Continuous LOD Model of Coniferous Foliage

Qingqiong Deng, Xiaopeng Zhang, Sebastien Gay and Xiangdong Lei

Abstract—**Interactive and realistic rendering of forest is a challenge due to a huge amount of tiny geometric detail. We present an approach to construct continuous level of detail (LOD) models for coniferous foliage geometry. The models are defined with multi-resolution cylinders and semi-transparent lines. When a coniferous leaf is wider than one pixel on the image space, cylinder is used to represent it; otherwise, line will be used. And Line transparency is applied to show the sub-pixel detail of foliage needles. In addition, lines are further merged to a new line when they are too thin on the image space, so that the number of lines will be decreased when they are farther from the viewer. All steps of simplification are recorded in preprocessing. During rendering, appropriate approximation models will be established for different viewpoints automatically on the fly, and the transition from high resolution to a lower one is continuous. Compared with other foliage simplification methods, our approach can produce more realistic results for close views, and can get higher compression ratio for distant coniferous foliage.**

Index Terms—**Coniferous foliage, level of detail, plant rendering.**

I. INTRODUCTION

Fast and realistic rendering of plant communities is an important task for numerous applications, such as computer animation, flight simulation, urban visualization, visual environment and entertainment. However, real-time rendering of such scenes is a big challenge, since brute force geometry rendering far exceeds current CPU and GPU capabilities. And it will become aggravated when rendering scenes that consist of a very common kind of tree: coniferous trees, such as pine, cypress, and fir. They usually contain much more leaves than other kinds of trees with similar height or crown volume, and their leaves are so tiny that aliasing is often involved.

On the other hand, the geometry and topology of plants and trees are very different from other objects, especially the foliage part, which is composed of many small and isolated surfaces. Other ordinary objects such as houses and tables are usually represented by some large and connected surfaces. Therefore, most accelerating methods that can work on common objects will fail, or produce visually unpleasant results when dealing with vegetation.

Some special approaches for foliage simplification have been proposed recently [1, 2, 3, 4]. They are efficient in geometric simplification and multi-resolution rendering of broad and thin leaves, but not coniferous foliage.

In this paper, we present a geometric simplification method for the particular case of the leaves of pine-trees or fir trees. It is based on a cylinder plus line representation combining with the LOD technique. Cylinder models are used to represent needles near the viewer, while translucent lines are used for those at a distance. In addition, lines will be merged together for views even farther away.

The following section is thus dedicated to previous related work. Section Ⅲ details the representations that we adopt, including both the cylinder and line models. In Section Ⅳ, lines are merged together for further simplification. After simplification, the geometry and simplification process data are recorded. Based on the record, the error controlled rendering is implemented (Section Ⅴ). The results of the implementation are shown in the next section, while the last section concludes.

II. RELATED WORK

Plant modeling has been extensively explored. At present, there are many mature systems that can generate realistic plants and trees efficiently. L-system [5], AMAP [6] and Xfrog [7] are the typical ones. L-system is a string rewriting system operating on a set of rules, i.e. bud functioning, branching patterns, and death processes. AMAP is a procedural model based on the birth and death of growing buds, and real measured data, so that it is faithful to botanic structure and development. Xfrog is a convenient tool for interactive plant modeling that allows easy generation of many branching objects including flowers, bushes, trees, and even non-botanical things. As the result of realistic plant modeling, the number of the polygons is often large for a normal plant. So the representation of a small scene or forest with hundreds of plants is usually very huge. Thus, real-time rendering of outdoor sences with the original full geometric models is impossible in practice.

Many methods have been proposed to accelerate the visualization of outdoor scenes. Most of them make use of image textures to represent original complex models, such as billboard [8], billboard clouds [9, 10], multi-layer Z-buffers [11], layered depth images [12], hierarchical bi-directional textures [13], and volumetric textures [14, 10]. These methods

Manuscript Received on March 23, 2007.

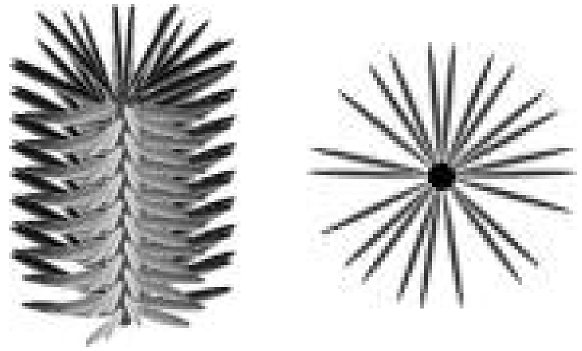
Qingqiong Deng is a PhD student in the Sino-French Laboratory LIAMA and National Laboratory of Pattern Recognition at CAS Institute of Automation. Her main interest is computer graphics. E-Mail: qqdeng@nlpr.ia.ac.cn.

Xiaopeng Zhang is an associate professor in the Sino-French Laboratory LIAMA and National Laboratory of Pattern Recognition at CAS Institute of Automation. His main interest is computer graphics and pattern recognition. E-Mail: xpzhang@nlpr.ia.ac.cn.

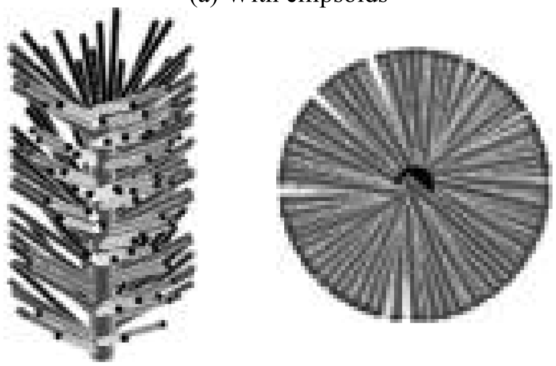
Sebastien Gay is a Master Student in the Biologically Inspired Robotic Group(BIRG) at the Federal Institute of Technology of Lausanne (EPFL) inSwitzerland. His main interest is Biologcally-Inspired computing. E-Mail: sebastien.gay@insa-lyon.fr.

Xiangdong Lei is an associate professor in the Institute of Forest Resource Information Techniques, Chinese Academy of Forestry. His main interests are forest management and growth modeling. E-Mail: xdlei@caf.ac.cn.

are efficient and can get fantastic results for distant plants and trees. However, they need immense memory to store the texture images. Their parallax effects are usually bad, and artifacts are obvious in close views. Some use a point instead of a triangle to represent a leaf when the projected size of the leaf on the image space is smaller than a pixel [15, 16]. Such technique is very efficient, but it can only be used for plants and trees that are not viewed from close up, and their rendering results are usually blurry, since the topology of a plant can not be well preserved with point model. In recent years, some people have developed polygon-based methods [1, 2, 3, 4] that can diminish the number of leaves in a crown while maintain its appearance by iterative leaf collapse process: two leaves disappear to create a new one. The polygon-based methods can produce highly realistic results. However, they are not very efficient in decreasing geometric complexity, and when applying them to coniferous leaves, the visual effect of original models can't be faithfully approximated (see Fig. 5).


45°view Top view

(a) With ellipsoids


45°view Top view

(b) With cylinders

Fig. 1. Representation of coniferous leaves [17].

Some models have been proposed especially for coniferous leaves. In these models, pine needles are represented by ellipsoids [17] (Fig. 1(a)) or cylinders [18] (Fig. 1(b)). They can get realistic results when rendering pines that are very near. But they have only one resolution, so the visualization is very costly. And even worse, such representations will result in aliasing when trees are not close. Meyer et al. present a method that builds three discrete LOD models for coniferous leaves based on the topology of trees [19]. It is able to render pine-trees efficiently in ray-tracing and free of aliasing. Nevertheless, popping is obvious when switching from one resolution to another.

III. MODELS OF CONIFEROUS FOLIAGE

The trees used in our method are generated by the AMAP Genesis^{TM} [6]. The needles of the pine trees are represented by polygons, more exactly, triangles. Such representation is too coarse for close views (see Fig. 4(a)), and too complex for far views. In order to get more realistic results when zooming in and decrease geometric complexity when zooming out, we use two different representations, cylinder and line, to represent coniferous leaves according to their distances from the viewer.

3.1 Cylinder Model

Using either ellipsoids or cylinders can get realistic results for close coniferous foliage. However, the polygonization of an ellipsoid is much more complex than that of a cylinder. For simplification, we prefer to use cylinders to represent needles when they are larger than one pixel on the image space.

The input of our method includes the centerline and the radius of each needle leaf, so it is easy to construct cylinders from the input data.

To display cylinders using OpenGL, tessellation is necessary. We convert each cylinder to a sequence of prisms with different numbers of sides. Where the minimum number of the sides is 3, and the maximum is L_{max}, which is defined by users. In this way, we can adopt appropriate detail level depending on the distance between the viewer and the rendered tree.

We know that the error of a circle with radius R to be represented by an inscribed m-sided polygon is $R\times(1-\cos\pi/m)$. During rendering, by given a pixel error threshold ε, which is a tradeoff between rendering speed and visual quality, we can get its corresponding space error threshold $e=2\varepsilon\times D_{tree}\times\tan(\theta/2)/W$, where D_{tree} is the distance from the tree to the viewer; θ is the view angle; W is the height of view plane. Then the number of the required sides of a cylinder can be obtained by the equation: $n=\pi/\arccos(1-\rho)$, where $\rho=e/R$.

As a result, when $\rho<2$, cylinder will be polygonized into a prism with corresponding number of sides. However, when $\rho\geqslant 2$, even the 3-side prism is too much detail, in this case, lines instead of cylinders will be used to represent needles.

3.2 Line Model

When the projected size of a leaf on the image space is equal to or smaller than one pixel, line model is used to represent the leaf. It is fit to use lines for distant needles, since their projected shapes are similar to lines in a raster display.

We replace each cylinder by its centerline, and the width of the line is one pixel on the image space. To determine the normal vectors of the two ends of the line, we use the information of the triangle replaced by this cylinder. The normal vectors of the two ends are the same, and they are both equal to the normal of the triangle. The error of approximating a cylinder by a line is defined as $2R$, where R is the radius of the cylinder.

When the distance from a tree to the viewer becomes larger, the sizes of lines should be decreased. In this case, we adopt translucent line model, and the transparency of a line is determined by the following equation:

$$T(\rho)=2/\rho \quad (\rho\geqslant 2)$$

Thus the farther a tree is, the more transparent its needles will be, which results in good visual quality.

IV. LINE MERGING

When a pine tree becomes farther away, we can use fewer lines to represent its leaves. We can apply a sampling technique such as the common point-based rendering methods usually do to randomly select a fraction of the lines to approximate the crown according to the projected size of the tree. But this technique often leads to unreasonable holes. We prefer to decrease the number of lines through line merging, where two lines are transformed into a new one, with the similar position as the two original lines.

The idea of the hierarchy in the HUO algorithm [4] is inherited in our method, and there are two levels: the phyllotaxy group and the whole tree. The line merging process is repeatedly implemented within the phyllotaxy group at first, until all the lines in the phyllotaxy group are united together as a single line. Such lines are then recursively merged in the second level.

4.1 Construction of a New Line

We use a simple way to construct the line generated in a line merging. The four vertices of the two simplified lines form a set, from which we can find out two vertices, whose distance is the longest. Such two vertices are then chosen as the two ends of the new line, and their normal vectors stay the same.

4.2 Cost Function

At each step of the decimation, to determine which pair of lines to merge, we apply a cost function to estimate the difference between each pair of lines. The less different a pair of lines is, the smaller its cost value will be, and the more prior it will be merged.

The function consists of three items, including direction difference, position difference and merging time difference. In the following, we will explain the meanings of the three items respectively.

Direction difference of line X and line Y is defined by the equation:

$$S_1(X, Y)=1-|<T(X), T(Y)>|$$

where $T(X)$ and $T(Y)$ are the two uniform direction vectors of the two lines, and<*,*>is the dot product of two vectors. $S_1(X, Y) \in [0,1]$, and the more collinear the two lines are, the smaller it will be.

Position difference $S_2(X, Y)$ is defined as the minimum distance between the two lines X and Y. To make it uniform, we multiply it with the reciprocal of the diameter of the bounding box of the crown, so that it will fall in the range of 0 to 1.

The merging time of a line records the number of needles that it replaces. The larger the merging time is, the later the line should be merged. And we believe that the pair of two lines with close merging time is more preferential to be merged than the pair with more different merging times. So merging time difference can be defined as:

$$S_3(X, Y)=[G(X)+G(Y)+|G(X)-G(Y)|\,]/(2\times N)$$

where $G(X)$ and $G(Y)$ are the merging times of the line X and Y, N is the number of needle leaves of the tree. Also, $S_3(X,Y)\in[0,1]$.

Based on the above three items, we define the cost function $S(X,Y)$as:

$$S(X,Y)=k_1\times S_1(X,Y)+ k_2\times S_2(X,Y)+ k_3\times S_3(X,Y)$$

where $k_i>0$ and $k_1+ k_2+ k_3=1$. In practice, we use $k_1=0.25$, $k_2=0.25$, and $k_3=0.5$.

4.3 Merging Error

The error of a line merging is defined as the following equation:

$$e(X)=2\times max\{D(Y_i, X)+e(Y_i)/2;\ i=1,2\}$$
$$D(Y_i, X)=max\{proj(p_j, X);\ p_j\in Y_i, j=1,2\}$$

where X is the new line, Y_1 and Y_2 are the original lines replaced by the line X, $e(Y_i)$ is the merging error of line Y_i, p_1 and p_2 are the two ends of a line, and $proj(p_j, X)$ is the projected distance from the point p_j to the line X.

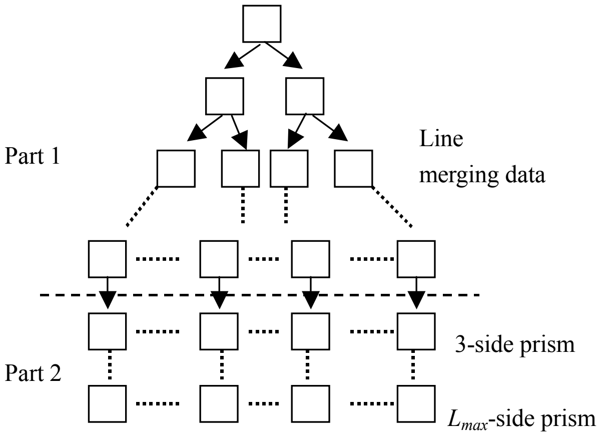


Fig. 2. Data structure of our method.

For each line, using the half of its merging error as a radius, we can construct a cylinder whose axis is the line. Then for a new line generated in a line merging process, the two cylinders corresponding to its children lines will fall inside the cylinder of the new line. As mentioned before, the error of approximating a cylinder by a line is defined as $2R$, where R is the radius of the cylinder. So the definitions of the two errors are coherent.

V. IMPLEMENTATION

5.1 Data Record

After simplification, the geometric information and the simplification process data are recorded in the hard disk.

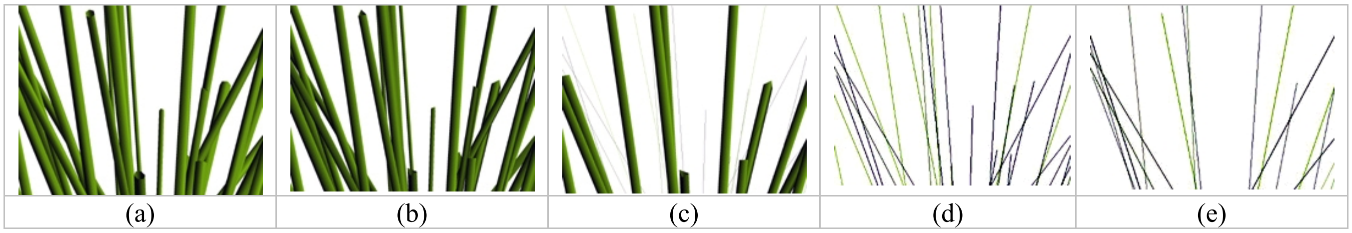


Fig. 3. The numbers of sides of cylinders are changed from 8 in (a) to 3 in (b) as the error threshold increase. In (c), some leaves are represented by cylinders and the rest are represented by lines. Leaves in (d) and (e) are described by lines, and lines in (e) are merged, so that fewer lines are used.

(a) Triangle model (b) Cylinder model

Fig. 4. Comparison between triangle model and cylinder model.

For geometric information, we first store the cylinder representations of whole leaves of a tree, and then the lines, including both the original lines and the lines generated in line merging.

A special tree is constructed to record the data of the simplification process. The tree is composed of two parts. As shown in Fig. 2, Part 1 is the binary tree of line merging, where the new line created in a line merging process is recorded as the father of the two collapsed lines. Part 2 stores sequences of prisms with different numbers of sides. For each node of the tree, the corresponding error due to simplification is recorded too.

5.2 Error Controlled Rendering

When rendering, the user defined pixel error ε can be converted to spatial error e. Based on this spatial error e, the tree record mentioned above will be traversed on the fly, until meeting the nodes whose recorded errors are equal to or smaller than e. At the same time their corresponding geometry will be sent to the GPU.

When the rendered pine is lush, the number of the nodes in the tree record is large. So we will spend quite a long time in traversal. To avoid traversing the tree from top to bottom every time, we partition the tree record into three parts: lines generated in line merging, original lines and cylinders. For each part, we calculate its range of errors. During the rendering, we first compare the spatial error e with the three error ranges, and then start the traversal from the part, whose error range contains the spatial error e. In this way, we can speed up the traversal, and improve the rendering efficiency.

5.3 Shading

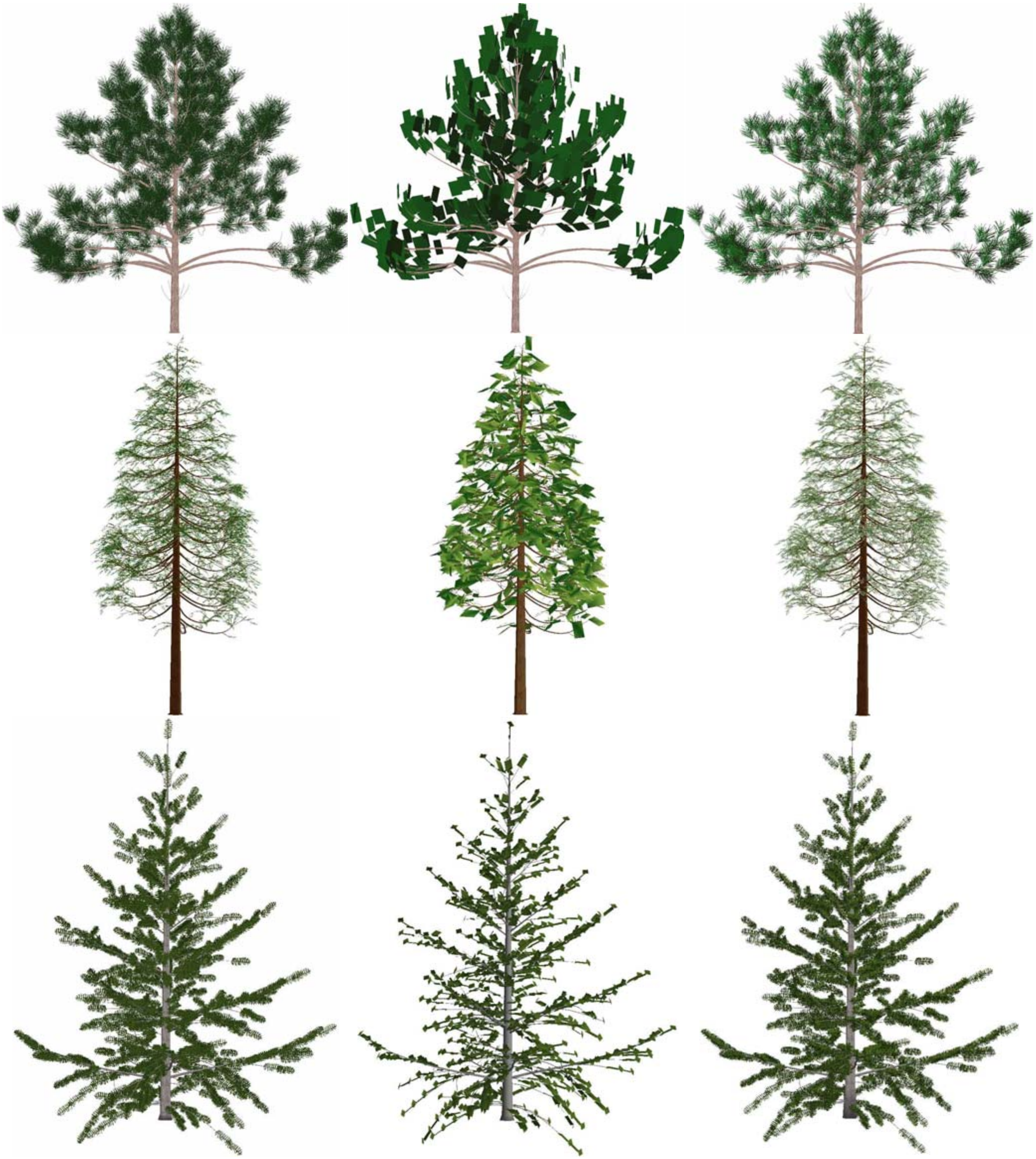
No texture is used for leaves in this work. The whole leaves of each tree share a same material, which consists of four items: ambient, diffuse, specular reflectances and specular exponent. The reflectance is described by four values: RGBA. With the default lighting model of the OpenGL, the triangles of prisms are shaded based on their normal and material. So are the lines, for which transparence is considered as a coefficient to affect the Alpha value of its material.

VI. EXPERIMENTS

The algorithm was written in the C language with the OpenGL, and it was implemented on a PC with Pentium Ⅳ Xeon at 2.8GHz, memory 512 MB.

The sparse parts of trees, such as pine needles and pinecones, are simplified using our method. Trunk and branches are simplified through a dynamic simplification of cylinders.

In the transition from prisms to the original lines, there will be a visible increase in the opacity of the tree. In order to make this transition smooth, more transparent original lines are needed. We adopt another parameter k as the coefficient to line transparence.



Original models HUO results Our results

Fig. 5. Comparison of the simplification results of HUO and our method.

Fig. 3 shows the simplification process. As the user defined pixel error increases, the representations of leaves change from cylinders to lines.

Using the cylinder model instead of the triangle model for coniferous leaves, we can get more realistic results for close views. Fig. 4 shows the comparison between the two models, where the leaves are represented by triangles in Fig.4 (a), and cylinders in Fig. 4(b).

Compared with the existing foliage simplification methods, our method can get results that are more faithful to the original models. Fig. 5 shows a comparison between the HUO algorithm and our method on three trees: a 10-year-old Scots pine, a 30-year-old Lawson cypress and an 8-year-old Norway spruce. The figures in the left column are the original models, the figures in the middle are the simplified results of the HUO algorithm, and the figures in the right column are ours.

Fig. 6. LOD models of a 10-year-old eastern white pine. See Color Plate 23.

Fig. 6 shows a LOD model of a 10-year-old eastern white pine. As the distance from the viewer increasing, the representation of the pine becomes coarser. The detail information of the four trees, including the distance from the viewer, the number of polygons and lines that are used to represent sparse organs, are shown in TABLE 1.

TABLE 1: THE INFORMATION OF THE FOUR PINES in Fig. 6.

Pine SN	a	b	c	d
Distance	12.22m	26.71m	56.15m	118.40m
Poly number	0	0	0	0
Line number	23,310	14,391	6,212	2,022

Fig. 7 shows a virtual forest with 120 trees. It contains 8 kinds of trees: a 10-year-old and a 20-year-old Scots pine, a 10-year-old and a 15-year-old holly, a 15-year-old Siberian columnar crab apple, a 10-year-old eastern white pine, a 12-year-old Lombardy poplar, and an 18-year-old Aleppo pine. They are placed in different positions with different orientations to generate the forest. The total polygon number of sparse organs (leaves, fruits, flowers, pine needles and pine cones) of the forest is 16,749,820. The sparse organs in this view are represented by 78,920 polygons and 1,588,092 lines. Treating lines equal to polygons, the compression ratio is 9.9%. And 836,481 polygons and 140,942 lines are included for the branch models. The image resolution is 2048 by 1536 pixel and the frame rate is 0.33Frame/sec (shadow calculation is not included since it is implemented in a post-process, and no

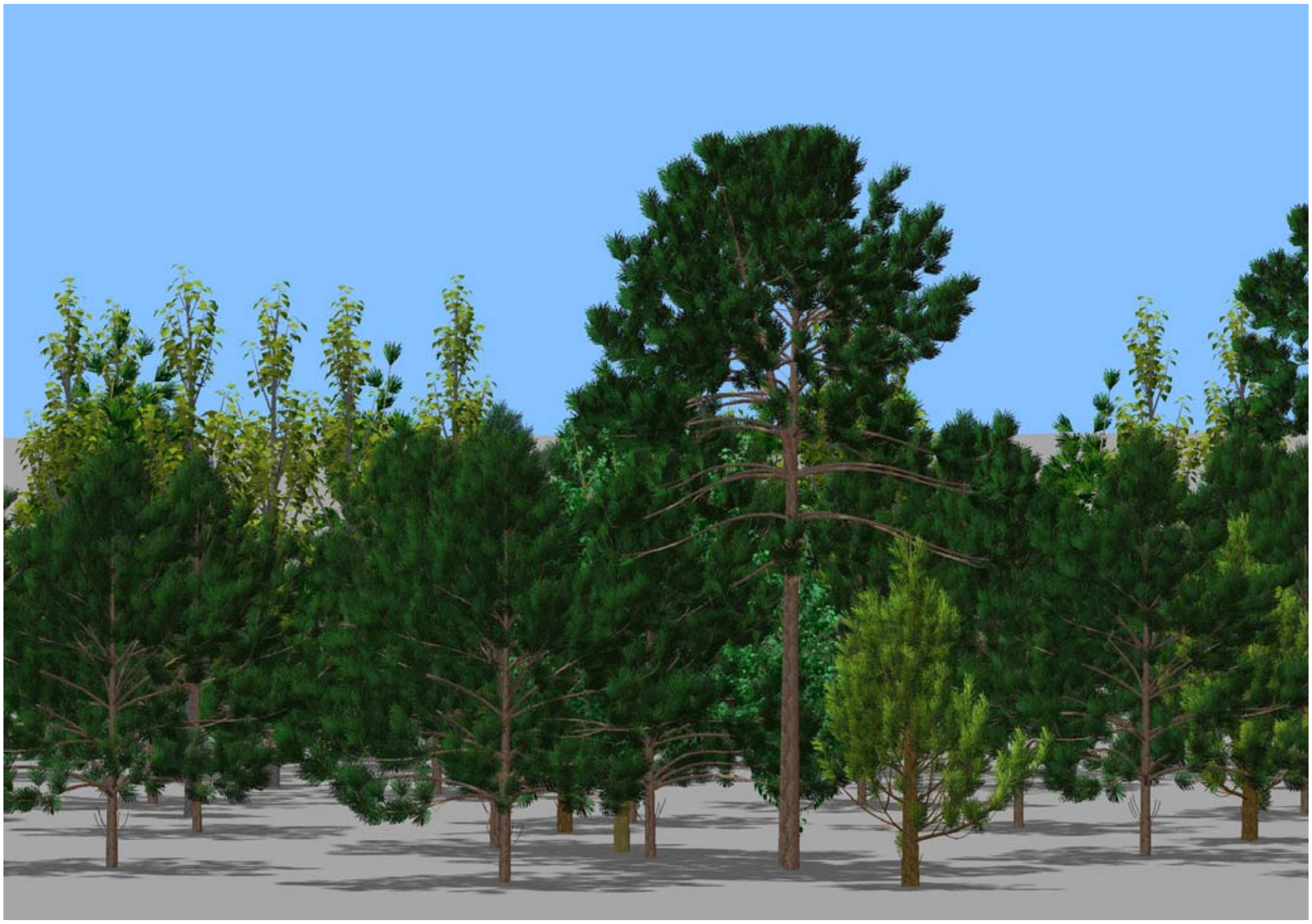


Fig. 7. Visualization of a forest. See Color Plate 24.

hardware support is used to accelerate rendering).

VII. CONCLUSION AND FUTURE WORK

We propose a simplification algorithm especially for coniferous foliage. It is based on a mixed model defined with cylinders and lines. By using the cylinder representation, we can get realistic results for close views. By using the line representation for pine needles when they are far from the view, we can dramatically decrease the geometric complexity. Besides that, lines can be merged, so that the number of lines can be further decreased. Compared with the other existing foliage simplification methods, we gain two main advantages. One is that the simplified results of our method are quite faithful to the original models. The other is that we have greatly sped up the visualization of the scenes that consist of pines or firs. Another advantage of our method is that it is free of aliasing, so that the visual qualities of our results are quite nice.

There are still some potential improvements in the future. Right now, we use the knowledge of the phyllotaxy for line merging, and it much improves the efficiency of the simplification. But there are some more important properties of plant structure, for example, the topological information, which we do not make use of present. So we should apply the other priori knowledge when merging lines in future work. Unfortunately, our method still requires large storage memory. For each tree, it needs to store the geometric information of prism sequence and lines (includes both the original lines and the lines generated in line merging), as well as the simplification process data. In most cases, we find that the prism consisting of 3 sides is enough, so a possible way to decrease storage cost is to use the 3-sideprism instead of the prism sequence. Another more technical way is that we can partition the information data into several parts, and only load the necessary part into the memory when rendering.

One interesting work is to speed up the rendering of coniferous foliage with the assistance of current powerful GUP. For example, shade cylinder on single rectangle using texture mapping instead of shading for each side of its approximate prism. Furthermore, to achieve better rendering results, special illumination models, such as the ad hoc lighting models, are also needed in the future.

ACKNOWLEDGEMENT

All original plant models in this paper are from the output of AMAP-Genesis^{TM}. This work is supported by National Natural Science Foundation of China projects No. 60073007, 60473110, 30371157; by National High-Tech Research and Development 863 Plan of China under Grant No. 2006AA01Z301; by the French National Research Agency within project NATSIM ANR- 05-MMSA-45; and by LIAMA funding with the project GreenLab.

REFERENCES

[1] I. Remolar, M. Chover, O. Belmonte, J. Ribelles and C. Rebollo. Geometric Simplification of Foliage, *Eurographics 2002-Short Presentations*, 2002.

[2] I. Remolar, M. Chover, J. Ribelles and O. Belmonte. View-dependent Multiresolution Model for Foliage, *The 11-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision' 2003, University of West Bohemia*, Campus Bory, February 3-7, 2003.

[3] X. Zhang and F. Blaise. Progressive Polygon Foliage Simplification, Proceeding of *2003' International Symposium on Plant growth Modeling*, simulation, visualization and their Applications. pp. 217-230, October 13-16, 2003.

[4] X. Zhang, F. Blaise and M. Jaeger. Multiresolution Plant Models with Complex Organs, Proceedings of *ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications VRCIA 2006*, pp.331-334. June 14-17, 2006.

[5] P. Prusinkiewicz and A. Lindenmayer. The Algorithmic Beauty of Plants, Springer-Verlag, New York, 1990.

[6] P. de Reffye, C. Edelin, J. Francon, M. Jaeger and C. Puech. Plant Models Faithful to Botanical Structure and Development, *Computer Graphics (Proc. Siggraph'88)*, vol. 22, pp.151-158, ACM Press, New York, 1988.

[7] O. Deussen and B. Lintermann. Interactive Modelling of Branching Structures Technical Sketch, *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH),* 1996.

[8] J. Rohlf and J. Helman. IRIS Performer: A high Performance Multiprocessing Toolkit for Real-time 3D Graphics, *In SIGGRAPH 1994 Conf. Proc*. pp. 381-394. 1994.

[9] A. Fuhrmann, E. Umlauf and S. Mantler. Extreme Model Simplification for Forest Rendering, in *Natural Phenomena 2005: Eurographics Workshop on Natural Phenomena (2005)*, Eurographics Association, pp. 57-66. 2005.

[10] S. Behrendt, C. Colditz, O. Franzke, J. Kopf and O. Deussen. Realistic Real-time Rendering of Landscapes Using Billboard Clouds, in *Proceedings of Eurographics 2005*, vol. 24, no. 3, pp. 507-516, 2005.

[11] N. Max. Hierarchical Rendering of Trees From Precomputed Multi-layer Z-buffers, in *1996 Eurographics Workshop on Rendering (1996)*, pp. 165-174.

[12] J. Shade, S. Gortler, H. L. Wei and R. Szeliski. Layered Depth Images, in *SIGGRAPH 1998 Conf. Proc. (1998)*, pp. 231-242.

[13] A. Meyer, F. Neyret and P. Poulin. Interactive Rendering of Trees with Shading and Shadows, in *2001 Eurographics Workshop on Rendering Techniques*, pp. 183-196. 2001.

[14] P. Decaudin and F. Neyret. Rendering Forest Scenes in Real-time, in *Rendering Techniques 2004*, pp. 93-102. June 2004.

[15] O. Deussen, C. Colditz, M. Stamminger and G. Drettakis. Interactive Visualization of Complex Plant Ecosystems, in *IEEE Visualization 2002*, pp. 219-226. 2002

[16] G. Gilet, A. Meyer and F. Neyret. Point-based Rendering of Trees, *Eurographics Workshop on Natural Phenomena*, August 2005.

[17] M. Disney, P. Lewis and P. Saich. 3D Modelling of Forest Canopy Structure for Remote Sensing Simulations in the Optical and Microwave Domains, *Remote Sensing of the Environment*, vol. 100, no. 1, pp. 114-132. 2006.

[18] S. Smolander and P. Stenberg. A Method to Account for Shoot Scale Clumping in Coniferous Canopy Reflectance Models, *Remote Sensing of the Environment*, vol. 88, pp. 363-373. 2003.

[19] A. Meyer and F. Neyret. Multiscale Shaders for the Efficient Realistic Rendering of Pine-trees, in Proceedings of *Graphics Interface 2000*, pp. 137-144.

Qingqiong Deng was born in Hunan province of China, in July 2nd, 1981. Received bachelor degree from Huazhong University of Science and Technology, Wuhan, China, in 2003. Since then, study in LIAMA (Sino-French lab in computer science, automation and applied mathematics), Institute of automation, Chinese academy of science, Beijing, China, for master degree and PhD degree. The major field of study is real-time visualization of outdoor scenes.

Xiaopeng Zhang is an associate professor in the Sino-French Laboratory LIAMA and National Laboratory of Pattern Recognition at CAS (Chinese Academic of Sciences) Institute of Automation. His main interest is computer graphics and pattern recognition. He received his PhD degree in computer application from CAS Institute of Software in Beijing, China in February 1999.

Dr. Zhang was invited as a foreign specialist for forest visualization in INRIA, the French National Institute for Research in Computer Science and Control. He is also an associate professor of Beijing Graduate School of CAS, and the course he is teaching there is *computer graphics and application*. He received the National Scientific and Technological Progress Prize (Second Class) in 2004. He has more than 50 publications and he advised 7 PhD students.

He is presiding some projects on computer graphics, pattern recognition and virtual reality from National Natural Science Foundation and National High-Tech Research and Development 863 Plan of China. He is the conference co-chair of PMA06: The Second International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications. And he is also the program co-chair of Edutainment 2008, the third International Conference of E-Learning and Games.

Sébastien Gay was born in Mulhouse (France) in 1984. graduated in 2007 as Informatics Engineer at the INSA Lyon (France). Received a Master degree in Artificial Intelligence at the university of Lyon in 2007.

In 2006, he did an internship at the LIAMA in Beijing (China) where he worked on the Plant Growth Modeling - Plant Visualization project. His work was included in a publication for *The Second International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications* in Beijing (2006). In 2007 he did a Master project at the BIRG in the Ecole Polytechnique Fédérale de Lausanne (Switzerland) on reconfiguration algorithms for modular robots. His research interests are now biological models and bio-inspired computing.

Xiangdong Lei received PhD degree in forest management from Beijing Forestry Univeristy, Beijing, P.R.China, in 2000.

Since 2000, he has been working in the Institute of Forest Resource Information Techniques, Chinese Academy of Forestry, Beijing. He is employed as an associate professor. He has published scientific articles in international journals and conferences. Here are two published articles: [1] X. Lei, Y. Lu, C. Peng, X Zhang and L. Hong. Development of growth and structure of semi-natural larch-spruce-fir forests in Northeast China：12 years results after thinning, Forest Ecology and Management, vol. 240, pp.165-177.2007. [2] X. Lei, M. Chang, Y.Lu , T. Zhao. Design and implementation of individual tree growth visualization system of Larch(Larix olgensis),Computer Engineering and Applications, vol. 42:180-183. 2006. He is leading some projects on forest growth modeling, sustainable forest management and forest resource monitoring from National Natural Science Foundation and Ministry of Science and Technology. His main interests are forest growth simulation and sustainable forest management.

Dr. Xiangdong Lei is the advanced member of China Society of Forestry.