

Real-Time Tree Rendering

I. Remolar, C. Rebollo, M. Chover, and J. Ribelles

Departamento de Lenguajes y Sistemas Informáticos,
Universitat Jaume I, Castellón
{remolar, rebollo, chover, ribelles}@uji.es

Abstract. Interactive rendering of outdoor scenes is currently one of the most important challenges in computer graphics. This article presents a new method of real-time visualization of trees and plants that combines both multiresolution modeling techniques and the dynamic generation of impostors. In our method, trees and plants are represented by continuous view-dependent levels of detail. This enables us to represent many complex tree models with variable resolution. The number of primitives rendered per tree is reduced according to their importance in the scene without loss of leafiness. Furthermore, trees are visualized using dynamic impostors that take advantage of the frame-to-frame coherence inherent in tree-dimensional scenes. The impostors avoid the need to redraw all the geometry of the scene continuously. This method permits visualization of outdoor scenes with a high number of trees in interactive applications such as computer games or virtual reality, adapting the level of detail to the capability of graphic systems.

1 Introduction

Rendering outdoor scenes with a great number of trees or plants cannot be accomplished in real-time with present graphic hardware. Many current interactive applications such as flight simulators, virtual reality environments or computer games happen in outdoor scenes and one of the essential components in these scenes is the vegetation. Tree modeling has been widely investigated [1], [2], and very realistic representations have been demonstrated. However, tree models are formed by such a vast number of polygons that real-time visualization of scenes with trees are practically impossible.

Various solutions have been researched for interactive visualization of complex models. Multiresolution modeling [3] is one of them. It makes possible the increase or reduction of the number of polygons in a geometric model according to their importance in the scene. Another approach is image-based rendering approximations [4],[5],[6] which substitutes some of the geometry of the scene by images textured on polygons. In this way, the models are represented in two dimensions. When the observer moves toward the object, the lack of details causes the realism of the scene to be lost. Point-based rendering is used in order to achieve interactive visualization of complex objects and is based on substituting the rendering primitive triangle mainly

by points [7],[8],[9]. But if the trees situated very close to the observer are rendered with points or lines, the details of the leaves can not be appreciated.

This paper presents a general scheme for rendering outdoor scenes with vegetation in real-time. It combines both multiresolution modeling and image-based rendering techniques. Our approach allows us to represent both near and distant trees with a high level of detail. The geometry of the trees is represented by means of dynamic view-dependent levels of detail. In this way, the number of the rendered polygons vary depending on the importance of the tree in the scene. The used multiresolution schemes accept a polygonal description of the tree, not exclusively restricted to the generation with L-systems.

Furthermore, in order to make possible real-time rendering, dynamically generated impostors [10] are combined with the geometric representation. These impostors have been successfully used in real-time cloud rendering [11]. They used to replace distant geometric models in the past, but in our method they are also used to represent some of the closer parts of nearby trees. This approach avoids redrawing the entire scene geometry in every frame because they can be re-used over multiple ones.

2 Previous Work

Research aimed at vegetation can be divided in two major fields: the generation of plants and trees, and their visualisation. Vegetation modelling has been explored extensively. The most important works in this field are Lindermayer-systems [1], used for generating realistic models of trees. Other solutions combine grammar based modelling with a graph description [2]. Apart from the great number of works that have appeared in the literature, some commercial applications have been developed for modelling trees. Some of the most important are OnyxTree (www.onyxtree.com), AMAP (www.bionatics.com), Xfrog (www.greenworks.de) and SpeedTreeRT (www.idvinc.com).

The analysis of previous work related with our approach can be divided as geometric representation and image-based rendering methods.

Geometric representation: Level of detail rendering [5] is one of the most popular methods to reduce the complexity of polygonal data sets in a smart manner.

The continuous multiresolution models presented thus far deal with general meshes and cannot be applied effectively to such scenes. Hoppe [13], Xia [14] and El-Sana [15] use methods based on the union of pairs of vertices in the construction process simplification. Luebke [16] uses a method based on vertex clustering: a set of vertices is collapsed into one vertex. These methods can not process the foliage without degradation of appearance [9].

Another technique in interactive visualization of complex plant models uses point-based rendering based on the idea of substituting the basic primitive triangle by points or lines. Reeves and Blau [17] rendered trees using small disks representing the foliage, and Weber and Penn [18] used sets of points for the leaves and lines for the tree skeleton. Stamminger and Dettrakis [8] visualize plants with a random sample set of

points. One of the most recent works in this field has been presented by Deussen et al. [9]. Their approach combines geometry with points and lines.

Image-Based Rendering methods: Billboarding is one of the most frequently used techniques due to its simplicity. The trees are reduced to images textured on polygons, which always maintain their orientation towards the observer. However this technique has great deficiencies, because the models are represented in two dimensions. When the observer moves toward the object, the lack of details produces a loss of realism in the scene.

Layered depth images [19], LDI, store in each pixel of the image a 2D array of depth pixels. In each depth pixel are stored, in proximity order to the point of view, the surfaces that appear in that image. But the LDI files created for trees are excessively large. Another similar method however using Z-buffers is presented by Max [4].

Jakulin [5] presents a method based on images with alpha-blended textured polygons. Lluch et al. [6] present a method based on a hierarchy of images obtained from pre-processing the botanical tree structure (a L-system) and storing the information in a texture data tree.

3 System Overview

The trees used in our study were modelled by the Xfrog application [2]. They are very realistic, but generally consist of more than 50.000 polygons each. This is a disadvantage when it comes to generating images in an interactive way.

Trees can be separated in two different parts:

- the solid component of the tree, the trunk and the branches, and
- the sparse component, the foliage.

In this work, two different multiresolution schemes have been used to represent each of the parts. The trunk and branches are represented by a set of connected surfaces, usually triangular meshes. There are many multiresolution models in the literature that deal with this kind of mesh. In this work the Multiresolution Ordered Meshes “*MOM*” [20] has been used to model them. An efficient implementation of the View Dependent Multiresolution Model for the Foliage, “*VDF*” [21], is used for the crown representation.

In order to construct a multiresolution model, a simplification method must be used. Qslim [22] is used to obtain the sequence of vertex simplification required by *MOM*. *VDF* is created from a sequence of leaf collapses determined off-line by the simplification algorithm called Foliage Simplification Algorithm, “*FSA*” [12]. The key to this simplification algorithm is *leaf collapse*. Two leaves are transformed into a single one, so that the area of the new leaf is similar to the area formed by the two leaves initially. This reduces loss of appearance at the end of the process. An error function is used to determine which pair of leaves will be simplified to create a new one.

The data obtained are stored once and many representations of this object can be visualized by instancing. A distribution editor is used to distribute every one of this

instances in a scene, in order to obtain a tree population. Fig. 1 displays the system outline.

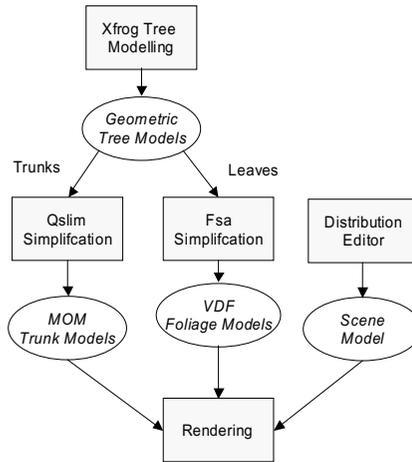


Fig. 1. System Overview

The tree instances are visualized with variable resolution depending on some criteria, such as distance from the observer or importance of the object in the scene. Furthermore, in order to increase the frame rate of the scene, dynamically generated impostors [10] are automatically combined with geometry. The zone nearest to the viewer will be represented by geometry, and the zone farther away, by an impostor. This impostor moves backwards when the viewer zooms in, disappearing when the observer is very close to the tree. In this case, the tree is represented only by geometry. On the contrary, when the viewer is moving away from the object, the impostor moves towards until the tree is represented only by the impostor.

Multiresolution modelling allows us to reduce the generation time of the impostors because the trees are visualized with less number of polygons in accordance to the importance in the scene.

4 Rendering Algorithm

Dynamically generated impostors have been combined in our rendering algorithm with multiresolution modelling in order to obtain real-time rendering of scenes. Impostors have been mainly used for objects situated far from the viewer. In our scheme, they are also used for close objects. Another novelty is that we use movable impostors: each one can move from the front part of the tree to the back (Fig 2). When it is situated in the front part of the foliage, tree is represented only by a textured image, and when the impostor is getting the back limit of the foliage, only by geometry. The movement of the impostor is conditioned on the distance to the observer. As the viewer moves towards the tree, the impostor moves backwards. The part of the tree

nearest the viewer is represented by geometry, and the rest of it, textured on the impostor.

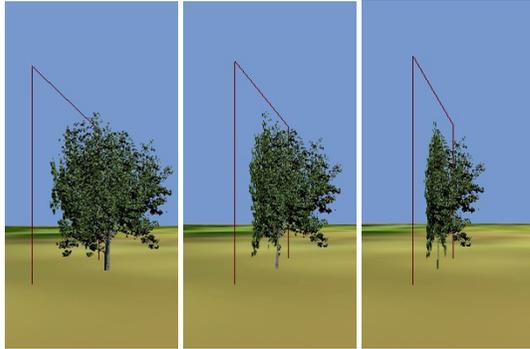


Fig. 2. Example of movable impostor

As far as the use of the dynamic impostors and the resolution variable in the tree, it is necessary to determine when they are no longer valid. We have used the same error measures used by Harris and Lastra in their work on real time clouds visualisation [11]. An impostor stops being valid when the tolerance of one of the following errors is surpassed:

Translation Error. This measures the validity of the representation in transferring movements of the observer. This error is calculated taking the angle that forms the present position of the observer with the position when the impostor was generated.

Resolution Error. This measures the validity of the resolution of the texture of the impostor. It is calculated by the following equation:

$$\text{resTexture} = \text{resImage} \times \frac{\text{objSize}}{\text{objDistance}} \quad (1)$$

While the impostors are valid, they are rendered as billboards in order to avoid unnecessary popping effects.

5 Results

The presented work has been implemented with OpenGL on a PC with Windows 2000 operating system. The computer is a dual Pentium Xeon at 1.8GHz. with a graphical processor NVIDIA Quadro2 with 64MB.

The trees used in our experiments is formed by 88.443 polygons. Their foliage were initially formed by 20.376 leaves, that is, 40.752 triangles and their trunks by 47.691 polygons.

The tests measure the frames per second in a scene where the number of trees is increased. The camera follows a random path in this scene. The trees are rendered using three methods. Fig 3 shows the results for these test.

- *Geometry*: every tree in the scene is represented with the maximum level of detail.
- *Multiresolution Modelling*: the level of detail of the tree is adapted to its importance in the scene. Trees close to the viewer are represented with high detail and it degrades as the observer is moving backwards.
- *Impostors + Multiresolution*: Trees are rendered using both multiresolution modelling and dynamically generated impostors.

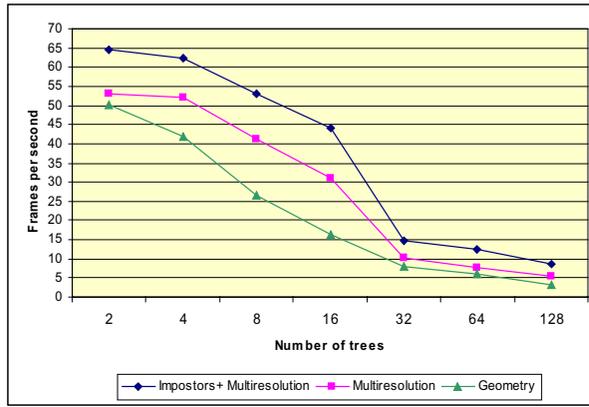


Fig. 3. Results of the experiments

As we can see in Fig 3, multiresolution modelling increases the frame rate. This is because the number of polygons that are drawn diminishes without reducing the realism of the scene. This chart shows that frame rate improves remarkably via the use of impostors. This allows us to render a scene with a high number of trees in outdoor scenes. In Fig 4, a scene of our test is shown, where the trees are represented by impostors. Fig 5 shows an example of the detail that can be obtained with our rendering method.

6 Conclusions and Future Work

In this paper, we have presented a system for the realistic visualisation of trees and plants in real time. The developed technique combines two suitable methods of acceleration for current graphics hardware: multiresolution modelling techniques and dynamically generated impostors.

The main contributions of our work can be summarized as:

- A general framework for the representation of outdoor scenes with trees. The method accepts a polygonal description of the tree, not exclusively restricted to the generation with L-systems.

- Efficient implementation of *VDF*, a multiresolution model specifically designed for foliage. It supports view-dependent visualization and multiples instances of a tree model can coexist.
- Use of movable, dynamically generated impostors both for distant and near trees. These are combined with geometry without producing artifacts. The management of the texture memory is optimized using this type of impostors. In our approach no images of different views of the tree are stored in memory as a part of preprocessing, as is the case in other methods [6].
- This general framework can easily be extended to other modeling representations such as particle systems.

The next step to improve the realistic representation of the trees, is to take account of illumination. We are developing solutions based on the use of light maps for trunks and terrain and per-vertex lighting for leaves.

Furthermore, the visualisation of scenes with multiple trees requires occlusion culling methods and multi-layered impostors.

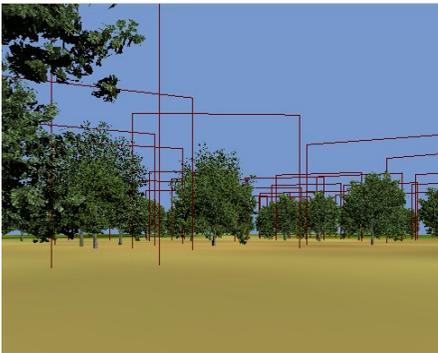


Fig. 4. Scene where impostors are visualized



Fig. 5. Detail of the trees

Acknowledgements. This work has been supported by the Spanish Ministry of Science and Technology grants TIC2001-2416-C03-02 and TIC2002-04166-C03-02, the Fundació Caixa-Castelló Bancaixa grant P1-1B2002-12 and the Integrated Action HU2000-0011.

References

1. P. Prusinkiewicz, A. Lindenmayer, “The algorithmic beauty of plants”, Ed. Springer-Verlag, New York, 1990.

2. B. Lintermann, O. Deussen. "Interactive modeling of plants", IEEE Computer Graphics and Applications, vol. 19(1), 1999.
3. J. Ribelles, A. López, Ó. Belmonte, I. Remolar, M. Chover. "Multiresolution Modeling of Arbitrary Polygonal Surfaces: A Characterization", Computers & Graphics, 26(3), pp. 449-462, 2002.
4. N. Max, K. Ohsaki. "Rendering trees from precomputed Z-buffer views". Eurographics Workshop on Rendering 1996, pp. 165-174, 1996.
5. A. Jakulin. "Interactive Vegetation Rendering with Slicing and Blending". Eurographics'2000, Short presentations 2000.
6. J. Lluch, E. Camahort, R. Vivó. "An Image-Based Multiresolution Model for Interactive Foliage Rendering", Journal of WSCG04, vol. 12(3), pp. 507-514, 2004.
7. J. Weber, J. Penn. "Creation and rendering of realistic trees", Proc of SIGGRAPH'95, 1995, pp. 119-128.
8. M. Stamminger, G. Drettakis. "Interactive sampling and rendering for complex and procedural geometry", Rendering Techniques 2001, Eurographics, Springer-Verlag, pp. 151-162, 2001
9. O. Deussen, C. Colditz, M. Stamminger, G. Dettrakis. "Interactive Visualization of Complex Plant Ecosystems", Proc. of the IEEE Visualization Conference, 2002.
10. G. Schaufler, "Dynamically Generated Impostors", GI Workshop, Modeling - Virtual Worlds - Distribute Graphics 1995, pp. 129-136, 1995.
11. M. J. Harris, A. Lastra, "Real-Time Cloud Rendering", Eurographics'2001, vol. 20(3), 2001.
12. I. Remolar, M. Chover, O. Belmonte, J. Ribelles, C. Rebollo, "Geometric Simplification of Foliage", Eurographics'02 Short Presentations, pp. 397-404, 2002.
13. H. Hoppe, "View-dependent refinement of progressive meshes", Proc. of SIGGRAPH'97, pp. 189-198, 1997.
14. J. Xia., A. Varshney, "Dynamic view-dependent simplification for polygonal models", Proc. of IEEE Visualization'96, pp. 327-334, 1996.
15. J. El-Sana, A. Varshney, "Generalized View-Dependent Simplification", Eurographics'99, pp. 131-137, 1999.
16. D. Luebke and C. Erikson, "View-Dependent Simplification of Arbitrary Polygonal Environments", Proc of SIGGRAPH'97, pp. 202-210, 1997.
17. W. T. Reeves, R. Blau. "Approximate and probabilistic algorithms for shading and rendering structures particle systems", Computer Graphics, Proc. of SIGGRAPH'85, vol. 19, pp. 313-322, 1985.
18. J. Weber, J. Penn. "Creation and rendering of realistic trees", Proc. of SIGGRAPH'95, pp. 119-128, 1995.
19. J. W. Shade, S. J. Gortler, L. He, R. Szeliski. "Layered depth images", Proc. of SIGGRAPH'98, pp. 231-242, 1998.
20. J. Ribelles, A. López, O. Belmonte, I. Remolar, M. Chover. "Variable Resolution Level-of-detail of Multiresolution Ordered Meshes" , Proc. of 9-th International Conference in Central Europe on Computer Graphics, vol. 2, pp. 299-306, 2001.
21. I. Remolar, M. Chover, J. Ribelles, O. Belmonte. "View-Dependent Multiresolution Model for Foliage", Journal of WSCG03, vol. 11(2), pp. 370-378, 2003.
22. M. Garland, P.S. Heckbert, "Surface simplification using quadric error metrics"; Proc of SIGGRAPH'98, pp. 209-216, 1998.