Physically based Animation of Broad-leaf Plant

Guo Wu, and Li WenHui, and Feng Guanghui

Key Laboratory of Symbolic Computation and Knowledge Engineer of ministry of Education,
College of Computer Science and Technology, Jilin University, Changchun 130012, P.R. China

Summary
This paper describes the modeling and animation of broad-leaf plant moving with the wind. There are many instances where animations of vegetation and foliage are limited to simple textures in non real-time. To address this limitation, we show how to use a physically based animation method to realistically simulate the movement of stem and foliage in real-time. Our emphasis is on the overall geometrical structure of the plant according to botanical principles, including hierarchically transaction of the part of branch and foliage. In particular, by introducing the transfer force, dynamics simulation is performed from the improved semi-rigidity model for foliage to the segment model for branch. By comparing with the other simulation methods, the experiment indicates that the proposed model can animate the movements caused by external forces such as winds both effective and realistic.
Key words:
physically based modeling, real-time, Animation, plant

Introduction

Much of the research involved in computer graphics is focused on eliminating the gap between art and science, as well as creating realistic images and animations world around us. Scientists and artists have a common goal, to realistically display the scene in our real world by their hands. The quest for photorealism challenges modeling and rendering algorithms, while a departure from realism may offer a fresh view of known structures [1]. And nowadays the computer animation is combing science with art.

In computer animation, realistic vision showed on computer is more attractive than physically accuracy. An extensive amount of work such as Growth Model has already been carried out in the area of plant. In general these works include three aspects: modeling the plant itself, animation and rendering of the plant, and interaction of the plant with its surrounding environment. Many computer game developers and scientists interested in visualizing natural environments make some efforts on modeling visual plant and rendering the plant, but not on the animation of plant in real time. Prusinkeiwicz and Lindenmayer built the tree model by using L-Systems method firstly, which is one of the parametric approaches used to make tree models [2,3,4]. Oppenheimer used fractals and the notion of fractal self-similarity for modeling plants in [5]. Weber and Penn also propose one method to construct the forest from one plant [6]. Recently, Wang divided leaves of plant into several segments in order to represent the softness of leaves in quaternion space. However it is difficult to simulate some plants in real time because of the expensive computation [7]. On the other hand, Deussen and Sakaguchi developed and improved the animation rendering technique in the quite complex scenery and interactive environment [8,9]. In the physically based methods, Ono proposed another segment based method in which braches are connected by springs [10].Shinya computed the distortion of tree with the wind using the stochastic theory through calculus equations [11]. And Stam introduced half Lagrange method to obtains the rapids effect through filtering white noise in the frequency field, and then to calculate the auto-oscillatory frequency of branch by using the finite element mode analysis, eventually to gain oscillatory effect of branch in the wind[12]. In order to compute the distortion of tree and to make the tree have good physical condition, Giacomo combined process-based method and physical-based method to realize animation of tree. He computed the time-consuming latter only in the near distance [13]. Feng Jin-hui simulated the form of willow tree in the wind by carrying on the branch classified processing [14]. Liu You-quan proposed that the finite element method and parallel computation on GPU could accelerate distortion computation of the willow branch in the wind in real time [15].

Most research above focus on simulating the movement of plant with thin leaves or small leaves, where each leave is modeled as a single polygon in the wind and they always have the same appearance. So they can not show the real shape of each leave, like blade bend of leaves. Our goal is to simulate the realistic movement of broad-leaf plant, like Japanese banana and violet and hollyhock, with the wind in real time. We propose a method for rendering and animating broad-leaf plant in real-time and realistic manner according to the familiar Newtonian notation. We describe the details of the process of stem and leaves when the wind force added. By introducing the notion of rigidity, it is not only to control the general behavior of the plant efficiently but to reduce the complexity of the process of force analysis. On the other hand, we model the leaves of plant in a new way with an improved semi-rigidity structure, which can greatly simplify the self-collision detection of leaves and expensive computation. The rest of the paper will proceed as follows. In section 2, we will review our physically based model for branches and leaves of broad-leaf plant. Section 3

Manuscript received February 25, 2006.
Manuscript revised February 28, 2006.

will discuss the detail of the system and dynamics of branches and leaves. Section 4 will evaluate of results in comparison with the other system. Finally, in Section 5, we conclude with some discussion on possible future work.

2. Building the Physically based Broad-leaf Plant

For the purpose of simulating the behavior of plant perfectly in the wind, we should model for the plant based on physics. There are several phases for this modeling. First, the stems or branches of plant have been divided into several little cylinders or prism primitives with equal length and different radius, which can generate rotation and translation acted by the internal forces and external forces. Thus it can effectively control the computation time and create animation in real time. Second, we also divided the parallel venation of leaves into several segments because of our relatively broad leaves of plant, which is different from other works. Finally, we adopt of improved semi-rigidity structure to control the points on leaves to make them move with the wind. It is obey the botanic characteristics in nature, and it makes the plant animation vivid and more realistic.

2.1 Creating the Stems or Branches

There is a possible segmentation of stems or branches divided shown in Figure 1. Matrixes were used to represent transformations (Cartesian coordinate system) to prevent Sakaguchi's[9] spherical coordinate system which can give rise to the Gimbal lock that can occur in using Euler Angles. Matrixes are considered a very convenient and simple method to calculate rotations and translations in the three dimensional space. It can also save massive computation time by reducing the essential transformation to the hardware applications, such as GPU [16].

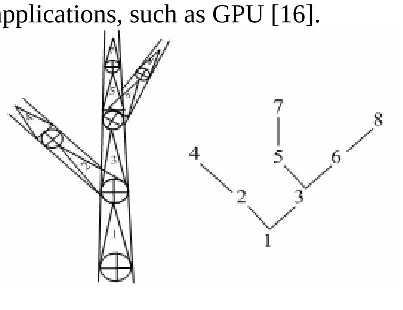


Fig. 1: Segment hierarchy of a plant.

The model on the left in Figure 1 shows a simple segment structure, and the right the segments form a hierarchical tree. Each segment has exactly one parent segment, except for the root segment which has no parent. In turn, each segment can have none, one, or more child segments. As we can see in figure 1, segment 1 is the root segment and therefore has no parent. Segment 5 and segment 6 have one parent segment, 3. For example if some external forces (wind force) cause segment 3 to be rotated to the right, segments 5, 6, 7 and 8 must follow this rotation such that there are no graphs in the model, and the segments stay attached to one another. We will later discuss the detail of its algorithm in Section 3.

2.2 Creating the Broad Leaves

If a displayed equation needs a number, place it flush with the right margin of the column (e.g., see Eq. 1).
For knowing this paper clearly, Figure 2 shows some botanic notions of leaf mentioned in this paper.

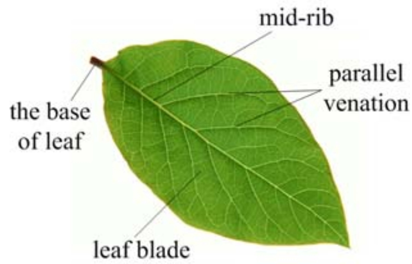
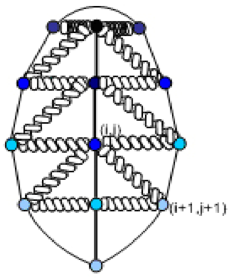


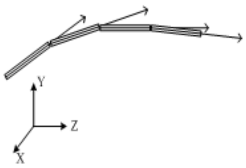
Fig. 2: A leaf of Plant.

In figure 3, we have been modeled the broad leaf blades in a special way — an improved semi-rigidity structure which is inspirited from the structure used in cloth [17], which is represented as an object composed of mass points connected by semi-rigid rods. This kind of mold is more suitable for some ordinary leaf forms of plant, such as oblong, elliptic, rhombic and spoon shape, which have parallel venation in common. It can be seen from the figure that there is one mid-rib in the middle of leaf blade in the longitudinal direction on which the rigidity and mass of points gradually lessen from the base of leaf to the apex. The mass points on the mid-rib have the biggest rigidity and mass than those on both sides in the same latitudinal direction, which is shown by the change of color in the figure. The constraint in the latitudinal direction may guarantee the movement of leaf blades with the wind. And the constraint added on parallel venation may avoid the abnormal phenomenon of tearing and over-bending of leaves in the movement. This improved model is very suitable for showing the form of the broad-leaf plants in wind.
Each mass point on the mid-rib accords with the point in the middle of leaf blade in the model, so the movement of points on the leaf model may lead to the movement of leaf segments connected with those points. Since the mid-rib of leaves is very thin, they can be rendered as cubic tubes, which make similar approximations for efficiency. Figure 3(b) shows a close up view of a leaf mid-rib. The size of the leaf can be modified by the depth of tree. Leaves closer to

the root segment are larger. The more segments the leaf is divided, the more realistic the leaf moves, and the more expensive the computation will be.



(a)



(b) (c)

Fig. 3: leaf blade model. (a) semi-rigidity and spring structure of leaf model (b) cubic tubes of mid-rib of leaf model (c) leaf model in our experimentation

3. Calculations and Animation of Broad-leaf Plant Model

The animation of Broad-leaf plant swaying in wind is mainly created by the forces acting on the mass points on leaves and vertexes on the stem or branch, which make meshes attached to them rotation and translation to form a variety of postures of plants. The overall animation process is as follow.

At each time step,

1. Clear the forces on all segments (i.e. set all forces to <0,0,0>)

2. Traverse the segment list and leaf points in reverse order (i.e. order from the leaves to the root), calculating all internal and external forces for each.

3. Traverse the segment list and leaf points again in the order from the root to the leaves, this time calculating the new rotation of each segment. Use this new rotation to calculate the new transformation matrix of each segment and to adjust the position and velocities of mass points for each leaf.

4. Rendering and displaying the image of plant at this time.

The local Z-axis of each segment mentioned in this paper always points along the length of the segment. We will discuss the detail of force calculations of stems or branches and leaf blades of plant respectively as follow.

3.1 Force Calculations of Stem and Leaf Mid-rib

The basic animation mechanism involves calculating a number of external and internal forces, acting on each plant segment (stem or branch and leaf mid-rib), and use these to calculate their rotational movements. When looking at each segment in isolation, we only consider rotational movement. Any translational movement of a segment is due to a movement in its parent segment.

Wind force, $\vec{F}_W^{\ stem}$ acts on the stems and leaf mid-rib segments of plant due to wind is simply calculated using the following equation:

$$\vec{F}_W^{\ stem} = S_{seg} P \tag{1}$$

where S_{seg} is the surface area of the segment which is facing the wind and p is the air pressure due to wind, at the surface of the segment. In our implementation we assume that pressure due to wind is proportional to wind velocity.

Weight force, $\vec{F}_G^{\ stem}$ due to gravity (weight) intuitively is calculated by

$$\vec{F}_G^{\ stem} = Mg \tag{2}$$

where M is the mass of a segment.

Restoration force, $\vec{F}_R^{\ stem}$ is an internal force which always acts to restore the initial orientation of the segment. To provide a robust and flexible technique to animate realistic plant movements, it is necessary to include a variable that can ultimately determine the behavior of the plant. We implement this feature through the notion of rigidity [18].

$$\vec{F}_R^{\ stem} = K_r r^2 (\Delta\vec{R} \times \vec{Z}) \tag{3}$$

where K_r is a user defined constant, representing the rigidity of the segments, r is the cross-section radius, $\Delta\vec{R}$ is the current rotation vector of the segment, and $\vec{Z}$ is a unit vector pointing along the local Z-axis. By performing a cross product between $\Delta\vec{R}$ and $\vec{Z}$ we obtain a vector which is perpendicular to both the segment itself, and the rotation axis as shown in figure 4.

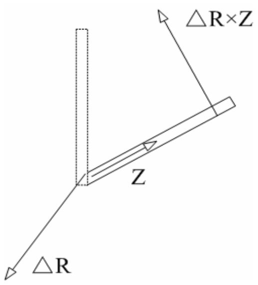


Fig. 4: Calculating the restoration force vector. Note that △R is pointing out of the paper and is perpendicular to Z.

Axial damping force, $\vec{F}_D^{\,stem}$ is an internal force which always opposes the angular velocity of the segment, and it can be calculated in the same way as $\vec{F}_R^{\,stem}$.

Transfer force, $\vec{F}_T^{\,stem}$ is defined as the force that is propagated from a segment, back to its parent. For example, when wind is exerted to a leaf (i+1 segment), some of this force is transferred to the branch it is attached, causing i segment to bend more than it would if the leaf did not exist. In our experimentation, we have taken a user defined fraction of the external forces acting on a segment, and apply to its parent, which is accordant with the order of nature.

$$\vec{F}_T^i = K_t \cdot K_r \sum_{i=1}^{n} (\vec{F}_W^i + \vec{F}_G^i + \vec{F}_T^{i+1}) \qquad (4)$$

where K_t and K_r are user defined constants representing transitivity and rigidity respectively, n is the number of child segments. $\vec{F}_W^i$ and $\vec{F}_G^i$ are the wind force and weight of the i*th* child, and $\vec{F}_T^{i+1}$ is transfer force of the i+1*th* child, after being transformed to the local space of i*th* segment. The value of the segment the apex of leaf attached is 0. We will discuss the transfer force of leaf blade in section 3.2.

Having calculated all the forces acting on a segment at a given time step, we can now derive the torque $\vec{\tau}^t$, and in turn the new rotation of the segment $\Delta R^{t+\Delta t}$.

$$\Delta\vec{R}^{t+\Delta t} = \Delta\vec{R}^t + \Delta t \cdot (\vec{V}_\alpha^t + \Delta t \cdot \frac{3}{ml^2} \cdot \vec{\tau}^t)$$

(5)

3.2 Force Calculations of points on foliage

We need also to calculate the external and internal forces acting on each point of leaves as those of the stem or branch segments. These forces include Wind force $\vec{F}_{W_{ij}}$, damping force of Point $\vec{F}_{Dij}$, Weight force $\vec{F}_{Gij}$, Support Force $\vec{F}_{Sij}$, etc. Figure 5 shows the forces model where $\vec{F}_{upij}$ and $\vec{F}_{dnij}$ are a pair of balance forces, representing the interaction relationship of upper point and lower point acting on the point in a same longitudinal (vertical) direction. Specially, we will discuss the detail of wind force and support force.

Wind force, $\vec{F}_{W_{ij}}$ is different from the stem's wind force, and it is calculated using the following equation:

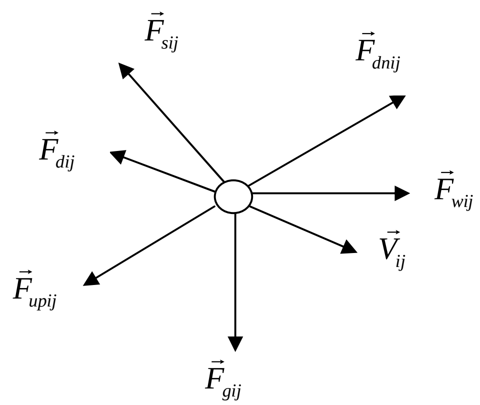


Figure 5: Modeling the forces on a point of leaf blade

$$\vec{F}_{W_{ij}} = C_w \vec{n}_{ij} \vec{v}_{ij} \bullet \vec{n}_{ij} \qquad (6)$$

where C_w is a user-specified viscosity constant for the wind, $\vec{n}_{ij}$ the normal to the surface at point P_{ij}, $\vec{v}_{ij}$ the wind vector. In order to compute the surface normals, at each point, P_{ij} of leaf, the sum of the effect of the wind on the surrounding triangles is calculated.

Support Force, $\vec{F}_{Sij}$ is expressed by

$$\vec{F}_{Sij} = C_r M_{ij} g \qquad (7)$$

where M_{ij} is the mass of the particle P_{ij}. The rigidity C_r is introduced this paper to show the feature of leaves in nature. Therefore, with a constant gravity force, the shape of the leaf can be solely determined by the rigidity, which can simplify the complexity of computation. It is different from cloth that there is a support force acting on the particle P_{ij} of leaf, which provided by the mid-rid segment of leaf.

The internal and external force vectors are summed to give the final force acting on the point P_{ij}, which is denoted by $\sum \vec{F}^{leaf}$. According to the relationship of angular quantity and linear quantity of rigid, we may decompose the final force of a point to three components of three axes in the rectangular coordinate system, and the component in the longitudinal direction is the transfer force of this point mentioned in section 3.1. Subsequently it can be calculated that the displacement of this point ran away from original point due to wind. Through constraint force in latitudinal direction, we can adjust the distance between every two points, for more details see [17].

$$\frac{d^2\vec{x}_{ij}}{dt^2} = \frac{\vec{f}_{ij}}{C_r m_{ij}} \tag{8}$$

For the points of semi-rigid, it would be necessary to respond to the collision detection explicitly after getting the new positions. In order to avoid the disadvantage of rigid, there are two parts as follow. First, we add the constraint force in parallel venation direction. Leaf point's movement will take place under this local coordinate system. The new displacement of $P_{i+1,j+1}$ is calculated using the following equation:

$$L_{ij}^{new} = \begin{cases} L_{ij}^{old} & if\ L_{ij}^{new} \geq L_{ij}^{old} \\ K_a L_{ij}^{old} & if\ L_{ij}^{new} < L_{ij}^{old} \end{cases} \tag{9}$$

where K_a is the user defined elastic constant according to the softness of leaf. L_{ij}^{new} is the new distance between two points, L_{ij}^{old} the original one. So the resulting leaf can be guaranteed not to be over-curled in wind due to this constraint force together with that in latitudinal direction. Secondly, when experiencing the third step of animation process, it would be another opportunity to adjust the new positions of leaf points properly.

The dynamic generation of the points provides the option to easily alter the number of points in longitudinal and latitudinal directions and their constraint force. The more points we are used to represent a model, the finer the movements that can be achieved, and the more expensive computation it will take.

3.3 Wind Force Model

Allowing for the effect of the wind, we can model the wind velocity with a single vector which is either randomly generated, or defined by users. To achieve this effect we have proposed the use of a regular 3D vector field, where each grid point in the field holds the velocity vector of the wind at that point. When we want to calculate the wind force on a particular segment or point, we simply look up the nearest grid point and obtain the velocity vector. The extents and resolution of the vector field is user defined, but typically it must completely contain the plant, even after application of large external forces.

In order to simulate the actual wind through this vector field, we use a series of velocity functions. Each of them in a grid point can be obtained by defining a function of position and time. The number of leading edges of the wind will affect the velocity of wind in the next grid. At any given point in time, there may be several velocity functions affecting the vector field. Therefore, the final velocity vector at each grid point is equal to the sum of all velocity functions at that point:

$$\vec{V}_{wind}(x,y,z,t) = \sum_{i=1}^{n} \vec{f}_i(x,y,z,t) \tag{10}$$

where $f_i(x,y,z,t)$ is velocity function. Note that at every time step, we keep track of the function values of each velocity function. If for a given velocity function, the resulting vectors at all the grid points are of length 0, it is no longer affecting the vector field for future time steps.

4 Experimental Results

In our experimentation, a broad leaf plant model was used to animate the realistic behavior of the plants in nature (shown in Fig. 6). The simulation is implemented by a computer with Pentium Ⅳ-2.8G MHZ CPU, 512MB DDR RAM and NVIDIA GeForce3 64MB.

Fig. 6 shows the broad leaf plant model without wind, and Fig. 7 shows some snapshots of the effects of varying wind vector acting on the plant model. At any point, the movement of plant can be dynamically controlled by injecting a global wind into the system by the user. And the structural attributes of a plant like the mass, length, rigidity and the visual attributes like its texture maps also can be interactively specified at run-time by the user.

The broad leaf plant model consists of 9529 vertices, including 6 stems and 54 leaves, and each leaf has 10 cubic tubes as mid-rib or parallel venation segment. The frame rate of the animation was found to run stably at approximately 44-46 fames per second, which is realized on the CPU and not on the graphics card, as we do not yet take advantage of any hardware acceleration.

Figure 6: The plant model without external forces at this point

Fig. 7 Simulation result of general broad-leaf tree with different force model

5 Conclusion

In this paper, we propose a physically based method for realistic rendering and animating broad-leaf plant models in real-time by combing segment method and mass point method. It is given the detail of the process of modeling the stems and leaves of plant, and the effects of varying wind. The animations of plant model are generated through a particular model of wind force. This allows for the animations to achieve suitable real-time frame rates as well as provide realistic movement. The emphasis is on a new method of combining the traditional segment methods for tree models with the improved semi-rigidity system for cloth models successfully, and we creatively apply improved semi-rigidity structure to the leaf mold in accordance with the Newton's law of motion and plant's natural attribute. By adding the elastic constraint to the parallel venation direction, the movements of the leaves are not only rigid but soft.

Moreover, the transfer force is introduced to keep the segment methods in accordance with the improved semi-rigidity methods, and it can generate more realistic movements of the plant. It is the force that the surrounding segments or points will be affected when any part of them acted by wind. The experimentation indicates that our simulation algorithm has high performance and the shapes of the plant in wind are well comparing with the former methods.

In this way, thin-leaf plant or small-leaf plant can be easily modeled by modifying the points on the semi-rigidity structure. The method will be suitable for the promising applications in computer games, animations, movie manufacture and Virtual Reality fields. Future investigations will be extended this method to be able to handle vast realistically animated landscape of vegetation and foliage.

Acknowledgments

Many thanks to the entire R&D software team at GLJL for their collaboration during the whole experimentation. Special Thanks go to Professor Sun Ji-gui for his timely help behind this piece and valuable feedback. In addition, we would also like to thank Meng Yue for providing computing resources used for rendering. This work is financially supported by the NSFC (60573182) and the NSFC (69883004).

References

[1] Prusinkiewicz, P., and Lindenmayer, A., The Algorithmic Beauty of Plants.1990

[2] Fuhrer, M., Jensen, W. H., and Prusinkiewicz, P., Modeling Hairy Plants. In Proceedings of Pacific Graphics 2004, pp.217-226.

[3] Prusinkiewicz, P., Mark, S., Hammel, E. M., Animation of plant development. ACM Press, 2001.

[4] Prusinkiewicz, P., Lindenmayer, A., and Hanan, J, Development models of herbaceous plants for computer imagery purposes. ACM Press, 1988, 141--150.

[5] Oppenheimer, P. E., Real time design and animation of fractal plants and trees. ACM Press, 1986.

[6] Weber, J. and Penn, J. Creation and rendering of realistic trees. In Proceedings of SIGGRAPH'95 (Los Angeles, Aug. 6-11). ACM Press, New York, 1995, 119-128

[7] Wong, C. J., Datta, A., Animating Real-time Realistic Movements in Small Plants. ACM press, 2004.

[8] Deussen, O., Colditz, C., Stamminger, M., and Drettakis, G. 2002. Interactive visualization of complex plant ecosystems. In Proc. IEEE Visualization 2002.

[9] Sakaguchi, T., and Ohya, J, Modeling and animation of botanical trees for interactive virtual environments. In

Proceedings of the ACM symposium on Virtual reality software and technology, ACM Press, 1999, 139-146.
[10] Ono H., Practical experience in the physical animation and destruction of trees, Proc. of Eurographics, pp.149-159, 201-203, 1997.
[11] Shinya, M., Fournier A., Stochastic Motion-Motion under the Influence of Wind[C]. Eurographics'92, 11(3), 1992. 119-128.
[12] Stam, J.. Stochastic Dynamics. Simulating the Effects of Turbulence on Flexible Structures[J]. Computer Graphics Forum (EUROGRAPHICS'97 Proceedings), 16(3): 159–164, 1997.
[13] Giacomo, D. T., An Interactive Forest[C]. Eurographics Workshop on Computer Animation and Simulation2001, 65-74, 2001.
[14] Feng Jinhui, Chen Yanyun, Yan Tao, Wu Enhua, Going with Wind—Physically Based Animation of Trees. (in Chinese), Chinese Journal of Computer, 1998, Vol.21, No.9, pp769-773.
[15] Liu Y. Q., Liu X. H., Wu E. H., Real-Time 3D Fluid Simulation on GPU with Complex Obstacles, Pacific Graphics 2004, Oct. 2004, IEEE Computer Society, Seoul, Korea, pp247-256.
[16] Dunn, F., Parberry, L., 3D Math Primer for Games and Graphics Development.
[17] Cheng C. Y., Shi J. Y., Xu Y. Q., Shum H. Y., Physically-based Real-time Animation of Draped Cloth, Proceedings Computer Graphics International 2001, 3-6 July 2001, Hong Kong.
[18] Weisstein, E. W., 2003. Cantilever – from eric weisstein's world of physics. http://Scienceworld.wolfram.com/physics/Cantilever.html

Guo Wu received the M.S. degrees in Software Engineering from college of computer science and technology in 2005. She is a Ph.D. student in Key Laboratory of Symbolic Computation and Knowledge Engineer of ministry of Education, whose major research interests include computer graphics and computer animation.