method was used to approximately simulate the effect of the blade on the wind.

As shown in Figure 5, the two-dimensional projection of the blade in the direction of the wind was calculated first; then the input wind velocity  $V$  of the voxel interacts with the blade. The output wind velocity  $V_1$  was calculated as follows:

$$V_1 = \left( 1 - \sin \theta \frac{S_l}{S_v} \right) V + k \sin \theta \frac{S_l}{S_v} V, \quad 0 < k < 1 \quad (11)$$

$S_l$  is the area of the leaf in the voxel in question.  $S_v$  represents the projected area of the voxel in the direction of the wind velocity.

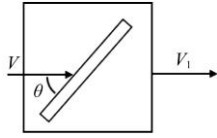


Figure 5 Schematic diagram of occlusion of leaves. The rectangle in the middle represents the projection of a leaf.  $\theta$  is the angle between the direction of leaf projection and input wind velocity

### 2.3.3 Application

Depending on the accuracy of voxel partitioning, the number of branches and leaves in a voxel were different. In actual calculation, there might be multiple branches and leaves in one voxel. The branches were discretized (Figure 6), i.e. each branch segment was considered as a node and then could be analyzed using the above-mentioned circular cylinder model. Similarly, the leaves were also discretized into nodes. Assuming that the number of branch segments is  $m$  and the number of leaves is  $n$ , the output wind velocity  $V_1$  will be:

$$V_1 = \left( 1 - \sum_{i=0}^n \sin \theta_i \frac{S_{li}}{S_v} - \sum_{i=0}^m \frac{S_{bi}}{S_v} \right) V + k \sum_{i=0}^n \sin \theta_i \frac{S_{li}}{S_v} V + \sum_{i=0}^m \frac{S_{bi}}{S_v} V' \quad (12)$$

where,  $S_b$  is the projected area of branch segment in the direction of wind velocity.  $V'$  represents the average wind velocity.

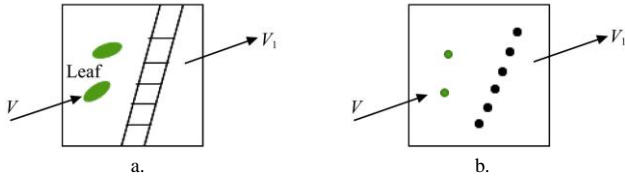


Figure 6 (a) Branch segments and leaves within a voxel; (b) Discretized nodes within a voxel, a green node indicates the leaf while a black node indicates the branch

### 2.4 Dynamic simulation

Deformations can be observed on branches when the wind blows through the trees. In doing so, the deflection angle was found to be the key parameter during simulation of the processes.

Factors affecting the deflection angle of the branches were mainly of three aspects, the wind velocity, branch diameter, and branch rigidity. The rigidity of the branch is in the Young's modulus: the higher it is, the less deformation the branch would be subjected to. As a complex system, the values of Young's modulus for different tree parts vary. To simplify it in the simulation, the branches of a tree were divided into three parts: trunk, branch, and twig. The range of values of Young's modulus for the parts are given in Table 1 (according to US Department of Agriculture<sup>[31]</sup>).

**Table 1 The value range of Young's modulus for a tree<sup>[31]</sup>**

Different parts of the tree	Young's modulus value/GPa
trunk	5-12
branch	1-5
twig	0.5-1

#### 2.4.1 Force analysis of branches

As shown in Figure 7, the branch segment was connected with its father branch at point A, and there might be a child branch at the end B. The branch segment was affected by the wind force  $F_1$  and the force  $F_2$  exerted by the rear segments at the end B if there was a child branch attached on end B. The point of action of the wind force  $F_1$  is at the center of mass of the branch segment. A moment of couple  $m$  is attached to the branch according to the

force translation theorem to move  $F_2$  to O.

$$m = F_2 \times L_{BO} \quad (13)$$

where,  $L_{BO}$  is the center of mass of the branch segment relative to the end point B of the branch segment. The relationship between wind velocity and wind force can be calculated with Equation (14).

$$\left. \begin{aligned} F &= F_d = F_l \\ F_d &= \frac{1}{2} C_d \rho V^2 S \\ F_l &= \frac{1}{2} C_l \rho V^2 S \end{aligned} \right\} \quad (14)$$

where,  $C_d$  represents the resistance coefficient;  $C_l$ , the lift coefficient;  $\rho$ , the air density;  $V$ , the velocity of wind;  $F_d$ , the resistance of the branch, and  $F_l$  the drag force of the branch.

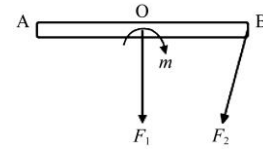


Figure 7 Force analysis for a branch segment. The force  $F_2$  was exerted by the rear segments at the B end in the case of a child branch attached to B

#### 2.4.2 Calculation of deformation

The end of a branch which attached to its father branch was considered as fixed on the upper branch and the other end was free to move. Thus, deformation of the cantilever beam from the mechanics of materials can be applied on this matter. Since uniformly distributed loads were acting upon the beam, the bending moment on the full length of the beam could be described by a function. As shown in Figure 8, it was cut off from any cross section, with a uniformly distributed load permanently present on the left side of the section, and a reactive force  $F_{RO}$  on the fulcrum generated by the wind load  $q$ , but only a uniformly distributed load existed on the right side of the section. The balance of all the forces on the right part was used to calculate the bending moment, as follows:

$$M(x) = -\frac{1}{2} q(L-x)^2 \quad (15)$$

where,  $q$  is uniformly distributed load;  $L$  is the length of the beam;  $M$  represents the bending moment. The deflection differential equation is:

$$EI \frac{d^2 W(x)}{dx^2} = -M(x) \quad (16)$$

where,  $E$  represents the Young's modulus of the branch, and  $I$  is the area moment of inertia. Equation (15) and Equation (16) are integrated in the following equation:

$$EI \frac{dW(x)}{dx} = -\frac{1}{6} q(L-x)^3 + C \quad (17)$$

where,  $C$  is an arbitrary constant. The constraints at the fixed end are as follows:

$$\left. \frac{dW(x)}{dx} \right|_{x=0} = 0 \quad (18)$$

Then  $C$  is calculated as Equation (19):

$$C = \frac{qL^3}{6} \quad (19)$$

Combining (17) with (19), the following equation can be obtained:

$$\frac{dW(x)}{dx} = \frac{q}{6EI} [L^3 - (L-x)^3] \quad (20)$$

Finally, the deflection angle can be calculated as:

$$\theta = \left. \frac{dW(x)}{dx} \right|_{x=L} \quad (21)$$

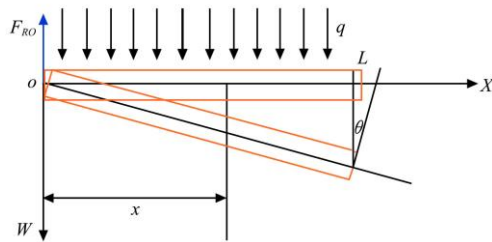


Figure 8 Diagram showing the stress analysis of cantilever beam. Uniformly distributed loads  $q$  are applied to the beam.  $F_{RO}$  is a reactive force on the fulcrum generated by the wind load  $q$

### 3 Simulation Results

Using the method described above, a dynamic simulation system with 3D tree model was implemented in the Eclipse development environment using Java programming language and JOGL graphics engine. The computer hardware used in the simulation was configured with Intel (R) Core (TM) i7-6700HQ CPU, 8GB memory, and NVIDIA GeForce GTX 970M graphics card.

Since the entire 3D space was divided into multiple voxels, the division accuracy would have a direct influence on the simulation results. With higher division accuracy or more voxels designed, the results would be more accurate, but also be more time consuming. Therefore, it was necessary to find a division model with a relatively high computational efficiency and which at the same time would meet the requirement of simulation accuracy. For a tree model, the total number of discretization nodes was set to a fixed value, which meant the number of nodes within a certain voxel would be different when the total voxel number was changed. For a tree model with 23,684 triangles, Table 2 shows the relationship between the division accuracy and the calculation time for each frame. It can be clearly observed that the calculation took much longer when the space was divided into more voxels, among which the  $80 \times 80 \times 80$  voxels took the longest time to calculate, while the  $40 \times 40 \times 40$  only took about one third of the former, which, compared to those lower division settings, meant that it gained enough accuracy but with acceptable calculation time.

**Table 2 Calculation time cost within the algorithm under different division accuracies**

Division accuracy	Calculation time/ms
$5 \times 5 \times 5$	5.6253
$10 \times 10 \times 10$	8.3641
$20 \times 20 \times 20$	13.8216
$40 \times 40 \times 40$	34.2537
$80 \times 80 \times 80$	113.2165

The simulation results were verified on three aspects: 1) movement of a single branch under different wind velocities; 2) movement of the same tree under different wind velocities; 3) movement of different trees under the same wind velocity.

A branch from *Koelreuteria paniculata* was used for simulation to demonstrate the results of the algorithm described for a single branch, of which the bending was affected with four settings of wind velocity values, i.e. 0 m/s, 4 m/s, 8 m/s, and 12 m/s. The Young's modulus of the branch is 3.4 GPa, and the maximum curvatures of the branch bending under different wind velocities are shown in Figure 9.

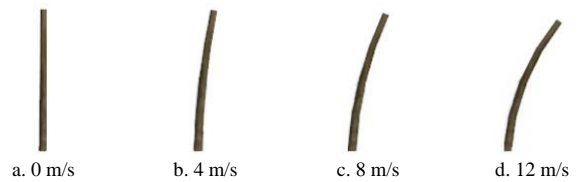


Figure 9 Maximum curvatures of the branch (*Koelreuteria paniculata*) bending under different wind velocities

The dynamics of two tree species (*Koelreuteria paniculata* and *Ulmus pumila*) under four different wind loads, i.e. 0 m/s, 4 m/s, 8 m/s and 12 m/s, were simulated as the second step. The maximum curvatures were captured and compared (as shown in Figure 10). Different dynamics of bending and swaying of the branches between two trees and among different wind velocities on the same tree could be observed. Physical properties of the two trees are different, including tree architecture, Young's modulus, etc. This is the main reason why the two trees reacted differently under the same wind load. Concerning the same tree under different wind velocities, it was clear that it became more bent as the wind velocity increased. Finally, the method proposed by Akagi and Kitajima<sup>[15]</sup> was applied to the same tree models, as shown in Figure 11, and the result shows that there was no significant difference between dynamic reactions from their method and the algorithm proposed in this study.

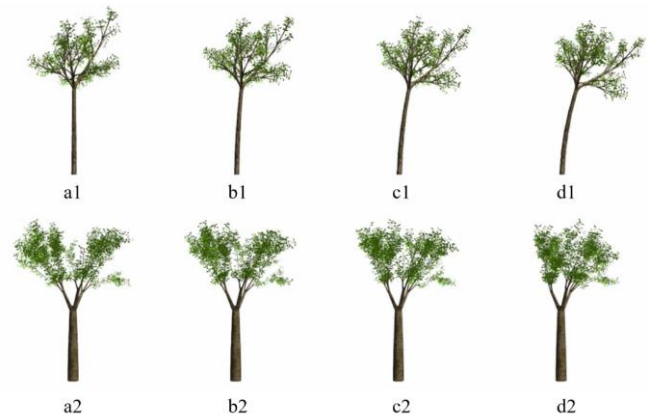


Figure 10 Snapshots of the maximum bending state of the two tree models, *Koelreuteria paniculata* (a1-d1) and *Ulmus pumila* (a2-d2), under four different wind loads with velocities of: a1, a2: 0 m/s; b1, b2: 4 m/s; c1, c2: 8 m/s; d1, d2: 12 m/s, with the physics-based algorithm proposed in this study

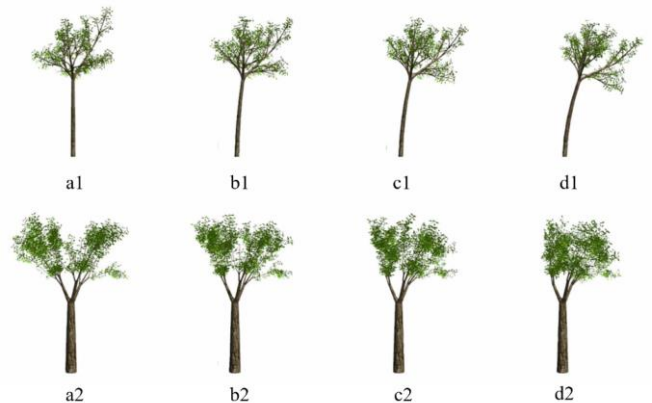


Figure 11 Snapshots of the maximum bending state of the two tree models, *Koelreuteria paniculata* (a1-d1) and *Ulmus pumila* (a2-d2), under four different wind loads with algorithm discussed in the paper<sup>[15]</sup>. Velocity of wind: a1, a2: 0 m/s; b1, b2: 4 m/s; c1, c2: 8 m/s; d1, d2: 12 m/s



## 4 Discussion

A physics-based algorithm to simulate tree dynamics of two different tree species under various wind loads was proposed in this study. By considering the physical properties of the trees, i.e. branches and leaves, the wind field and forces on trees can be simulated. Leaf shedding under high wind load has not yet been considered in our model, which should be considered in the simulation in further study.

One of the applications of the algorithm lies in tree cultivation and forest industry. In the forest, the strong wind can affect the growth process of the trees, resulting in structural deformation of the trees, even affecting the structural layout of the forests and the tree species diversity<sup>[32]</sup>. In urban areas, trees on both sides of a road are always at risk to be ruptured by strong winds, often causing harm and damage to people, vehicles or houses<sup>[33]</sup>, all this representing substantial financial risks for insurance companies. In a forestry context, the simulation algorithm can be used to predict storm damage or to optimize planting density and row orientation in a stand in certain geographical locations, and it can also be used to simulate and quantify the financial risks for urban landscaping. Currently, only the morphological effects of the trees under wind load were considered in the algorithm, while the impacts on physiological processes of trees were not. It is also essential to combine the effects of the wind load on biomechanics of the trees in order to better simulate its dynamics. We will do this in the near future.

Another application of our model could be in the context of ecological modelling, more particularly bioclimatology and photosynthesis models: it is well known that the rate of photosynthesis depends on the concentration of CO<sub>2</sub> inside the leaf, which in turn depends on stomatal conductance<sup>[34]</sup>. The latter process is controlled by a number of plant and biophysical processes, amongst which features wind velocity. Detailed information (at a fine spatial resolution) about the speed and direction of air movement around the leaf would greatly improve estimations of the CO<sub>2</sub> concentration and relative humidity in the leaf boundary layer, which latter serves as the base for estimating the amount of CO<sub>2</sub> available for photosynthesis, and which currently cannot be very well quantified.

## 5 Conclusions

A method for simulating the movement of trees in the wind based on physics was presented in the study. The L-system formalism was used to construct a tree model and a whole branch was divided into multiple segments. Considering the wind load acting on each branch, we calculated the deflection of each branch of the tree in the wind, and then superimposed the deflection of all the small segments to obtain the movement of a whole branch. The experimental results show that the method is more realistic and greatly reduces the computation time.

It is true that L-systems are not necessarily the only method to simulate trees with static architecture exposed to wind loads, as other methods of architecture reconstruction would be more efficient in some way. L-systems were used in this study, as it is only the first step to consider the static structure of the simulated trees. Our next step will include the effect of wind load on growth of a tree, and eventually consider the influence on the whole growth period and not only on structure, but also on functional physiological processes.

Admittedly, distribution of the wind field in a 3D space is relatively complicated, not to mention movement of a tree in the wind. To simplify the question, we assumed that the power of the wind within the space stayed the same, in an attempt to control calculation cost, however, simulation results were also less convincing. In order to get more realistic results, analysis of spatial distribution of wind field will be added in the future.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 31301230, 61571400, 31471416); and the Scientific Research Fund of Zhejiang Provincial Education Department (Grant No. Y201224867). The authors are grateful to the anonymous reviewers whose comments helped to improve this paper.

## [References]

- [1] Ancelin P, Courbaud B, Fourcaud T. Development of an individual tree-based mechanical model to predict wind damage within forest stands. *Forest Ecology Management*, 2004; 203(1-3): 101–121.
- [2] Sakaguchi T, Ohya J. Modeling and animation of botanical trees for interactive virtual environments. *ACM Symposium on Virtual Reality Software and Technology*, London, United Kingdom, 1999; pp. 139–146.
- [3] Ono H. Practical experience in the physical animation and destruction of trees. *Eurographics Workshop on Computer Animation and Simulation*, Vienna, Austria, 1997; pp. 149–159.
- [4] Giacomo T, Capo S, Faure F. An interactive forest. *Eurographics Workshop on Computer Animation and Simulation*, Vienna, Austria, 2001; pp. 65–74.
- [5] Cannon J, Mandelbrot B. The fractal geometry of nature. *The American Mathematical Monthly*, 1984; 91(9): 594.
- [6] Teng C, Chen Y. Image-based tree modeling from a few images with very narrow viewing range. *The Visual Computer*, 2009; 25(4): 297–307.
- [7] Aono M, Kunii T. Botanical tree image generation. *IEEE Computer Graphics and Applications*, 1984; 4(5): 10–34.
- [8] Oppenheimer, Peter E. Real time design and animation of fractal plants and trees. *ACM SIGGRAPH Computer Graphics*, 1986; 20(4): 55–64.
- [9] Shlyakhter I, Rozenoer M, Dorsey J, Teller S. Reconstructing 3D tree models from instrumented photographs. *IEEE Computer Graphics and Applications*, 2001; 21(3): 53–61.
- [10] Lindenmayer A. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology*, 1968; 18(3): 280–299.
- [11] Okabe M, Owada S, Igarashi T. Interactive design of botanical trees using freehand sketches and example-based editing. *Computer Graphic Forum*, 2005; 24(3): 487–496.
- [12] Ijiri T, Owada S, Okabe M, Igarashi T. Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints. *ACM Transactions on Graphics*, 2005; 24(3): 720–726.
- [13] Jos S. Stochastic Dynamics: Simulating the effects of turbulence on flexible structures. *Computer Graphics Forum*, 2010; 16(3): 159–164.
- [14] Ota S, Tamura M, Fujita K. A hybrid method for real-time animation of trees swaying in wind fields. *The Visual Computer*, 2004; 20(10): 613–623.
- [15] Akagi Y, Kitajima K. Computer animation of swaying trees based on physical simulation. *Computers and Graphics*, 2006; 30(4):529–539.
- [16] Hu S, Zhang Z, Xie H, Igarashi T. Data-driven modeling and animation of outdoor trees through interactive approach. *Visual Computer*, 2017; 33(6-8): 1017–1027.
- [17] Fan J, Xiao S. The study of real-time animation of forest scene in wind projection. *ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry*, Kobe, Japan, 2015; pp. 101–104.
- [18] Ancelin P, Courbaud B, Fourcaud T. A population approach to study forest stand stability to wind: individual tree-based mechanical modeling. *International Conference Wind Effects on Trees*, University of Karlsruhe, Germany, 2013.
- [19] Cullen S. Trees and wind: A practical consideration of the drag equation velocity exponent for urban tree risk management. *Journal of Arboriculture*, 2005; 31(3): 101–113.

- [20] Gaffrey D, Kniemeyer O. The elasto-mechanical behaviour of Douglas fir, its sensitivity to tree-specific properties, wind and snow loads, and implications for stability - a simulation study. *Journal of Forest Science*, 2002; 48(2): 49–69.
- [21] Gaffrey D, Sloboda B. Modifying the elastomechanics of the stem and the crown needle mass distribution to affect the diameter increment distribution: A field experiment on 20-year old *Abies grandis* trees. *Journal of Forest Science*, 2004; 50(5): 199–210.
- [22] Sellier D, Brunet Y, Fourcaud T. A numerical model of tree aerodynamic response to a turbulent airflow. *Forestry*, 2008; 81(3): 279–297.
- [23] Ping S, Tao Y. Research and realization of modeling method for virtual geographic scenes. *International Conference on Information Science and Engineering*, 2011; pp. 2297–2299.
- [24] Kolivand H, Rhalibi A, Sunar M, Saba T. ReVitAge: Realistic virtual heritage taking shadows and sky illumination into account. *Journal of Cultural Heritage*, 2018; 32: 166-175.
- [25] Prusinkiewicz P, Lindenmayer A. *The algorithmic beauty of plants*. New York: Springer-Verlag, 1990.
- [26] Abelson H, Disessa A. *Turtle geometry*. MIT Press, Cambridge, Mass.-London, 1980.
- [27] Hu S, Zuo Z, Sun J. Approximate degree reduction of triangular bezier surfaces. *Tsinghua Science and Technology*, 1998; 3(2): 55–58.
- [28] Prusinkiewicz P. Graphical applications of L-systems. *Proceedings - Graphics Interface*, 1986; pp. 247–253.
- [29] Pei J, Yuan S, Yuan J. Fluid-structure coupling effect on periodically transient flow of a single-blade sewage centrifugal pump. *Journal of Mechanical Science and Technology*, 2013; 27(7): 2015–2023.
- [30] Li F. Realistic Simulation of three-dimensional trees swaying in the wind. *Computer Science*, 2012; 39(11): 254–260.
- [31] Concepts R. *Wood handbook - wood as an engineering material*. Agriculture handbook / United States. Dept. of Agriculture (USA). No. 72. 2013, 1.
- [32] Dupont S, Pivato D, Brunet Y. Wind damage propagation in forests. *Agricultural and Forest Meteorology*, 2015; 214-215(3): 243–251.
- [33] Kenneth J, Nicholas H, Peterk A. Mechanical stability of trees under dynamic loads. *American Journal of Botany*, 2006; 93(10): 1522.
- [34] Nikolov N, Massman, W, Schoettle A. Coupling biochemical and biophysical processes at the leaf level: an equilibrium photosynthesis model for leaves of C3 plants. *Ecological Modelling*, 1995; 80(2-3): 205–235.