

Fast Rendering Method of Lots of Dynamic Trees Based on OSG

Qingxuan Jia, Zuxin Wang, Jingzhou Song

¹Beijing University of Posts and Telecommunications Beijing 100876, China
{qingxuan & wangzuxin & sjz2008 }@bupt.edu.cn

Abstract - In virtual reality, the fast rendering of trees is always a difficult part of scene rendering, so it is a challenge to render a large scale of trees with rich geometric details. We present a new method to get fast and real rendering. The LOD technique and Billboard technique are both combined in applying in the rendering method based on OSG to realize the fast requirement, and GPU technique is adopted to realize the swinging effect, in order to be more real, shadow map technique is also utilized in the method. The method is proved to be feasible to simulate large scale of tree scene.

Index Terms – OSG, LOD, Billboard, virtual reality, GLSL

1. Introduction

OSG (Open Scene Graph) Graphics Engine is a kind of software engine based on OpenGL, which allows the programmers to write graphic programs with high performance and cross-platform quickly and conveniently^{[1][2]}. The requirement of fast and real rendering of virtual scenes in virtual reality is more and more sophisticated, owing to its high performance, OSG technique is widely applied in virtual reality field. 3D studio Maya is a popular software in 3D design and plays an important role in 3D modelling[3][4].

Fast scene rendering with high reality is a significant and challenging subject in virtual reality. Tree rendering determines the effect of the scene rendering to some degree. In the tree rendering process, it highlights the rendering time and the frame. As is known tree modelling usually uses polygons to build its shapes, so in the circumstance a lot of trees contain a number of polygons, such as a forest with 5000 trees can contains 1 billion polygons.

In this paper, the OSG terrain and detailed models are created by Maya software and exported as ANSIC file of OSG with osgEXP plug-in based on OSG (Open Scene Graph)2.8.2. Besides the billboard model is pre-processed from the OSG model in billboard technique. We present a new method to render the trees that integrates a set of techniques to render highly detailed trees with real-time shadows and swinging effect of the trees. Because a lot of trees contain many polygons, to get fast effect, we combine IBR and GBR method. We invent a scene management especially to manage different parameters including the parameters of LOD. Different OSG models with different details are transferred and the nearer the distance is, the more detailed model is used, and the most coarse model is a pre-processed billboard model. Then we apply an improved shadow mapping to achieve soft shadow effects. In the end, the GPU technique is adopted to realize the swinging effect of the trees on the basis of simplification of the wind in the directions.

2. Related Work

A lot of efforts have been done in the tree modelling and rendering because of the complexion and difficulty. In summary there exist two main directions: image-based rendering (IBR) and geometry-based rendering (GBR). Fig 1 shows the two main directions.

A. Modelling of the trees and the terrains

Maya is the 3D developing tool that enables the modelling of any terrains with the stages which are modelling, material, texture-mapping, and light. The modelling is shown in Fig 2

OSG constructs scene with tree structure to improve the efficiency of rendering and transforming models by callback functions which is convenient to achieve rendering effect[1].

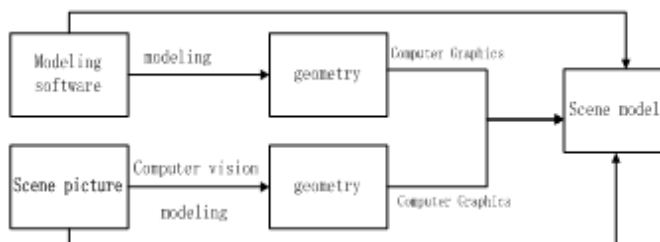


Fig.1 two main rendering methods

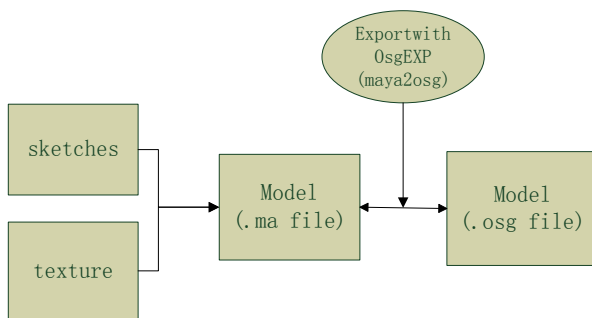


Fig.2 Modeling from Maya to OSG

B. Image-based rendering

Image-based rendering is a well-known approach used to render trees[5]. Imposter[6] and billboard[7] are the most sufficient methods of image-based rendering. Billboard technique theory is: a graph is drawn on a surface of a polygon to the view point, while the view point varies, the polygon changes its direction. In the rendering process, the new pose

and orientation of the panel with the camera can allow the rendering scene a 3D effect by a 2D texture. Billboard theory is shown in Fig 3. In the paper we process the models in billboard as is shown in Fig 4 and Fig 5.

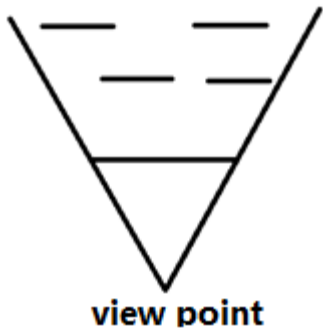


Fig.3 Billboard theory

C. Geometry-based rendering

Geometry-based rendering is a well applied approach to render tree[8]. LOD technique is very common in the geometry-based rendering. LOD is short for Levels of Details[9] Clark brought about in 1976. LOD technique is mainly adopted in rendering complex scenes quickly. Owing to the limitation of our visions, the requirements for the complexion of the models in different areas are quite different, therefore, we can determine whether the detailed model or the coarse model is loaded when we are loading the same scene model at different distance from the camera. LOD technique determines resource allocation of scene rendering, and it decreases the sides and the degree of detail of insignificant objects to improve the rendering efficiency as is shown in Fig 6. LOD0 loads the most detailed models, and then LOD1 loads the less detailed models and so forth.



Fig.4 An image



Fig.5 Billboard model

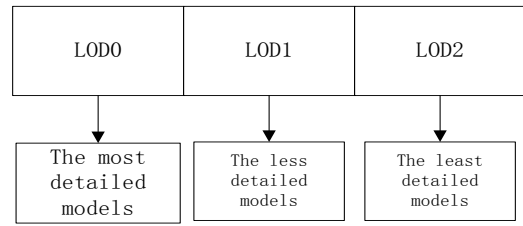


Fig.6 LOD determination

In this paper, we combine the billboard and the LOD to process the models. In order to get the perfect vision, we balance the reality and the fast request, so in the background we use the OSG models pre-processed by billboard, and in the view frustum we adopt the more detailed OSG models. And the distance between the trees and the camera is the criteria that distinguish whether the tree is in the view frustum or in the background. Besides the distance can be decided by us.

D. Shadow map technique

Shadow map[10] is a kind of shadow algorithm based on images. Its idea is as follows: There is an object in the shadow area, there for sure exists an obstacle between the object and the light. Therefore the shadow area is definite as long as the distance between the receiver and the light and the distance between the obstacle and the light are definite when rendering. Shadow map algorithm is based on depth buffer, so it can create shadows for any obstacle. With the graph processing nowadays, shadow map algorithm can calculate shadows efficiently. In the paper, first we adapt the shadow map algorithm to generate the dynamic shadows. But the aliasing problem is obvious in the shadow map sampling rates near the view-point is much higher than others and the aliasing problem occurs[7] as is shown in Fig 7. Many techniques have been exploited in the anti-aliasing resolution. One of the most popular methods is perspective shadow map (PSM) [11]. It generates more samples in the centre of the projection and fewer samples to the edge by create a perspective transformation by rendering the shadow map. Another popular approach is parallel-split shadow map (PSSM), which splits the view frustum into different depth ranges. The view frustum is splited by split planes paralleled to the view plane. And the shadow maps for them are rendered. Such we can get soft shadow map that is more realistic to the real world.

In this paper we adopt the improved perspective Shadow Map to render the shadows as shown in Fig 8.



Fig.7 aliasing shadow maps



Fig.8 improved shadow maps

E. GLSL tree swing based on GPU

GLSL is a high-level shading language of OpenGL based on C/C++, and contains three types of shaders including vertex shaders, fragment shaders and geometry shaders. The main use of GLSL is to work as an API of GPU to render the scenes efficiently[12]. The process of GLSL is shown in Fig 9.

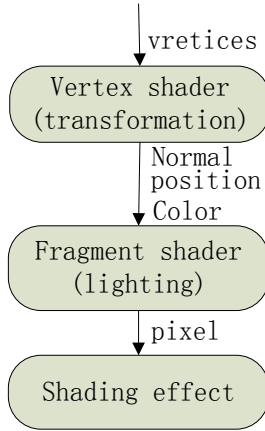


Fig.9 GLSL shading process

The traditional pipeline rendering process of the 3D graphs simplifies the developing processes, but the fixed processing modes and the inflexibility of the processing procedure lead to the result that it is difficult to achieve more difficult simulations for the scenes and it constrains the development for the scene graph. However in the GPU programming rendering, we substitute vertex shader for the transformation, the light calculation and the projection transformation in the pipeline and therefore it is easy to calculate the light, deal with the shadows and other calculations easily. GPU programming[13] can transfer all the vertices to the vertex shader to process in the same way parallelly and automatically due to its stream line data transfer. GPU provides the programmable rendering method for the pixel data. In fragment shader, multi-texture mapping, multi-scene mixing can be realized and more vivid rendering effects are also available. In GPU programs, GLSL is a convenient and easy language, so we adapt the GLSL to realize the swinging effect of the trees.

Wind is the cause to swing the trees, and wind consists of the wind direction and the velocity in mechanics[14]. The

vector of the wind at sometime can be decomposed into the sum of the average wind and the pulse wind as is shown in (1).

$$\vec{V} = \vec{V} + V_d \quad (1)$$

Wind has an effect on the trees mainly manifesting in the branches and the leaves bending and offsetting. The degree is determined by the direction and the velocity of the wind. In mechanical analysis, we decompose the instantaneous vector of the wind into the horizontal one (V_1) and the vertical one (V_2). The horizontal wind (V_1) is the cause that the trunk tilts to some degree in the horizontal direction, and the vertical wind is the reason that the branches move up and down in the vertical direction. Because the horizontal component of the wind is enough to simulate the real world and fast rendering of the trees in GLSL requires some simplification, only the effect of the horizontal component of the wind is calculated. For simplification, the direction angle and the tilt angle is generated under the horizontal component of the wind as is shown in Fig 10.

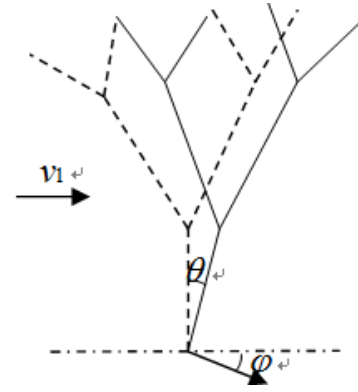


Fig.10 wind model

For the simplified mechanical model, the deformation equation is shown in (2):

$$\begin{cases} \varphi = \arccos\left(\frac{v_1[0]}{v_1}\right) \\ \theta = |v_1| \cdot area \cdot stiffness \cdot angleMax \end{cases} \quad (2)$$

In (2), $v_1[0]$ is one of the components of v_1 , $area$ is the area of the tree facing the wind, $stiffness$ means the stiffness coefficient of the trunk simply ranging from zero to one, when stiffness is zero, it means that there is no wind affecting the wind, while when the stiffness is one, it means that the wind has the most effect on the tree, leading to the result that the tilt angle of the tree reaches the expected maximum angle $angleMax$. $angleMax$ of different trees is different, and it is decided by experience. Owing to the determinacy of the angle φ , the achievement of rendering of θ is enough to meet the demand of the tree animation. In the real process of simulation, a parameter that can change with the time is enough to simulate to the motion of the tree swing.

In this paper, we ignore φ to simplify the swaying model to get real and fast rendering effect. The whole swaying effect is shown in Fig 11.



Fig.11 Swaying effect of the trees

3. Results

All the tests have been performed on an Intel (R) Xeon (R)W3530 with 6GB of RAM and a NVIDIA GeForce GTX 470 graphics card. C++ and OSG are the main engines for the rendering and GLSL is the main development tool of GPU programs to render the scene with more than 1000 trees.

Some results of the experiment are in the following tables:

The frame comparison of the LOD determination is shown in Table I:

TABLE I

	Full detailed models	Full billboard models
frame	43	60

The frame comparison of the shadow and the non-shadow effect is shown in table II:

TABLE II

	shadow	Non-shadow
frame	49	57

The frame comparison of the swaying trees and the non-swaying trees is shown in Table III.

TABLE III

	Non-swaying trees	Swaying trees
frame	57	47

Conclusions and Prospects

In this paper we combine the LOD and the Billboard techniques to alter the OSG tree models and the OSG-Billboard tree models dynamically. In this paper we bring about a simplified method of swaying tree rendering based on the GPU programming technique combined with the OSG technique. We adapt the soft shadow.

Reference

- [1] Yingchao Zhang, Lipin Zou, Jia Liu, "Simulation of snow effects in visual simulation of virtual campus based on OSG," *Multimedia Technology (ICMT)*, 2011, pp.3658-3658.
- [2] Paul, Martz, "Open Scene Graph Quick Start Guide. California: Computer Graphics System Development Corporation, 2007. Pp .17-30.
- [3] Sung K, Craighead J, Changyaw Wang, BakshiS, Pearce A, Woo A, "Design and Implementation of the Maya Renderer," *Computer Graphics and Applications*, 1998. *Pacific Graphics 98. Sixth Pacific Conference*, 1998, 1(1) pp.150-150.
- [4] Xia San-ao, "Application of Maya in film 3D animation design," *Computer Research and Development (ICCRD)*, vol.8, no.2, pp. 357-358, 2011.
- [5] Guanbo Bao, Hongjun Li, Xiaopeng Zhang, Weiming Dong, "Novel Applications of VR Large-scale forest rendering: Real-time, realistic, and progressive," *Computers & Graphics*, vol.36, no.1, pp.141-144, 2012.
- [6] W Pasman, FW Jansen, "Comparing Simplification and Image-Based Techniques for 3D Client-Server Rendering Systems," *Visualization and Computer Graphics, IEEE Transactions*, vol.9, no.2, pp. 226-229, 2003.
- [7] Guanbo Bao, Xiaopeng Zhang, Wujun Che, Jaeger M, "Billboards for Tree Simplification and Real-Time Forest Rendering," *Plant Growth Modeling, Simulation, Visualization and Applications (PMA), 2009 Third International Symposium*, 2009, pp.434-436.
- [8] Jesu 's Gumbau, Miguel Chover, Inmaculada Remolar, Cristina Rebollo, "View-dependent pruning for real-time rendering of trees," *Computers & Graphics*, vol.35, no.1, pp.365-365, 2010.
- [9] Hesse M. Gavrilova M L, "An Efficient Algorithm for Real- Time 3D Terrain Walkthrough," *International Journal of CAD/CAM*, vol.3,no.2,pp.111-117,2003.
- [10] Nan Liu, Ming Yong Pang, "Shadow Mapping Algorithms: A Complete Survey," *Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium*, 2009, pp.1-5.
- [11] Xun Hu, Yue Qi, Xukun Shen, "A Real-time Anti-aliasing Shadow Algorithm Based on Shadow Maps," *Pattern Recognition, 2008. CCPR 08. Chinese Conference 2008*, pp.1-1.
- [12] Marroquim, R, Maximo, "A. Introduction to GPU Programming with GLSL," *Computer Graphics and Image Processing (SIBGRAPI TUTORIALS), 2009 Tutorials of the XXII Brazilian Symposium*, 2009, pp.7-8.
- [13] Tan Dunming, Zhao Gang, Yu Lu, "GPU Based Compression and Rendering of Massive Aircraft CAD Models," *Virtual Reality and Visualization (ICVRV), 2012 International Conference*, 2012, pp. 50-52
- [14] LIU You Quan, WANG WenCheng, WU EnHua, "Fast and Realistic Simulation for Trees Swaying in Wind," *CHINESE JOURNAL OF COMPUTER S*, vol.28, no7, pp.1185-1190, 2005.