Informe Técnico
DLSI 1/3/2002

Real-Time Tree Rendering

Inmaculada Remolar, Miguel Chover, Óscar Belmonte, José Ribelles, Cristina Rebollo

Departamento de Lenguajes y Sistemas Informáticos,

Correo electrónico: remolar@uji.es

Departamento de Lenguajes y Sistemas Informáticos,
Universitat Jaume I
Campus de Riu Sec
E-12080 Castellón, Spain

Real-Time Tree Rendering

Inmaculada Remolar, Miguel Chover, Óscar Belmonte, José Ribelles, Cristina Rebollo

Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I
Campus Riu Sec
E-12080 Castellón, Spain

E-mail: remolar@uji.es

Abstract:

One of the most important challenges in real-time rendering of outdoor scenes is the representation of vegetation. This is due to the vast amount of polygons that are used to model vegetable species. This paper presents a new method of real-time visualisation of trees and plants that combines both the dynamic generation of impostors and multiresolution modelling techniques. Trees are rendered using dynamic impostors that take advantage of the frame-to-frame coherence inherent in three-dimensional scenes. Impostors avoid the need to redraw all the geometry of the scene continuously. Furthermore, the trees and plants are represented by multiresolution modelling. In this way, the level of detail can be adapted automatically and the cost of impostor generation will be reduced. This method deals independently with the trunk and the crown. The trunks are represented by a general multiresolution model. A special multiresolution model and an automatic simplification algorithm have been designed for foliage representation. This method permits visualisation of outdoor scenes with a great number of trees in interactive applications such as computer games or virtual reality, adapting the level of detail to the capability of graphics systems.

Keywords:

Vegetable species, real-time rendering, multiresolution modelling, dynamic impostors.

Visualización de Arboles en Tiempo Real

Inmaculada Remolar, Miguel Chover, Óscar Belmonte, José Ribelles, Cristina Rebollo

Departamento de Lenguajes y Sistemas Informáticos
Universitat Jaume I
Campus Riu Sec
E-12080 Castellón, Spain

E-mail: remolar@uji.es

Resumen:

Uno de los problemas más importantes en la visualización de escenas exteriores, es la representación de especies vegetales. Esto es debido a la gran cantidad de polígonos que se utilizan para modelar esta clase de objetos. En este informe se presenta un nuevo método para la visualización en tiempo real de árboles y plantas, que combina tanto la generación dinámica de impostores, como las técnicas de multiresolución. Los árboles se visualizan mediante impostores dinámicos, que aprovechan la coherencia entre secuencias consecutivas, inherente en las escenas tridimensionales. Los impostores evitan redibujar continuamente toda la geometría de la escena. Además, los árboles y plantas se representan mediante modelos multiresolución. De esta forma, el nivel de detalle puede adaptarse automáticamente y el coste de la generación de impostores se reduce. Este método trata el tronco y la copa del árbol de forma independiente. Los troncos se representan mediante un modelo multiresolución general. Sin embargo, para el follaje del árbol se ha diseñado un modelo multiresolución especial y un algoritmo de simplificación automático. Este método permite la visualización de escenas al aire libre con un gran número de árboles en aplicaciones interactivas como los juegos por ordenador o la realidad virtual, adaptando el nivel de detalle a la capacidad del sistema gráfico.

Palabras clave:

Especies vegetales, visualización en tiempo real, modelado multiresolución, impostores dinámicos.

Real-Time Tree Rendering

I. Remolar, M. Chover, Ó. Belmonte, J. Ribelles, C. Rebollo

Departamento de Lenguajes y Sistemas Informáticos

Universitat Jaume I, 12080 Castellón, Spain
e-mail: remolar@lsi.uji.es
Phone: +34-96-472-83-15, Fax: +34-96-472-84-35

1. Introduction

Many of the current interactive applications such as flight simulators, virtual reality environments or computer games take place in outdoor scenes. One of the essential components in these scenes is the vegetation. The lack of trees and plants can detract from their realism. Tree modelling has been widely investigated [1,2], and its representation is very realistic (Figure 1). However, tree models are formed by such a vast number of polygons that real-time visualisation of scenes with trees is practically impossible.

Real-time visualisation of vegetable species has not been extensively explored. Some applications utilise easy solutions such as using a tree image to texture two intersected quadrilaterals. One of the most commonly used techniques is billboarding [3], where trees are represented by a single textured quadrilateral, always oriented toward the viewer. Although there are more complex techniques, most of the applications utilise image-based rendering approximations. If, in some applications, geometry is required, this has been simplified beforehand. The images obtained with these methods are still not too realistic and for this reason it is necessary to introduce new solutions.

The new rendering method presented in this article allows real-time visualisation of scenes with trees modelled by a vast number of polygons. The trees used in our study are modelled by the Xfrog application [2] (Figure 1). They are very realistic, but are generally formed by more than 50.000 polygons. In our work, tree rendering is chiefly accomplished by dynamic impostors [4] that can be re-used over multiple frames. This method avoids the need to redraw all the scene geometry every frame. Dynamically generated impostors were used to replace geometric models in the past and currently they are successfully used in real-time cloud rendering [5]. In order to reduce the generation time of the impostors, multiresolution modelling techniques have been used [6,7].

Figure 1: *Aesculus hippocastanum. Tree modelled with the commercial modelling tool Xfrog. 192.179 triangles*

With these techniques it is possible to increase or to diminish the number of polygons in a tree, according to its importance in the scene. A tree situated far from the viewer is represented by a low number of polygons, so the impostor generation time decreases. Furthermore, in order to increase the realism of the scene, impostors are automatically combined with geometry when the trees are very close to the viewer. The nearest zone to the viewer will be represented by geometry, and the zones further in the background, by an impostor.

With this schema, it is possible to adapt the number of polygons drawn to the requirements of the graphics hardware. The information sent to the graphics processor remains constant, either by drawing the trees with fewer polygons or by re-using the impostors during more frames.

After reviewing of previous work in section 2, section 3 presents briefly our method for real-time rendering of trees. The multiresolution model specially designed for this kind of 3D models is studied in depth in section 4. In section 5, the visualisation algorithm is analysed. This section shows how we combine multiresolution modelling with the dynamically generated impostors. Section 6 presents the results and some implementation details. Finally, these results are analysed and some ideas for future research are discussed.

2. Previous Work

The research conducted about vegetation, could be divided into two broad fields: the generation of plants and trees, and their visualisation. Vegetation modelling has been extensively explored. 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 traditional techniques [2]. Apart from the great number of studies that have appeared in the literature, some commercial applications have been developed for modelling trees. Three of the most important are OnyxTree (www.onyxtree.com), AMAP (www.bionatics.com) and Xfrog (www.greenworks.de).

In contrast, real-time vegetation rendering continues to be a problem at present. Billboarding is one of the most widely used techniques [3] due to its simplicity. The trees are reduced to images textured on polygons that always maintain their orientation towards the observer. But this technique has important shortcomings, since 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.

Another solution consists in reducing the geometric complexity of the trees, but we are not aware of automatic simplification techniques [8] for this kind of polygonal models. Simplified versions of trees and plants can be obtained, in the case of using L-systems, by limiting the number of polygons at the time of generating the object. A discrete multiresolution model can be constructed with several independent representations of the same tree with different levels of detail, similar to node LOD of VRML or OpenInventor. During the rendering, the different instances of the tree are changed depending, for example, on the distance from the observer. This produces popping effects that can be solved with blending techniques. The main disadvantage of the discrete multiresolution models is their high storage cost.

Schaufler [9] proposes an image-based rendering technique. Layered depth images, LDI, store in each pixel of the image a 2D array of depth pixels. The surfaces that appear in that image are stored in each depth pixel, in proximity order to the point of view. But the LDI files created for trees are excessively large. Another similar method but using Z-buffers is presented by Max [10]. Marshall et al. [11] propose a multiresolution representation of botanical scenes integrating polygonal representations of large objects with tetrahedron approximations of the less representative parts of the scene.

Jakulin [12] presents a method based on images with alpha-blended textured polygons. Trees are divided in two parts: the solid part or trunk, and the dispersed part or crown and branches. The trunk is dealt with using a general simplification method of meshes. For the correct visualisation of the crown, several sets of slices must be pre-calculated. When the tree is finally visualised, the two sets of images nearer to the point of view of the scene are rendered, varying their level of transparency. This method does not support aerial views, and has some failures in the visualisation, such as the lack of union of the branches.

One of the most recent methods [13] uses a discrete multiresolution model for leaf rendering. The authors propose a hierarchy of images obtained through a pre-process of the generation of leaves. The model adapts the level of detail, extracting the appropriate set of images according to the position and direction of the observer. The transition from one level of detail to another is achieved by blending. The fundamental problem of this technique is that the transitions between levels of detail are not continuous. Another problem is the overload of textures.

3. Our approach

One of the latest techniques used in the rendering of complex objects is the dynamic generation of impostors [9, 5, 12]. The complex objects that form a scene are rendered in images and replaced by impostors. These impostors are polygons and the images obtained are textured on them. This technique considerably reduces the number of polygons needed

to visualise the scene by exploiting the frame-to-frame coherence. The images are re-used over multiple frames while a certain tolerance error is not exceeded. This process is supported by current graphics hardware and the requirements of texture memory are minimum. Only one texture is necessary per object and its resolution decreases with the distance. The problem arises when these impostors require frequent updates.

The trees used in this work have been generated with the commercial application Xfrog [2]. The trees modelled are very realistic, but are geometrically very complex. This is a disadvantage at the time of generating images in an interactive way. One solution to this problem is multiresolution modelling. This technique allows us to adapt the number of polygons of the model according to its importance in the image and to generate the impostor afterwards.

Some authors [6,7] have classified multiresolution models into basically two groups: discrete and continuous. Discrete models contain a finite number of levels of detail and a control mechanism to determine which is the most adequate in each moment. On the other hand, the continuous models capture a vast range of approximations of an object, virtually continuously. The discrete multiresolution models present a series of disadvantages. There is no relation between the stored levels of detail, so the size of these models increases quickly when some new levels of detail are included. The continuous models offer the possibility of adapting the level of detail in real time. This accelerates the visualisation. Some of them can represent an object with variable resolution, that is, different resolutions can coexist in different regions of the rendered object.

In this work, a specific multiresolution model for trees has been designed. This is a continuous model that allows us to represent the foliage with variable resolution. The multiresolution models appearing in the literature deal with general polygonal meshes. The leaves of the trees are modelled, not by a mesh, but as a set of independent polygons. The existing multiresolution models do not work properly for this reason.

Our solution for the interactive visualisation of trees and plants combines two methods of real-time rendering: the dynamic generation of impostors and multiresolution modelling techniques. The images generated in previous frames can be re-used due to the frame-to-frame coherence. The use of a special multiresolution model for trees allows us to adapt the number of polygons, depending on the importance of the tree in the scene. This reduces the generation cost of the impostors that represent trees situated farther away from the viewer. Moreover, the variable resolution of our model makes it possible to represent, with more or less detail, certain zones of the tree. For example, the central hidden leaves of the foliage can be represented with a lower resolution, while a much higher one is used to represent the edges. When the distance is such that the observer can appreciate the details of the object, the impostors will be combined with geometry. The back part of the trees will be represented with less detail and by an impostor, whereas the front part will be shown only with its geometry. Trees have a three-dimensional look that does not exist in the classic techniques of image-based rendering.

4. Multiresolution model of the foliage

The trees can be separated into two different parts: the solid component of the tree, the trunk and the branches, and the foliage or leaves. The trunk and the branches are represented by triangle meshes and each of the leaves, by an image of a leaf textured in a quadrilateral.

In this work, each of these parts has been treated in a different manner. The trunk is formed by a set of meshes of polygons. A great number of multiresolution models existing in the literature represent this type of objects [15,17,18]. The model that has been used in this case is Multiresolution Triangle Strips. This is a model for general meshes developed by the authors themselves. It is the first model appearing in the literature that uses the primitive strip as the base of the data structure and not just in the visualisation process, as happens in other models [19]. Once the appropriate level of detail has been determined, the trunk is drawn with strips. The use of this primitive allows us to send a smaller amount of information to the graphics system, leading to a decrease in visualisation time.

On the other hand, the foliage of the tree is formed by a set of independent polygons. The multiresolution models that have appeared up to now do not work properly with this type of representation. The majority have been defined by meshes, not by isolated polygons. We have designed a multiresolution model that allows us to represent the foliage with different levels of detail, and the data structure and the extraction algorithm of the required level of detail have been defined. The proposed data structure has been inspired in View Dependent Refinement of Progressive Meshes, VDPM [20]. This is one of the most popular continuous multiresolution models for general meshes that support variable resolution. In this model, the transition from one level of detail to the next is made by collapsing two

vertices into one. This creates a father-children relationship that establishes a vertex hierarchy represented by binary trees.

In a similar way, our model collapses the polygons that represent two leaves into a single polygon, or leaf. As in VDPM, this conditions a hierarchical binary tree structure. The leaf nodes of this structure are the polygons representing the leaves of the tree with a maximum degree of detail, and the root nodes are the polygons needed to represent the foliage with a minimum amount of detail. In this way, the structure is made up of a "forest" of binary trees.

In order to construct the data structure of the model it is necessary to use a simplification method that determines the pair of leaves that will be collapsed. There is no automatic algorithm simplification in the literature for this kind of problem, so it was necessary to define one, the *Leaf Simplification algorithm, LSA.* The algorithm is described below.

4.1 Leaf Simplification algorithm

The leaves defined with the Xfrog application are represented by quadrilaterals formed by two triangles. The final appearance is obtained by texturing these quadrilaterals with the image of a leaf. LSA collapses two leaves at each iteration and obtains a new one, maintaining a similar area to that of the pair of collapsed leaves. This is done in order to preserve the appearance of the foliage of the tree when the number of leaves is reduced (Figure 2).

The simplification method is characterised by two elements: the measurement that specifies the cost of collapsing two leaves and the position of the vertices that form the newly created leaf.
Leaf Collapse Cost. Given a set of candidate leaves to be collapsed, a pair will be chosen so that the error function is diminished. This function combines distance and planarity between the pair of evaluated leaves. Assuming that l_1 and l_2 are two leaves pertaining to a certain level of detail, the error function is as follows:

$$e(l_1,l_2)=d_H(l_1,l_2)\times(k_1\times d_H(l_1,l_2)+k_2\times c(l_1,l_2))$$

where $d_H(l_1, l_2)$ is the distance between leaves, and $c(l_1, l_2)$ the planarity level. Measurement of the distance between leaves is according to *Hausdorff.* This measurement makes geometric comparisons between two sets of points using the shortest distance between a point x and a set of points Y:

$$d(x,Y)=min_{y\in Y}\mathrm{d}(x,y)$$

where $d(x, y)$ is a measurement between a pair of points. The distance of *Hausdorff* is defined as:

$$d(X,Y)=\frac{\sum_{x\in X}d(x,Y)+\sum_{y\in Y}d(y,X)}{|X|+|Y|}$$

Figure 2: *Results of the simplification algorithm. a) 20.376 leaves, b) 6.710 leaves, c) 779 leaves, d) 236 leaves. e) Decreasing the detail with the distance from the observer*

We also measured the planarity between two leaves. This was done by comparing the angles formed by the normal vectors of the leaves. But the distance between leaves is a more important criterion than planarity in our error function. Two constants have been used, k_1 and k_2 in the function. We have verified empirically that the best results are obtained with k_1 = 0.8 and k_2 = 0.2.

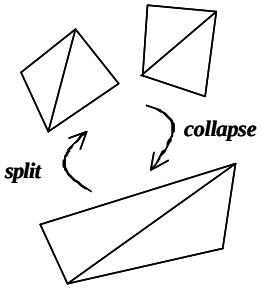


Figure 3: *Simplification of two leaves, creating a new one. The vertices of the original leaves remain.*

Vertex placement. The simplification algorithm LSA does not introduce new vertices in the model in order to avoid an overload in the data structure. The vertices of the new leaf will be two vertices of each of the collapsed leaves. For this reason, the two vertices of each leaf furthest from the other leaf would be chosen. This method will allow us to maintain an area similar to that of the two original leaves. The two triangles that will form the new leaf, generally speaking, are not in the same plane.

4.2 Foliage representation with variable resolution

In order to define a multiresolution model, it is necessary to establish the data structures to store the information, and the algorithms for accessing these structures in order to recover this information in the most efficient manner. These algorithms select the most appropriate level of detail in real time and following a criterion, such as distance from the observer or importance of the object in the scene.

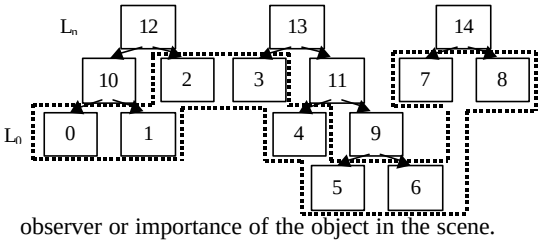


Figure 4: *Example of data structure of the multiresolution model of the foliage of a tree.*

The multiresolution model defined to represent the foliage of a tree is created from a sequence of collapses obtained from the simplification process. Each collapsed pair of leaves conditions a hierarchic relation. The node representing the new leaf is the father of the nodes of the collapsed leaves. The data structure is created bottom-up as a binary tree. In Figure 4, we show an example of a data structure. Let L_0 be the set of leaves forming the original foliage, in this case only 9 leaves. The root nodes correspond to the worst level of detail, L_n. Eventually, this level of detail would be represented by only 3 leaves.

Principal data structure. The main data structure is *Leaf* (Figure 5). It stores the vertices that make up a leaf, as well as the pointers necessary to maintain the tree structure -a pointer to the father leaf, and two pointers to the two daughter leaves.

```
struct Leaf{
   vertex leafVertex[4];
   leaf* father;
   leaf* left;
   leaf* right;
}
```

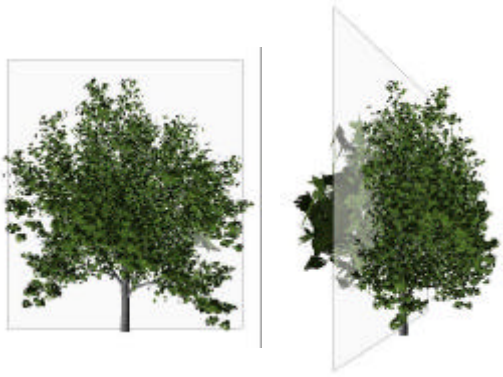
Figure 5: *Principal C++ data structure*

Selective refinement algorithm. The algorithm uses a list of active leaves L_s that enables us to know which leaves are to be visualised for the current level of detail. The basic idea consists in checking L_s, verifying that every leaf in this set is valid for the required level of detail. If this is not true, it is necessary to apply one of the following operations (Figure 3):

- *collapse*. The leaf is drawn in the worst detailed zone and it has to be collapsed.
- *split*. The leaf is not valid, because it is in the most detailed zone. It has to be refined.

The validity criteria of the leaves depend on the application. In the tests carried out, the leaves situated in the interior of the trees and far away from the observer are drawn with a worse level of detail than the leaves situated in the contours or next to the point of view. In a crown, different levels of detail can be maintained because the proposed model supports variable resolution (Figure 6).

In order to apply the *collapse* operation, it is necessary for the leaf not to be a root node. Next, it is verified that both this leaf and its sister are active, and neither of them are in the zone where maximum detail is required. If all these conditions are fulfilled, these two leaves will not be visualised in the following level of detail, and will be replaced by their father leaf. As regards the *split* operation, the active leaf must have sons and they have to be in the zone of the crown where more detail is desired. If this is so, this leaf will not be active and it will not be visualised in the following level of detail, being replaced by its daughters.



a) b)

Figure 6: *Example of variable resolution in the foliage, which is more detailed in front of the plane.*

The core of the traversal algorithm is summarised below in Figure 7.

```
for each leaf l ∈ Ls {

//COLLAPSE
if(not(Criteria(l))and(HaveFather(l))
{
  if(not(Criteria(sister)){
    ready = TRUE
    if(not isActive(sister))
      ready = ForceCollapse(sister)
  if ready collapse (h);

//SPLIT
nodoSplit = CheckSplit (l);
if (nodoSplit != NOTFOUND)
    forceSplit (l,nodoSplit);
}
```

Figure 7: *Pseudocode of the extraction algorithm of the required level of detail.*

The *Criteria* function is *TRUE* if the evaluated leaf belongs to the zone that must have more detail. If it is in the coarsest zone, the preconditions are evaluated in order to perform a collapse transformation. The *ForceCollapse* function is called when the sister of the candidate leaf to be collapsed is not active. In this case, some of its descendants are drawn in the current level of detail. This function checks its descending active leaves, and evaluate whether they are in the minimum detail zone. If they are, they will be collapsed. This process is repeated until the target leaf is left active. If any of the descending leaves are in the zone of interest, the function returns *FALSE*, indicating that the target collapse cannot be performed.

```
Procedure CheckSplit(l)
n= NOTFOUND;
if haveSon(l){
 if(Criteria(l.left)or Criteria(l.right)
    n=l.left
 else
    n=CheckSplit(l.left)
if (n == NOTFOUND)
    n=CheckSplit(l.right)
return (n)
```

Figure 8: *Pseudocode of the CheckSplit procedure.*

As regards the operation *split*, the *CheckSplit* function deals with the sub-tree formed by the descendants of the checked node *l*. The procedure evaluates its descendants until it finds a node *n* that is in the zone of interest. When this happens, the function *ForceSplit* refines all the descendants of node *l*, until the activation of this leaf *n* is achieved.

5. Rendering algorithm

Rendering outdoor scenes with a great number of trees or plants cannot be accomplished in real time with present graphics hardware. Some solutions have been studied. One of them is multiresolution modelling. If geometry is required, it is necessary to use techniques that allow us to reduce the number of polygons that are sent to the graphics system. Another solution consists in using image-based rendering techniques. In this line, dynamic impostors have been used [4]. They re-use information sent to the graphics hardware, in consecutive frames.

The visualisation algorithm presented in this paper combines both techniques of rendering acceleration. The use of multiresolution modelling techniques enables us to reduce the number of polygons drawn. The model used for the trees represents the trunks with a multiresolution model based on strips. The use of strips increases the drawing speed. The vertices sent to the processor to draw *n* triangles are *n+2*, but if they are drawn as isolated triangles *3n* vertices are required. In addition, the multiresolution model developed for foliage can represent variable resolution. The contours or the front parts of the crown will be represented with a better level of detail. Specifically, the number of leaves diminishes depending on:

- *The distance to the point of view.* The trunks and the leaves are represented with fewer polygons as they move away from the point of view.

- *An influence sphere.* This criterion is applied only to the crown in order to obtain variable

resolution. The surrounding sphere is taken and it moves backwards (Figure 9). All the leaves that are outside the sphere are drawn with maximum detail and it is progressively reduced towards its interior. Depending on the shape of the tree, other primitives can be used instead of the sphere.

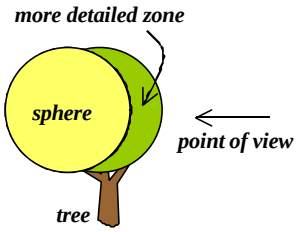


Figure 9: *Variable resolution criterion.*

With regard to the use of dynamic impostors, it is necessary to determine when they are no longer valid. We have used the same error measurements as those used by Harris and Lastra in work on real-time visualisation of clouds [5]. An impostor stops being valid when the tolerance of one of the following errors is surpassed:

- *Translation Error*. This measures the validity of the impostor in transferring movements of the observer. This error is calculated by 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:

$$resTexture = res\,\mathrm{Im}age \times \frac{objSize}{objDistance}$$

The fundamental advantage of impostors is that they work very well when the objects are far away. If we attempt to draw close objects with sharp edges, the impostors need frequent updating. The diffuse nature of the trees is ideal for representation with impostors.

In addition, the visualisation algorithm combines the impostors with geometry when the trees are too close. An image of the middle part of the tree is rendered and textured on an impostor. In front of it, the real geometry represents the front part of the tree. This geometry conceals the impostor, which prevents its updating from being perceived. In this way, a non-existent three-dimensional appearance in the classic techniques of image-based rendering is obtained.

6. Results

The method developed was implemented with OpenGL on a PC with Windows 2000 operating system. The computer is a dual Pentium III Xeon at 1.5GHz. with an NVIDIA Quadro2 Pro graphics processor with 64MB.

The trees used in our experiments are represented by 4.791 leaves, that is, 9.582 triangles. The tests render scenes of increasing numbers of trees. The camera follows a circular path around the scene.

Figure 10 shows the results for the movements of the camera. The chart shows the difference between using only geometry, only multiresolution and, finally impostors plus multiresolution.

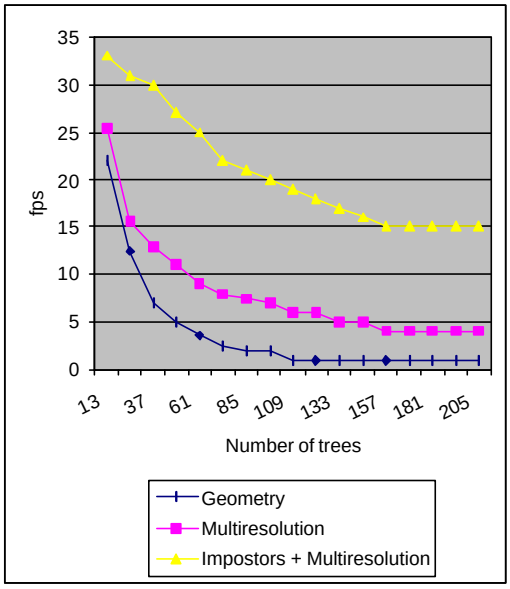


Figure 10: *Results of the experiments*.

As we can see in the graph, multiresolution modelling increases the frame rate. This is because the number of polygons that are drawn diminishes extraordinarily without reducing the realism of the scene. The chart shows that frame rate improves remarkably with the use of impostors. This allows us to render a scene with more than 200 trees with a frame rate of 15. In Figure 11, a scene from our test is shown, in which the trees are drawn by impostors. Figure 12 shows an example of approaching movement towards a tree. In this figure, we can appreciate the detail that can be achieved with our rendering method.

7. Conclusions and future work

In this paper, we have presented a system for the realistic visualisation of trees and plants in real time. The method accepts a polygonal description of the tree, not exclusively restricted to the generation with L-Systems. The technique developed combines two methods of acceleration suitable for use with the current graphics hardware: dynamic impostors and multiresolution modelling techniques.

The use of dynamic impostors exploits frame-to-frame coherence and reduces the number of polygons sent to the graphics system. Trees may easily be represented with impostors due to their diffuse nature. When the observer moves towards the object, they are only used to represent the rear side of the trees.

The multiresolution model of the trees allows us to adjust the level of detail to the application requirements. In this way, fewer polygons are used to draw distant objects. The remote trees are drawn with an impostor that has been generated with few polygons. When trees are close to the viewer, the front parts are drawn with geometry and the rear side with impostors. Furthermore, the model of the crown can show the crown at variable resolution.

We have presented a method of automatic simplification of foliage. This algorithm has been used to construct the multiresolution model and it can work in an independent way.

The next step to improve the realistic representation of trees is to take account of illumination. We are developing solutions based on the use of light maps. On the other hand, the visualisation of scenes with many trees requires the use of techniques such as hierarchical subdivision of scenes [21], occlusion-culling methods and multi-layered impostors [22]. Another possibility that could be analysed is the representation of animation effects on the trees, such as those produced by the wind [23].

For more information and colour images, please visit http://www3.uji.es/~remolar/trees

Acknowledgements

This work was supported by the Spanish Ministry of Science and Technology grants TIC1999-0510-C02-02 and TIC2001-2416-C03-02

References

1. P. Prusinkiewicz, A. Lindenmayer, “The algorithmic beauty of plants”, New York, Ed. Springer-Verlag, 1990.
2. B. Lintermann, O. Deussen. “Interactive modelling of plants”. IEEE Computer Graphics and Applications, 19(1), 1999.
3. T. McReynolds, D. Blythe. “Advanced Graphics Programming Techniques Using OpenGL”. Siggraph’99 Course, 1999.
4. G. Schaufler, “Dynamically Generated Impostors”, GI Workshop, Modelling - Virtual Worlds - Distribute Graphics 1995, pp. 129-136, 1995.
5. M. J. Harris, A. Lastra, “Real-Time Cloud Rendering”, Eurographics’2001, 20(3), 2001.
6. J. Ribelles, A. López, Ó. Belmonte, I. Remolar, M. Chover. "Multiresolution Modelling of Arbitrary Polygonal Surfaces: A Characterization", *to appear* in Computers & Graphics, 2002.
7. E. Puppo, R. Scopigno, “Simplification, LOD and Multiresolution – Principles and Applications”. Eurographics’97, Tutorial Notes, 1997.
8. P. Heckbert, M. Garland, “Survey of Polygonal Surface Simplification Algorithms”, Siggraph’97 Course Notes, 1997.
9. G. Schaufler. “Per-object image warping with layered impostors.” Eurographics Rendering Workshop 1998, pp. 145-156, 1998.
10. N. Max, K. Ohsaki. “Rendering trees from precomputed Z-buffer views”. Eurographics Workshop on Rendering 1996, pp. 165-174, 1996.
11. D. Marshall, D. Fussell, A. T. Campbell III, "Multiresolution Rendering of Complex Botanical Scenes", Graphics Interface '97.pp. 97--104, 1997.
12. A. Jakulin. “Interactive Vegetation Rendering with Slicing and Blending”. Eurographics’2000, Short presentations. 2000.
13. J. Lluch, S. Fernández, R. Vivó. “Multiresolution Model for foliage visualization in real time”. Technical Report II-DSIC-24/01, 2001.
14. K. H. Nielsen, N. J. Christensen. “Real-Time Recursive Specular Reflections on Planar and Curved Surfaces using Graphics Hardware, WSCG’2002. 2002.

15. H. Hoppe. “Progressive meshes”. Proceedings of Siggraph’96, pages 99-108. 1996.

16. Ó. Belmonte, I. Remolar, J. Ribelles, M. Chover, C. Rebollo, M. Fernández. "Multiresolution Modelling Using Connectivity Information", WSCG’2002, 2002.

17. Ó. Belmonte, I. Remolar, J. Ribelles, M. Chover, M. Fernández. "Efficient Implementation of Multiresolution Triangle Strips", *to appear in* Proc. of the International Workshop on Computer Graphics & Graphics Modelling, 2002.

18. J. Ribelles, A. López, Ó. Belmonte, I. Remolar, M. Chover. "Variable Resolution Level-of-detail of Multiresolution Ordered Meshes". WSCG’2001, Vol. 2, pp. 299-306, 2001.

19. J. El-Sana E. Azanli, A. Varshney. “Skip Strips: Maintaining Triangle Strips for View-dependent Rendering”, IEEE Visualisation '99, pp 131-138, 1999.

20. H. Hoppe, "View-dependent refinement of progressive meshes", SIGGRAPH'97 Proc., 1997, pag. 189-198.

21. J. Shade, D. Lischinski, D. Salesin, T. DeRose, J. Snyder, “Hierarchical image Caching for Accelerated Walkthroughs of Complex Environments”, Siggraph’96, pp. 75-82, 1996.

22. X. Decoret, G. Schaufler, F. Sillion, J. Dorsey, “Multi-layered impostors for accelerated rendering”. Eurographics’99, 18(3), 1999.

23. J. T .Barron, B. P. Sorge, T. A. Davis “Real Time Procedural Animation of Trees”. Eurographics’2001, Short presentations, 2001.

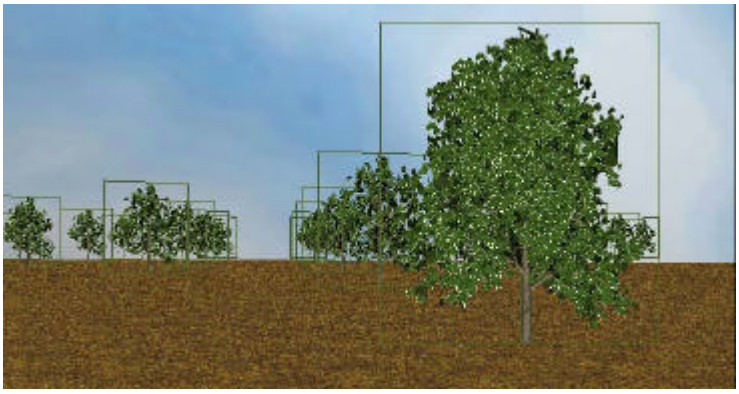


Figure 11: Scene with impostors**.**

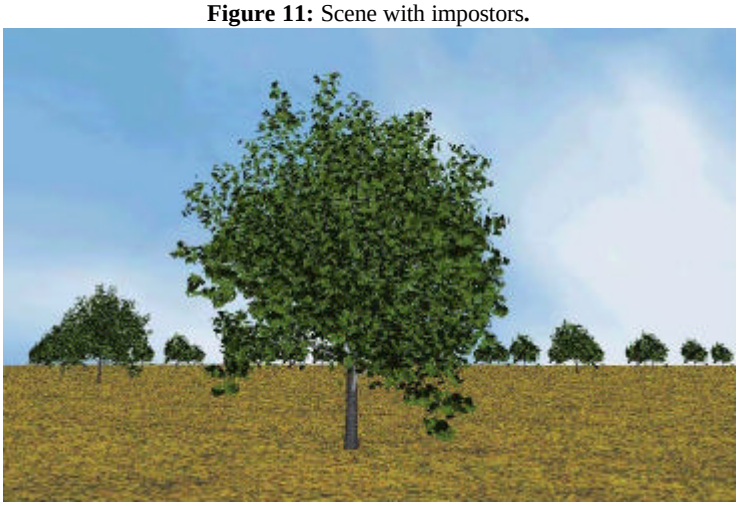


Figure 12: *Approximation movement towards a tree.*