グラフィクスとCAD 110－4
(2003. 2. 14)

Efficient Geometric Modeling of Weeping Willows

Hong-Ping YAN[1,2] **Norishige Chiba**[1,3]

Abstract: *This paper presents an efficient method for modeling weeping willow trees considering the similarity existing in their structures. Weeping willow tree can be decomposed into three types of components: trunks, weeping branches, and organs (here, only leaves are considered), and object instancing technique is introduced to represent weeping branches and leaves. The geometrical size and topological information of these components can be obtained through plant growth model. Each trunk and each weeping branch are modeled as generalized cylinders. Leaves are digitized from real leaf, and texture mapping is applied to generating barks of trunks. In addition, view-dependent LOD and object instancing technique is employed to simplify geometric modeling of weeping willows. Therefore, this method can speed up modeling weeping willow tree, and meanwhile lower memory consuming. Moreover, this method can be generalized to the other weeping trees. Finally, it is conceptually simple.*

Keywords: *Geometric modeling, Growth model, Weeping willows, Object instancing, View-dependent LOD*

しだれ柳の効率的な幾何モデリング

巌紅平[2,4] **千葉則茂**[3,4]

概要：本論文では，しだれ柳の構造の類似性を考慮し，その幾何形状モデルを効率的に与える手法を提案する．本手法では，しだれ柳を，幹，しだれ枝，および器官（ここでは，葉だけを考える）の３つの構成要素に分けて考え，しだれ枝と葉はインスタンシングにより定義する．本研究で使用した植物の成長モデルからは，各構成要素の幾何学的サイズと位相的な情報を得ることができる．幹としだれ枝は一般化シリンダとしてモデリングし，葉は実際の葉に類似した形状としてモデリングする．幹の樹皮はテクスチャマッピングにより表現する．さらに，視野の深さに依存する LOD 技術を実現するために，しだれ枝と葉の幾何モデルの簡略化法も与えている．これらの方法によれば，しだれ柳のモデリングスピードを向上させ，使用メモリ量を減らすことができる．本方法は，単純であり，他のしだれ樹木にも適用することが可能である．

キーワード：幾何モデリング，成長モデル，しだれ柳，オブジエクトインスタンシング，視点依存 LOD

[1] Department of Computer Science, Faculty of Engineering, Iwate University, Morioka 020, Japan
[2] yhp@eb.cis.iwate-u.ac.jp or hongpingy@yahoo.com
[3] nchiba@cis.iwate-u.ac.jp
[4] 岩手大学工学部情報システム工学科，020-8551 盛岡市上田 4-3-5
019-621-6465，019-624-4078 fax，http://www-cg.cis.iwate-u.ac.jp/

1. Introduction

Visual plants are widely used in virtual reality environments, computer games, or architectural design. Computer modeling and simulation of trees has been of interest for many years. In this topic, most researchers focused on how to **efficiently** create the **realistic** model of tree based on certain plant growth model and geometric model.

Using particular branching patterns, some very impressive virtual plant have already been obtained. Based on 23 types of plant architectures defined by Halle [HALLE78], De Reffye created one plant model named as Automaton theory [REFFY88]. By this model, rich variety of plant structures faithful to the botanical nature could be simulated. Honda introduced a model to define the monopodial and dichotomous branching system [HONDA71]. Aono and Kunii [AONO84] demonstrated their model which can simulate some interesting features of plant, such as attraction, inhibition, and statistical variation of angle, of monopodial, dichotomous, and ternary branch patterns. Chiba [CHIBA94] brought up another plant growth model to simulate dichotomous branching system. In this model, by assuming the existence of virtual plant hormone, which can control the growth of each bud and thus of the plant, some plant growing phenomena like geotropism, apical dominance and dormancy break, can be modeled. Our weeping willow model is implemented on the basis of the model provided by Chiba.

As known, lots of plants grow in such a way named as self-similarity given by Mandelbrot [MANDE77], which has already become a paradigm in the nature world. Computer scientists use fractal-pattern technology to represent and describe the self-similarity of plants, and to recursively generate virtual plants. For example, Oppenheimer presented a fractal model of branching objects to create simply orderly plants, complex gnarled trees, leaves, vein system, etc [OPPEN86]; "L-system" [LINDE68] introduced by Aristid Lindenmayer is another fractal patterns approach. Generally, L-system works with a seed of a single character, a series of rules and symbols to generate known plants. Up to now, L-system has already been developed into some versions and widely applied in CG [PRUSI90]. Considering the similarity existing in plant structures, YAN [YAN02] developed the substructure based plant modeling method, by which most species of plant structures defined by Halle [HALLE78] can be quickly generated. In our paper, by utilizing the similarity existing among all the weeping branches and leaves in one weeping willow tree, both the weeping branch and the leaf are separately defined only once, and their different occurrences (instances) are specified by affine transformations of the prototype. This is the technique called **object instancing**. By using this technique, we can reduce the memory consuming for the geometric representation of the rendered weeping willow tree.

In order to get visually realistic plant, polygonal tree models has been widely investigated. However, due to the vast number of polygons, modeling such trees is quite time-consuming. Therefore, it is necessary to present some methods to reduce the number of polygons that form the object, without loss of the appearance. Weber and Penn [WEBER95] presented one model to create and render trees without complex botanical rules of trees. This model focused on visual effect of trees instead of their botanical nature; meanwhile they also proposed a method to smoothly degrade the tree geometry at long ranges to optimize the rendering of large quantities of trees in forested areas. In our paper, an efficient method is proposed to build 3D weeping willow trees. Our method follows the similar guidelines as that in [WEBER95].

Our method in this paper focuses on fast modeling the overall visual structure of weeping willow tree instead of a strict adherence on its botanical nature. In order to get the visually convincing plant model, both botanical and visual characteristics of plant have to be examined. To save time-consuming for modeling, **view-dependent LOD** technique is employed to lower the number of leaves in one weeping willow tree at large view depth.

The rest of this paper is organized as follows. In Section 2, we give an overview of the characteristics of weeping willows, and point out their visual features and understandable botanical nature. Based on these features, we provide some hypotheses for creating a realistic model of weeping willows. In Section 3, we introduce the detail of our modeling method according to the proposed hypotheses on modeling weeping willows. We implement our method and obtain 3D weeping willows. The results and extensions of our work are presented in Section 4, and 5.

2. Features of weeping willows

From computer graphics point of view, in order to get a realistic 3D model of weeping willow and meanwhile to simplify the rendering procedure, we need to know the visual characteristics and simple botanical natures of weeping willows.

We took some photos of one weeping willow tree (Figure 1(a), (b), and (c)). From these photos and what we observed, we can point out the outstanding features of weeping willows:

Weeping willow is a graceful, fast growing tree with a single rounded and short main trunk, and lots of long, slender weeping branches, which reach to the ground, attached by large quantity of long, wide, lance-shaped drooping leaves. Its bark is rough and dark-brown or gray. Fall color of leaf is yellowish.

(a) Branches (b) Leaves (c) Trunks

Figure 1. Weeping willow trees

Considering these features of weeping willows, we decompose its structure into three types of components: **trunks** (including **main trunk** and **lignifying branches**), **weeping branches** and **organs** (**leaves** with leafstalks, flowers, and fruits. Here, only leaves are modeled). The geometrical size and topological information of trunks are obtained through the growth model explained in [CHIBA94] in detail. Each trunk and each weeping branch are modeled by generalized cylinders. Leaves are digitized from real leaf. Object instancing technique and view-dependent LOD technique are used to simplify our geometric model of weeping willows.

In the next section, we will explain how to model weeping willow trees.

3. Modeling of weeping willow trees

3.1. Growth model

In this paper, we use the dichotomous plant growth model to simulate the growth of weeping willow. Using this model, we can obtain the topological information of the structure of weeping willow (the direction and location of trunks, and the initial shooting directions and positions of weeping branches on their parent trunks), and the geometrical information of the trunks (length and thickness). The original geometrical information of one weeping branch (length and thickness) and one leaf (length and width) are fixed by the users.

Based on the topological and geometrical information, we begin to separately create each of 3D trunks, weeping branches and leaves according to different geometric models.

3.2. Geometric model

As mentioned in Section 2, we regard each trunk and each weeping branch as generalized cylinders, i.e. a set of cylinders with certain length and thickness calculated from the growth model as described in [CHIBA94]. Each cylinder is triangulated in the way shown in Figure 2. The origin of the local coordinate system of the cylinder is located at the center point of its bottom cross section, and the X-axis and Y-axis are located on the plane to which the bottom cross section belongs, and the Z-axis is the rotation axis of the cylinder. Suppose XOY plane of the world coordinate system of one weeping willow tree is the ground, and the positive Z-axis is away from the ground to the sky. It is well known that the weeping branches of weeping willow bend downwards. Considering the similarity existing among all the weeping branches, we assume that the tip of each weeping branch always points toward the ground, and parallels the negative Z-axis (See Figure 3). Additionally, we suppose all the cylinders representing one weeping branch are in the same plane with the first cylinder. In order to simplify the geometric model of weeping willow, we think all the weeping branches on one weeping willow always have the same properties: the same length, approximated by the same number of cylinders. Therefore, we introduce object instancing technique to our geometric model. We just represent the weeping branch once in the memory, which can reduce memory consuming than that store every weeping branch in the memory.

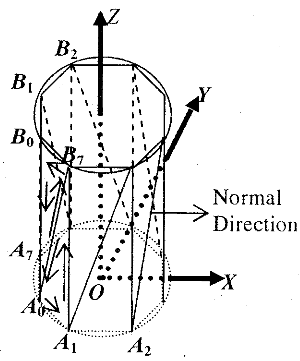


Figure 2. Triangle approximation of cylinders

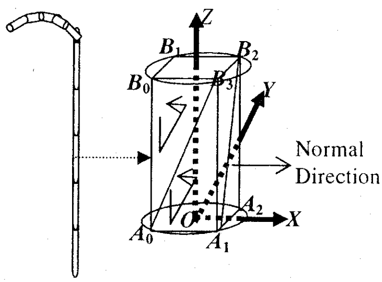


Figure 3. Geometric model of weeping branch

For the weeping willow, one leafstalk has only one leaf, so we consider leaf and leafstalk as one object, simply called leaf. Without loss of the general appearance of weeping willow, suppose all the leaves on one weeping willow always have the same geometrical size, although it is not true in the nature. Therefore, similar with weeping branch, we just store the information of leaf once in the memory. The data file representing X and Z coordinates of the polygons composing the leaf is given by the users. We think Y coordinates of all the polygons for each leaf as 0. As observed, the leaves only attach along the weeping branches and grow towards the ground. In our model, we attach the leaf at the tip of each cylinder representing each weeping branch, and one cylinder has only one leaf. So, according to the assumption on modeling weeping branch, the number of leaves attaching on every weeping branch is the same. For modeling simplification, we regard the growing direction of each leaf as the same of the negative Z-axis.

Now, we get the geometrical models for trunks, weeping branches and leaves in their local coordinate systems. To obtain the whole weeping willow tree, we just need to transfer these models from their local coordinate systems to the world coordinate system by making a series of transformations, such as translation, rotation, on these objects.

3.3. Leaf simplification

As observed, there are a lot of leaves attaching on each weeping branch. In our model, at least ten leaves are created along each weeping branch, and at least two weeping branches for one lignifying branch. For a weeping willow tree at t growing step, if each leaf is approximately modeled by n triangles, and suppose there are m weeping branches attaching on one lignifying branch, l leaves on each weeping branches, we can compute the number of leaves on this tree by $l*m*(2^{t-1}+2^{t-2})$. This value is defined as the maximum number of leaves on this tree; and the number of triangles for modeling all these leaves will reach $n*l*\mathrm{m}*(2^{t-1}+2^{t-2})$. Therefore, the older the tree is, the more triangles it has, which will result in slow visualization speed and high memory consuming.

However, at long range, such as 100 meters, a low-resolution tree can be rendered fast by decreasing the number of leaves with little or no loss in the apparent quality, which is known as view-dependent LOD (Level of Detail) in computer graphics domain. Therefore, it is reasonable to progressively reduce the number of leaves attaching on every weeping branch with the increase of view range (Figure 4), here different view depth, i.e. the distance between the viewer and the observed weeping willow. When the view depth increases, the number of leaves on each weeping branch will decrease progressively and meanwhile the number of polygons for each leaf keeps unchanged. The relationship between the view depth and the number of leaves on the designated weeping willow tree can be expressed as following:

$$LeafNum = MaxLeafNum^{\left(1-\frac{Depth}{MaxDepth}\right)}$$

Here, *MaxDepth* is the maximum view depth; *Depth* is the current view depth; *MaxLeafNum* is the maximum number of leaves on designated weeping willow tree; *LeafNum* is the number of leaves on the tree after simplification, and it is rounded to the nearest integer.

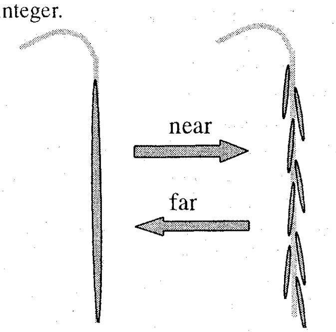


Figure 4. View-dependent leaf simplification

4. Results

We have implemented our method as described above on PC platform based on the software of "Virtual Bonsai". The software was developed by the JFP, Inc. Company, which employed the theory described in [CHIBA94]. The implementing procedure can be described as the following steps:

(1) Build growth model;

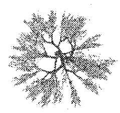
(2) Triangulate trunks, weeping branches and leaves according to the results from step (1);

(3) Create 3D model of the weeping willow tree;

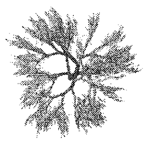
(4) Render the model of weeping willow at different view depths.

We generate some images of weeping willows: the same weeping willow observed from different viewpoints (Figure 5), different kinds of weeping willows (Figure 6) and weeping willows in different seasons (Figure 7). Here, we set the maximum value of view depth as 20, l as 12, n as 20, and m as 5, so the maximum number of leaves for creating such a weeping willow tree at growing age t is $60*(2^{t-1}+2^{t-2})$.

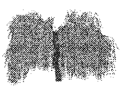
The growing age of the weeping willow in Figure 5 is 8 growing periods. In Figure 5, the trees in (a), (b) and (c) are observed respectively from bottom viewpoint, top viewpoint and side viewpoint at view depth 10. Figure 5(d) is the side view of the same tree observed at view depth 5. From different viewpoints, we can get different visual features of the tree. Figure 5(a) and (b) give us the information of the round and wide spread crown of weeping willows; From Figure 5(c) and (d), we can have the weeping impression, and also have the information of the main short trunk of weeping willow tree. In addition, as mentioned above, the view-dependent LOD technique is employed to reduce the number of leaves in one weeping willow at different view depths. In Figure 5(c) and (d), the number of polygons approximating the leaves on this weeping willow is about 2147 and 22240 respectively, and the corresponding total time for modeling and visualizing this tree is about 35.9 and 268.2 seconds respectively. In the case of Figure 5(c), the visualization speed is faster than that in the case of (d), and the memory consuming in the case of (c) is also much less than that in the case of (d) while the tree in both cases can give us realistic impressions. This demonstrates the efficiency of our method presented in this paper.



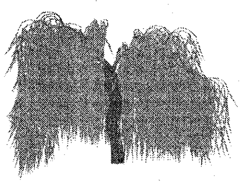
(a) Bottom view



(b) Top view



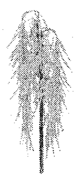
(c) Side, far view



(d) Side, close view

Figure 5. Same weeping willow observed from different viewpoints

(a)

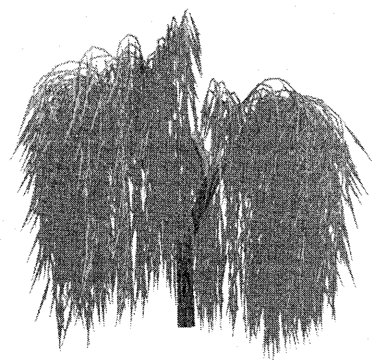


(b)

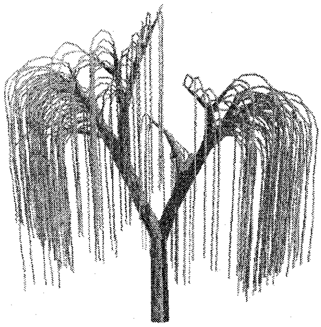
Figure 6. Different kinds of weeping willows. (a) Golden weeping willow; (b) Weeping pussy willow

(a) Spring

(b) Summer



(c) Autumn



(d) Winter

Figure 7. Weeping willow in different seasons

5. Conclusions and extensions

We explored the modeling method of weeping willow in this paper. This method is inspired by the similarity existing among weeping branches of weeping willow. Since the weeping branches of all the weeping trees have the same property: bending downward, the modeling method given in this paper can be generalized to all the weeping trees. Furthermore, by employing the object instancing technique and view-dependent technique, our method can lower memory consuming and speed up modeling procedure. Finally, this method is conceptually simple.

In this paper, we assume that all the weeping branches and leaves have the same geometrical size and direction, and that there is the same number of leaves attaching along each of them; in fact, all these parameters vary with the growing time of the branches, therefore the rendered model looks unnatural at the small view range. To get a realistic model, one of our future works is to build the relationship between the information of weeping branches and their growing time. Obviously, more computer resources (e.g. CPU time and memory) are needed to generate realistic model when it is near the viewer. In order to obtain a good tradeoff between the realism of the images and the computer resources, some techniques have to be employed to control the Level-of-Detail (LOD) of weeping branches. In addition, some weeping trees have very complex trunks, how to simulate complex tree trunk is also our future research topic.

ACKNOWLEDGEMENTS

We would like to express our sincere appreciation and our gratitude to the Telecommunication Advancement Organization of Japan (TAO) for their financial support, and Dr. Ch.-H. Pan, Mr. K. Suzuki and Mr. Mamat for their helpful discussion and help.

REFERENCES

[AONO84] M. Aono, T. Kunii. Botanical tree image generation. *IEEE computer graphics and applications*, Vol. 4, No. 5, pp. 10~34, May, 1984.

[BLOOM85] J. Bloomenthal. Modeling the mighty maple. *Computer Graphics*, Vol. 19, No. 3, pp. 305~311, 1985.

[CHIBA94] N. Chiba, K. Ohshida, K. Muraoka, M. Mura, and N. Saito. A growth model having the abilities of growth-regulations for simulating visual mature of botanical trees. *Computer & Graphics*, Vol. 18, No. 4, pp. 469~479, 1994.

[HALLE78] F. Halle, R. Oldeman, and P. Tomlinson: *Tropical Trees and Forest: an architectural analysis. Springer-Verlag, Berlin,* 1978.

[HONDA71] H. Honda. Description of the form of trees by the parameters of the tree-like body: effects of the branching angle and the branch length on the shape of the tree-like body. *Journal of Theoretical Biology*, pp. 331~338, 1971.

[LINDE68] A. Lindenmayer. Mathematical model for cellular interactions in development, I&II. *Journal of Theoretical Biology*, pp. 280~315, 1968.

[MANDE77] B. Mandelbrot. *Fractals: form, chance and dimension, W.H. Freeman and Co., San Francisco*, 1977.

[OPPEN86] P.E. Oppenheimer. Real time design and animation of fractal plants and trees. *SIGGRAPH'86 Conference Proceedings*, Vol. 20, No. 4, pp. 55~64, 1986.

[PRUSI90] P. Prusinkiewicz, A. Lindenmayer. *The algorithmic beauty of plants. Springer-Verlag, New York*, 1990.

[PRUSI01] P. Prusinkiewicz, L. Mundermann, R. Karwowski, B. Lane. The use o positional information in the modeling of plants. *Proceedings of SIGGRAPH 2001, Annual Conference Series,* pp. 289~300, 2001.

[REEVE85] W.T. Reeves, R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Computer Graphics*, Vol. 19, No. 3, pp. 313~322, 1985.

[REFFY88] P. Reffye, C. Edelin, J. Francon, M. Jaeger and C. Puech. Plant models faithful to botanical structure and development. *Proceedings of SIGGRAPH 88, Annual Conference Series*, pp. 151-158, 1988.

[SMITH84] A.R. Smith. Plants, fractals and formal languages. *Computer Graphics*, Vol. 18, No. 3, pp. 1~10, 1984.

[WEBER95] J. Weber, and J. Penn. Creation and rendering of realistic trees. *Poceedings of SIGGRAPH95, Annual conference Series*, pp. 119~189, August, 1995.

[YAN02] H.P. Yan, J.-F. Barczi,, Ph. de Reffye, and B.G. Hu. Fast Algorithms of Plant Computation Based on Substructure Instances. *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*, Vol.10, No.3: 145-153, 2002.