Grid-based View-Dependent Foliage Simplification

Qingqiong Deng1,2, Xiaopeng Zhang1,2 †

1 Sino-French Laboratory LIAMA, Institute of Automation, CAS, Beijing 100190, China
2 National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing 100190, China

Abstract

This paper presents a new algorithm to efficiently construct level of detail plant foliage models based on a uniform subdivision of the crown volume. This algorithm includes two phases: preprocessing and rendering. In the preprocessing, the leaves of a tree are first clustered by a regular 3D grid structure, and then for the leaves in each cell of the grid, a recursive leaf union is performed hierarchically in two levels: phyllotaxy group and whole foliage. The simplification data of each cell will be stored in the hard disk as a binary tree structure in the end of preprocessing. In the rendering phase, the global level of detail of a tree is constructed according to its distance to the viewer. Then, the local level of detail model of each cell is extracted according to a simple heuristic rule: leaves on the rear or inside of the crown are statistically less visible than front ones, so they can be rendered with coarse models. The contributions of this approach are higher efficiency in level of detail construction in preprocessing and higher data compression for view-dependent rendering.

Keywords: Simplification; Level of Detail; Visibility; Foliage; Geometric Compression

1. Introduction

Today, the major bottleneck preventing the widespread use of digital video is rapid retrieval of desired information based on content from a huge database. An important initial task for video indexing is to partition video into temporal segments. Scene change detection is one of common approaches of video segmentation. Detection of scene transitions is a challenging research topic and many algorithms have been presented. Model-based methods [1, 2] were applied in certain specific videos, such as news video and sports video, but these methods need good domain knowledge for modeling. The methods based on shot clustering [3, 4, 7] are also widely applied for scene detection. However, most of these methods only effectively parse shot clusters of interlaced relationship into the same scene. In fact, those shot clusters of adjacent relationship may also belong to the same scene. In temporal graph-based methods [5, 6], shot clusters forming loops in a temporal graph were grouped into one scene. However, loops do not always exist for shot clusters of some scenes.

Self-occlusion normally exists in the crown of a tree. The leaves on the rear part of the crown are more likely to be occluded than the leaves in the front. Classical painters are good at applying the occlusion. They usually use initially large rough strokes to paint the leaves that are hidden, and then details are added over with finer strokes to represent visible leaves.

This foliage simplification method in this paper takes benefit of this technique. In the preprocessing, the leaves are clustered with a uniform grid structure, and LOD model series are constructed for each cell of the grid then. A LOD model is extracted for each cell according to both distance and visibility in rendering. As a result, the foliage is not uniformly simplified: coarse elements are used in the rear part of the crown and fine ones in the front. Therefore, the overall compression can be higher with non-uniform LOD model.

† Corresponding author.

Email address: xpzhang@nlpr.ia.ac.cn (Xiaopeng Zhang)

1553-9105/ Copyright © 2008 Binary Information Press
August, 2008
2. Related Work

Plant modeling has been extensively explored, and many advanced systems are well developed, including L-system [1], AMAP [2] and Xfrog [3]. Using these systems, we can generate realistic plants and trees efficiently. However, the number of the polygons is often very big for a normal plant. So the representation of a small scene or forest with hundreds of trees is usually very huge. Thus, real-time rendering of outdoor scenes with original full geometric models is impossible in practice.

Acceleration techniques of rendering have been well cared and many methods have been proposed. These methods can be classified into three general categories: point-based, image-based and polygon-based. Point-based rendering uses points to represent leaf geometry when its projected size on the image space is smaller than a pixel [11, 12]. It is efficient, but it can only be applied to plants and trees that are not viewed close up. Besides, the synthesized images are usually blurry, since the topology of a plant is destroyed. Image-based rendering has the advantage of its rendering cost independent of the geometric complexities of objects. There are various approaches in this field, such as billboard [4], billboard cloud [5, 6], multi-layer Z-buffers [7], layered depth images [8], hierarchical bi-directional textures [9], and volumetric textures [10, 6]. They are efficient and can get fantastic results when rendering plants and trees at a distance. However, the methods need a huge memory to store texture images. Besides, they parallax effects are usually weak, and artifacts are obvious on close views. Although the imaged-based and point-based methods are efficient in rendering plant scenes, geometry is necessary in many cases, for example, when the plant is close.

Many polygon simplification methods were proposed [13] on continuous manifold objects. However, when applied on trees, they may produce acceptable results for trunks, but will fail on foliage. To solve this problem, some especial approaches have been proposed for the simplification of foliage [14-18].

The Foliage Simplification Algorithm (FSA) [14] was proposed for quadrilateral leaves, diminishing the number of leaves but keeping the appearance of the crown by leaf collapse. FSA was improved in Progressive Leaves Union (PLU) [15] by more reasonable measurements of similarity to select the best leaf pair in decimation. In addition, the PLU is view-dependent, and all simplification information is recorded on the hard disk. Appropriate approximation models are extracted for different viewpoints automatically. Hierarchical Union of Organs (HUO) [16] is developed in PLU to hierarchically and respectively simplify tree organs with respect to some basic botanical structure of phyllotaxy and anthotaxy. Hidalgo et al propose a method to simplify foliage Based on FSA and PLU recently [17]. Pairs of leaves that minimize a cost function are selected repeatedly and collapsed. Its main contributions are texture generation and LOD model extraction with a linear data structure rather than a binary tree structure. However, it is not error-controlled. The leaves in the list are neither sorted by the steps in which they generated, nor values that measure their errors due to simplification, so that fine and coarse details may irrelevantly coexist in a single LOD model. This will be a big cost in LOD model extraction.

Visibility in the foliage is not considered in the above approaches, and the simplification is uniform all around the crown. Many view-dependent non-uniform algorithms were proposed in [19, 20], but they are for manifold mesh models. The algorithm View-Dependent Foliage (VDF) was proposed [18] to deal with the isolated foliage polygons, but an active list of visible polygons need be kept and modified using temporal coherence when the viewer shifts. So it is costly both in computation time and storage.

3. Grid-based Foliage Simplification

The idea of hierarchical simplification in the Hierarchical Union of Organs (HUO) [16] is extended here. There are two levels in the foliage crown: phyllotaxy group level and the whole tree level.

In the preprocessing phase, we first construct leaf clusters, or phyllotaxy groups according to the phyllotaxy structure for a tree. And then, we organize the leaf clusters using a regular structure: 3D grid. Finally, we construct continuous LOD models for each cell separately and independently by implementing recursive organ union process on the two levels one after the other. We do not cluster the leaves of the tree
but the phyllotaxy groups, since we want to keep the topology information of leaves. We can use hierarchical simplification within each cell to accelerate the preprocessing cost.

A 3D uniform grid is constructed and used to partition the crown of a tree in Fig. 1. The criterion of a leaf cluster belonging to a grid cell is that its center is inside. The center of a leaf cluster is defined as the barycenter of the vertices of the leaves belonging to this leaf cluster. The sizes of the grid in three coordinate directions are $G_x$, $G_y$, and $G_z$ respectively, which are dependent on the size and the shape of the bounding box of the crown. In practice, we use 4 for the direction with shortest length, and other two sizes can be calculated according to the ratios of their lengths over the shortest one. Fig. 1 shows the grid-based simplification, where Fig. 1 (a) is a uniform grid to partition the crown volume.

![3D uniform grid](image1)
![Grid-based Classification](image2)
![Binary tree structure](image3)

Fig. 1 Grid-based simplification

The leaves in different cells constitute some clusters in Fig. 1 (b) and they are simplified independently and respectively. For each cell, the Progressive Union of Complex Organs [16] is performed in two levels sequentially. The first is within each leaf cluster, or phyllotaxy group of the cell, until all the quadrangles in the leaf cluster are united as a single quadrangle, which is called the representative quadrilateral. After obtaining all the representative quadrilaterals of the cell, the Progressive Union of Complex Organs are performed on them until only one quadrangle left.

In each step of organ union, the new generated leaf is recorded as the father of the two collapsed leaves. Therefore, the simplification process data constitutes a binary tree in Fig. 1(c). The green nodes are original leaves before simplification, and they form the finest representation of the cell. The blue nodes are leaves generated by organ union process composing coarser representations of the cell. For each node of the binary tree, its corresponding error due to simplification is recorded also.

Using grid-base simplification, we have two main advantages. The first is that each cell has its own binary tree recording the simplification process since leaves in each cell are simplified independently. The cells can be processed separately rather than globally according to the distance from the cell to the viewer. In addition, the visibilities of all cells can be easily taken into account for LOD extraction. As a result, different simplification degrees coexist in the same crown.

The second advantage is that the preprocessing can be much improved. The existing foliage simplification methods are very slow. They need to compare all leaf pairs and select the pair with minimal cost value to collapse in each step of decimation, thus shows a typical $O(n^2)$ complexity, where $n$ is the number of leaves of a tree. When $n$ is large, the methods will spend very long time in preprocessing. In our method, the simplification is confined to each cell. So the complexity of selection will be decreased as $O(n^2/(G_xG_yG_z)^2)$, so that the simplification (see table 1) can be much accelerated.

4. Multi-resolution Foliage

In rendering, the user defined pixel error can be converted to a special error through the distance from the camera to the tree and the camera characteristics. Based on the spatial error (noted as $e$), the binary tree of each cell will be traversed until meeting the leaves whose simplification errors are not larger than $e$. Such
leaves will compose an appropriate LOD for current view.

However, recursive traversals of binary trees are time-consuming, especially when the number of leaves of rendered tree is huge. Dachsbacher et al. [21] have proposed a method to sequentialize a binary tree of a point cloud model to a list, so that the recursive rendering procedure can be replaced by a sequential loop over the list. We use this method to sequentialize the binary structure to a list cells.

We can represent the leaves in different cells by a uniform LOD. In this case, as the tree becomes farther, the spatial error \( e \) will become larger, thus the representation of its foliage will become coarser.

It is more reasonable to use different resolutions for a tree, since occlusion exists widely in the crown. The leaves on the rear part of the crown are more likely to be hidden than the leaves in the front, so that they can be rendered coarsely. As a result, the number of rendering primitives can be decreased, and then a higher rendering efficiency can be achieved.

Using the distance from the viewer to the crown, and the distance from the viewer to a cell, it is easy to determine whether the cell is in front of the crown or not. If the latter is smaller than the former, the cell is though of as in the front, otherwise, it is on the back. The cells in front are determined to be visible, while the ones on the rear are occluded or partially occluded. The occlusion degree of a rear cell depends on the relationship between the viewing direction that starts from the camera toward the center of the crown and the position of the cell. The closer the cell to this viewing position, the lesser visible it will be. This comes from the fact that when projecting the crown onto the image plane, the cells falling in the inner part of the projected image are more likely to be occluded than those whose projected regions are close the silhouette.

To discriminate leaves in different spaces of the crown taking advantage of their visibilities, we adapt the spatial error for each cell. We first calculate a global spatial error \( e \) using the distance from the camera to the tree and the camera parameters, and then we convert the global error \( e \) linearly to get the local one \( e_i \) of cell \( i \) according to the occlusion degree \( h_i \) with the following equations:

\[
e_i = (1 + h_i)e
\]

\[
h_i = \begin{cases} 
0, & \text{if } d_i \leq d \\
\hat{v} \cdot \hat{p}_i, & \text{if } d_i > d
\end{cases}
\]

where \( d \) is the distance from the viewer to the center of the crown, \( d_i \) is the distance between the viewer and the center of the cell \( i \), \( \hat{v} \) is the normalized vector from the camera to the center of the crown, and \( \hat{p}_i \) is the normalized vector from the center of the crown to the center of the cell \( i \).

The spatial errors of cells thus are non-uniform after this adaptation. Fig. 2 (a) is the front view, Fig. 2 (b) is the left view, and Fig. 2 (c) is the top view of the same model. The colors are defined based on the value of \( h_i \) (Fig. 2(d)). The green color means visible regions, where leaves are represented by finer LOD, while the red means the most occluded ones, and the leaves falling in these regions are described with less details. Larger spatial errors are adopted for cells on the rear of the crown than that in the front; so coarser representation will be used for leaves that are less visible.

---

**Fig. 2** A non-uniform LOD model of a 15-year-old holly represented by false colors

5. **Implementation and Results**
We implemented the proposed approach in C language, using the OpenGL library, running on a PC with Pentium™ Xeon at 2.40GHz, 512M RAM. Virtual trees used in this paper are generated by AMAP Genesis™ software [2]. Experiments have been performed on different species of trees. The sparse organs of trees, such as leaves, flowers and fruits, are simplified with the proposed approach while the trunk and branches have been simplified through a dynamic simplification of cylinders.

In order to show the effect of multi-resolution on compression ratio, we compare the HUO algorithm [16] and our method on four species of trees: a 15-year-old holly tree, a 25-year-old Siberian columnar crab apple tree, a 35-year-old white poplar tree, and a 25-year-old Scots pine tree. The results are shown in Fig.3, where Fig. 3(a1), Fig. 3(b1), Fig. 3(c1), and Fig. 3(d1) are original models. Fig. 3(a2), Fig. 3(b2), Fig. 3(c2), and Fig. 3(d2) are the simplified models of the HUO with a uniform level of detail depending on the distance to the viewer. Fig. 3(a3), Fig. 3(b3), Fig. 3(c3), and Fig. 3(d3) are the simplified models of our method with different levels of detail depending on the distance to the viewer and visibility. The sub-title of each figure is the number of leaves. It can be seen that the visual qualities of each pair of associated simplified results are quite similar, but the compression ratios of the multi-resolution results are higher.

![Fig. 3 Comparison on quality and compression ratio between our method and HUO [3]](image-url)
Table 1 Comparison on preprocessing time

<table>
<thead>
<tr>
<th>Tree name</th>
<th>Holly Poly number 30,350</th>
<th>Crab apple Poly number 129,489</th>
<th>White poplar Poly number 568,645</th>
<th>Scots pine Poly number 191,940</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing time of HUO (s)</td>
<td>484</td>
<td>7,948</td>
<td>668</td>
<td>191,940</td>
</tr>
<tr>
<td>Our preprocessing time (s)</td>
<td>3</td>
<td>187</td>
<td>76</td>
<td>3</td>
</tr>
<tr>
<td>Gained time ratio</td>
<td>161</td>
<td>43</td>
<td>105</td>
<td>223</td>
</tr>
</tbody>
</table>

We did also compare our method with the HUO on pre-processing time with the same tree models in Fig.3. The information details, including tree names, the number of leaves of the original model, the time spent for simplification of leaves with HUO, the corresponding time of our method, and the times of acceleration, are shown on Table 1. It turns out that our method is much more efficient.

Fig. 4 (a) shows LOD models of a 25-year-old chestnut tree. As the distance from the viewer increasing, the representation of the tree becomes coarser. But their visual effects are similar. A zoomed view of the four models is shown for comparison in Fig. 4(b, c, d, e) with the number of leaves below.

Fig. 5 is the real-time rendering of a small forest. It consists of 119 trees from 6 instances: a 20-year-old Japanese weeping apricot tree, a 25-year-old southern catalpa, a 25-year-old Siberian columnar crab apple, a 20-year-old holly, a 12-year-old common linden, and a 20-year-old Yunnan poplar. The total foliage polygon number in the forest is 8,307,081. The corresponding number of foliage polygons for the view shown is 815,532, so the compression ratio is 9.8. The image resolution is 1280 by 1024 pixel and the frame rate is about 25 frame/second.

6. Implementation and Results

We present an efficient way to construct non-uniform LOD models for tree leaves with the help of the 3D grid, a uniform subdivision of the crown volume. The proposed approach is efficient to compress foliage.

Compared with previous foliage geometric simplification methods, it has two main advantages. It is more efficient than the former methods; and its data compression is higher with mixed LOD models of different resolutions.

There is still some work that deserves to improve in the future. At present, the leaves are clustered using a static uniform grid. In the future, the crown volume subdivision should be closely related to the shape of the crown, so that the size of the bounding box of the crown and total number of leaves included are
considered. The uniform grid should be replaced by an irregular spatial subdivision, such as an octree or a Binary Space Partitioning (BSP) tree. If the density of leaves in the crown is considered, the simplification should be more reasonable.

For real-time rendering, significant improvements could be achieved when this approach is combined with other advanced real-time rendering techniques, such as image-based rendering on far plants, or deferred rendering on closer ones. Point based rendering is another possible way when the plants are distant, so that grid cells themselves can be used as points to render foliage.

Fig. 5 Rendering of a virtual forest

Acknowledgement
All original plant models in this paper are from the output of AMAP-Genesis™. This work is supported by National Natural Science Foundation of China projects No. 60473110, 60073007; by National High-Tech Research and Development 863 Plan of China under Grant No.2006AA01Z301, 2008AA01Z301; by the French National Research Agency within project NATSIM ANR-05-MMSA-45; and by MOST International collaboration project No. 2007DFC10740.

Reference