SNS COLLEGE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)
COIMBATORE – 35
DEPARTMENT OF COMPUTER SIENCE AND ENGINEERING
UNIT 2

Management of Large Scale Environments & Real Time Rendering

Realistic Real-Time Rendering for Large-Scale Forest Scenes

Rendering a realistic large forest scene in real-time plays an important role in virtual reality applications, such as, virtual forestry, virtual tour and computer games. Since a forest consists of an extensive number of highly complex geometric models, real-time forest rendering is still a challenge. Several techniques are there to render highly realistic forests with realtime shadows. Since a forest with thousands of plants contains a vast amount of geometry, an efficient level of details (LOD) algorithm can be used to generatemultiresolution (MR) models according to forest features. Leaf modeling method is used to have leaf models match leaf textures well. Parallel-split shadow mapping (PSSM) generation scheme can be used in rendering system. The data of tree models are organized into vertex buffer objects to enhance the rendering performance. A tree clipping operation is designed both in the view frustum and in the light frustum to avoid rendering models outside thecurrent frustum and to remove the popping up and off effects. The combination of these techniques allows to realistically render a large forest with a large number of highly detailed plant models in real-time.

TREE MODELING AND MODEL PROCESSING

Modeling of tree foliage and processing of tree model are two key aspects of this work. Special techniques are used to construct LOD tree foliage models.

Tree modeling and simplification

A new technique with texture mapping of LOD tree foliage models is presented in this subsection.

Branch model processing

Tree skeleton models could be obtained from plant modeling software where tree skeleton models can also be obtained by two input images: sketches of main branches and crown silhouette on one input image, and sketches of two boundary branches and crown silhouette on anther input image. A series of static multiresolution models are constructed from the tree skeleton, which are used in real-time scene rendering talked next section. This method is used to construct branches only because it can generate continual LOD models. Continual LOD models are useful not only for efficient memory cost but also for model switching while wandering in a forest.

Leaf model processing

A leaf can be approximated as a mesh by two rows of quadrilaterals. The major axis of the rectangular mesh coincides with the main vein of the leaf. It is observed that the main vein usually determines the curl degree of the leaf, so a quadratic function is used to fit the main vein and the leaf geometry can then be obtained. Accordingly, the leaf LOD models can be easily constructed. However, as the main vein curves more and more in space, the two boundary edges of the mesh along the direction of the major axis should curl accordingly in order for better approximation of the mesh to the leaf shape, as shown in Figure 3.10(b).

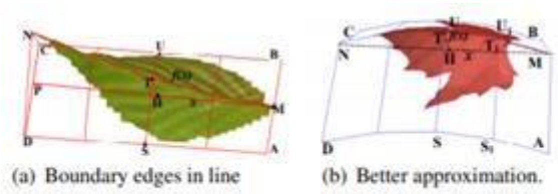


Fig. 3.10. Leaf modeling

A main vein, call as an arc, can be seen as a part of a parabola. The arc length, that is the leaf length L, and the arc height D can be measured from a real leaf. Leaf curl degree can be defined by the dihedral angle which can be represented by 6 AMB. The coordinates of A,B,M,N and the leaf unit normal vector $\vec{n}$ are given before a leaf is drawn in a designated position. A parabola function can be used to model the main vein:

$$f(x) = 4D(L-x)/L^2,\ 0 \le x \le L \qquad (1)$$

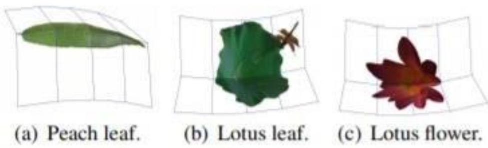


Fig. 3.11. A leaf model with different texture images.

The proposed leaf geometry model has an advantage over other models in that different texture images (in alpha format) can be changed conveniently to a same leaf model with non-degenerative visual effect, as displayed in Figure 3.11.

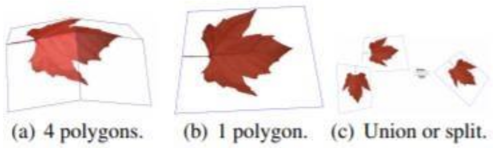


Fig. 3.12.Leaf models change base on quadric function

Not only can the leaf models easily match a given leaf texture, but also they can balance the visual effect and rendering speed when a large forest are to be displayed. If a tree is closed to the view point, the highly refined leaf model can be adopted; if it is far away fromthe view point, the model of low resolution is employed. Figure 3.12 shows a series of leaf models with different complex degree. Transition between a polygons model as in Figure 3.12 (a) and the model of one quadrilateral as in Figure 3.12(b) can be finished by using the function 1. If the distance between the tree and the view point decreases, the value of D decreases and the number of polygons decreases too. Transition between the Figure 3.12(b) and Figure 3.12 (c) is performed using the a method that can simplify plant organ following the structure of leaf phyllotaxy and flower anthotaxy, it is adopted to manage foliage in rendering for high simplification. In addition, the number and distribution of phyllotaxy in each branch are considered.

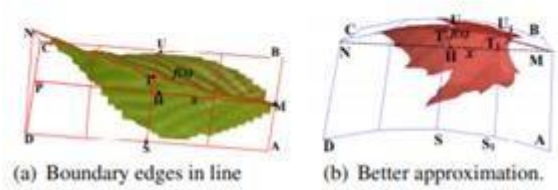


Fig. 3.13 Rendering results when a leaf represented with different polygons.

Phyllotaxy is the basic information of the leaf and flower distribution in the tree structure, so phyllotaxy geometry is constructed with experience or by measuring a real phyllotaxy of a tree. Some important parameters, such as the number of leaves in each node, the angle between two leave, the angle between axis and leaf, should be taken into account. The visual realism of a tree becomes less pleasing when the number of applied polygons for a leaf decreases. Figure 3.13 (a) shows a part of a tree whose leaves are represented with 4 polygons, and it looks realistic. In Figure 3.13 (b), each leaf is represented with one rectangle and the visual realism decreases. And in Figure 3.13 (c), several leaves is represented with one quad by union strategy and its realism is the weakest too.

Construction of Plant LOD models

A large forest often consists of trees from thousands to millions. If each tree is modeled with full information, memory will be exhausted quickly. In order to save memory while keeping the realistic, LOD models are often used in practice. However, too many LOD models also exhaust the memory for a large forest. Therefore, 4 or 5 models of different resolution is used as the LOD series to represent a tree instance. Because occlusion iscommon in forest rendering, the trees in the distance can be simplified greatly. With the simplification methods, five LOD models are selected for a tree instance according to rendering effect at different ranges of distance. The closer the tree is to the view point, the finer its selected model is. Figure 3.14 illustrates a LOD series. The number of polygons in each model is shown in the subtitle of each sub-figure.

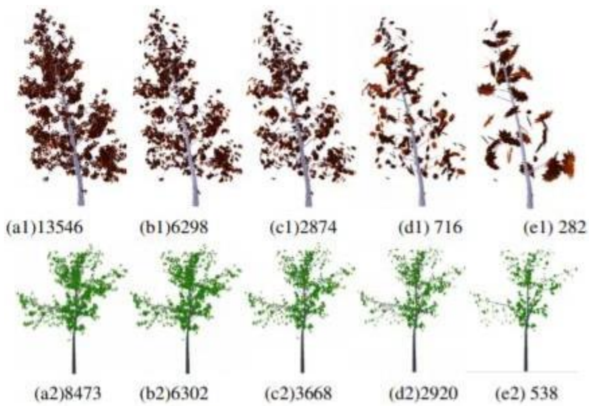


Fig. 3.14 LOD series of tree models

REAL-TIME FOREST RENDERING

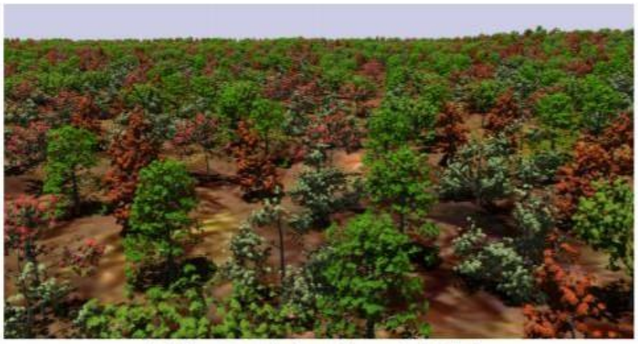
PSSM is employed to produce real-time and anti-aliasing shadows. Figure 3.15 shows the rendering results of different single trees used in our system.

(a) (b) (c)

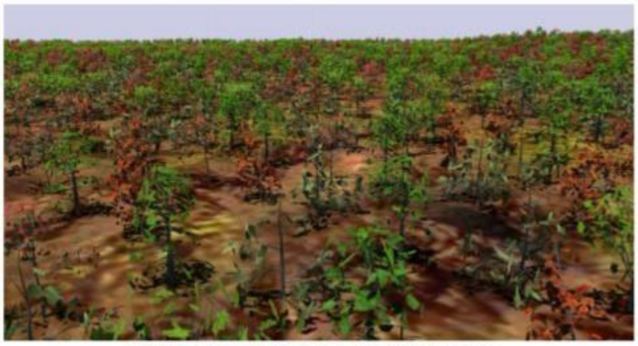
Fig. 3.15 Some single trees rendered with real-time and antialiasing shadows. (a) A simple tree. (b) A black popular with complex branching structure. (c) An apple tree with fruit and dense foliage.

Forest Scene Layout

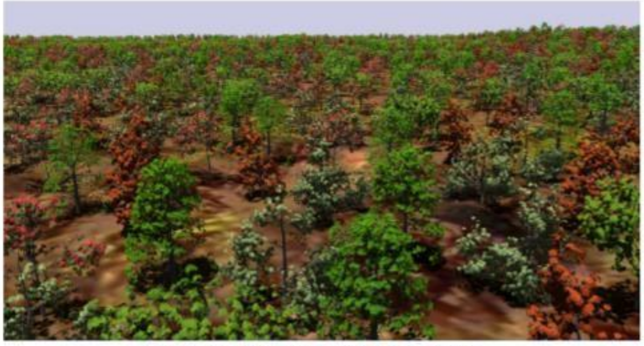
With some LOD tree models created with the method presented above, a large forest scene can be constructed as follows. A digital terrain model (DTM) covers an area of 262144 square meters. In the digital terrain model, there are 5 instances which produce 7446 tree positions with uniform distribution. Each instance owns 4 LOD models (from highest resolution to lowest one denoted as t1,t2,t3,t4)and one of them will be used according to the distance dct between the view point and the tree position. Figure 3.16(c) shows the experiment result with real-time shadow. In the scene, there are 7446 trees and 1671 in view port which displays 2162588 polygons, and the time costs about 0.083s per frame. Without LOD strategy, the forest costs more than 0.33s per frame, if only the detail models are used atall positions (Figure 3.16 (a)). If all the trees are represented with the simplest model(Figure 3.16(b)), it costs about 0.0625s per frame, but the realism is poor.



(a) Forest rendering with most detailed.



(b) Forest rendering with simplified models.



(c) Forest rendering with LOD models.

Fig. 3.16 Forest rendering with real time shadows.

Forest Rendering

Several techniques are employed to improve the rendering performance, such as clipping tree operation, vertex buffer objects. Since it is unnecessary to render trees outside the current camera frustum, clipping operation for each tree is used to cull those outside the frustum. Eight corners of each model's bounding box are projected from the object coordinates to the viewport coordinates. If neither of the projected points is in the window's viewport space, the tree will not be rendered. This approach is effective to cull trees far away from the camera. However, it will cause popping up and popping out artifact such that the trees near the viewer are unseen. To fix these popping effects, two additional points are checked while culling. One is the center of the front face of the bounding box and the other isthe back face's center. If any projection of the four points (including the bounding box's left bottom and the right top points) is in the window's viewport, the tree will be rendered. Another restriction is the distance. A clipping distance threshold μ is set which is not too large. Trees with distances to the viewer less than μ will be rendered no matter whether they are in the camera frustum or not. Although a few trees outside the camera frustum could be rendered, it brings little overhead to the overall performance. The clipping operation isemployed in both view frustum before rendering and the subdivided light frustums when generating the shadow maps. The clipping operation is based on the bounding boxes of the trees. Taking full advantage of the GPU capacity, the tree models' vertices, normals and texture coordinates are organized into vertex buffer objects. Vertex buffer object is an OpenGL extension. It provides an interface that allows the array data to be stored in high performance graphics memory, thereby promoting an efficient data transfer and avoiding repetitive calls of graphics functions. This technique dramatically enhances the performance. LOD are used for tree rendering based on distances to the viewer. Four levels of detail can be made for each tree species to reduce the rendering burden. In the implementation, the finest LOD is made up of 6000 - 8000 triangles and the coarsest LOD consists of 800 - 1100 triangles. The detailed data processing flow of the rendering system is shown in Figure 3.17.

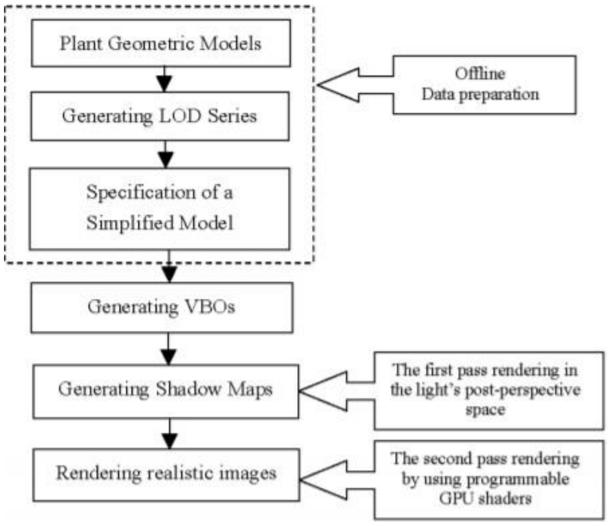


Fig. 3.17 Data processing flow chart

The techniques presented to render large forest scenes consists of tens of thousands of highly detailed trees at interactive frame rates, even with a realistic real-time shadowing effect. Close-up viewing for trees, walk-through and flyover a forest are available. It can be applied easily to the video games and interactive visualization.