2012 4th International Conference on Signal Processing Systems (ICSPS 2012)
IPCSIT vol. 58 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V58.26

Volumetric Lighting Realization Based on OGRE

Chaohui Lv[+],Tuo Shi, Yinghua Shen and Yuhua Fang

School of Information Engineering, Communication University of China
Beijing, China

Abstract. Volumetric lighting is widely used in 3D virtual-reality stimulation or some 3D rendering games. A solution on realizing volumetric lighting in OGRE (Object-Oriented Graphics Rendering Engine) is proposed and the approach based on slice volume rendering in OGRE is discussed. The implementation is introduced with some extensions to take advantage of shader hardware. A camera which represents the light projector frustum is declared. Rushes (500 in the demo) of billboards according to the light camera's frustum are made to simulate the volumetric lighting. A depth map is used in the light shafts material when the actual pixel is occluded by an object. Then the billboards are attached to a scene node in OGRE to achieve the related rotation of the beam. Through the proposed method, the light ray could be faked vividly in OGRE. Moreover, light shaft could be imitated in 3D scenes in variety of area.

Keywords: volumetric lighting, OGRE, billboards, camera.

1. Introduction

In the real world, we rarely see things in a vacuum, where nothing exists between an object and its observer. In real-time rendering, the effect of participating media on light transport is often subject to low-complexity homogeneous assumptions. This is due to the intractable nature of the radiative transport equation, accounting for emission, absorption, and scattering, in a complex interactive animated environment [1].

The volumetric lighting is a technique used in 3D computer graphics to add lighting effects to a rendered scene. It allows the viewer to see beams of light shining through the environment [2]. Seeing sunbeams streaming through an open window is an example of volumetric lighting, is also known as crepuscular rays. The term seems to have been introduced from cinematography and is now widely applied to 3D modeling and rendering especially in the field of 3D gaming.

OGRE (Object-Oriented Graphics Rendering Engine) is a scene-oriented, flexible 3D engine written in C++ designed to make it easier and more intuitive for developers to produce applications utilizing hardware-accelerated 3D graphics [3]. The class library abstracts all the details of using the underlying system libraries like Direct3D and OpenGL and provides an interface based on world objects and other intuitive classes.

To realize volumetric lighting in OGRE is always a difficult problem now. Due to GPU's efficiency and the effects of light, shaft must be taken consideration in our rendering engine at the same time. The methods in present world to stimulate light beam is using a mesh shaped like a cone beam. In order to alternate the color and the effects of the light beam, textures with different images are exploited in 3D scenes. This method seems easy and convenient, but in reality, it is not realistic enough. The biggest disadvantage is that can't simulate the shielding effect if there is an object occluded the light god.

In this paper, an approach that using numbers of billboard in OGRE to render light frustum is proposed. Through this mean, the volumetric lighting faked by the slice volume could be imitated vividly.

2. The Structure of the Stimulated Volumetric Lighting

[+] Corresponding author.
E-mail address: llvch@hotmail.com.

Realizing the volumetric light in OGRE mainly depends on the camera frustum and the related material written by scripts. The specific steps are described as follows.

2.1 . Declare a camera

First, we create a camera in the create scene function, and then setting the clip distance of it. The attribute of the clip distance can control the length of the volumetric lighting. The camera's projection type should be set as perspective mode. In this paper the angle of the frustum in the y-axis as 18.5.

As we know, A SceneNode is a type of Node which is used to organize objects in a scene [4]. It has the same hierarchical transformation properties of the generic Node class, but also adds the ability to attach world objects to the node, and stores hierarchical bounding volumes of the nodes in the tree. Child nodes are contained within the bounds of the parent, and so on down the tree, allowing for fast culling.

So in the next step, a scene node must be created in the scene by the manager. In that case, the camera created before could be attached to this node. The position and direction of this scene node determines the position and direction of the volumetric lighting. The observers can set these two parameters according to their wish.

2.2 . Make a rush of billboards

The volumetric lighting is inspired by the technique that Dobashi and Nishita [5] first developed in several papers. Sampling planes are shaded in light space. Then they are composited into frame buffer to approximate integral a long views. This principle can be described as Fig.1.

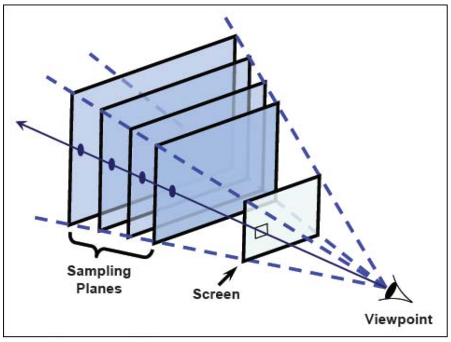

Fig. 1: Szmpling the light shaft

With this principle, the billboards should be created for volumetric rendering by the scene manager in OGRE. Then the material of these billboards would be set as the related material that we created by the material scripting language.

Here we should pay attention to the rotation type of the billboard settings. This setting controls the billboard rotation type. You can decide rotate the billboard's vertices around their facing direction or rotate the billboard's texture coordinates [4]. In this paper, this attribute is set as vertex mode. The cast shadow of the billboard is also set as false.

In the end, the billboard has to attach to the scene node which has already bounded a camera in previous.

2.3 . Declare a RTT

As we all know, the real light will be shielded by the object which is blocked the light beam. For that reason, the material must be created as depth material. In this paper, the depth map is established by high level shading language.

Firstly, a render target need to be created, then the depth listener would be added by the manager. In such way, the light would be updated each frame.

The material used by the billboards should be filled with the depth map material.

2.4 . A Little Noise Effects

In order to simulate the effect of flashing lights, a little noise effect is applied to stimulate air particles. The noise texture is shown as Fig.2.

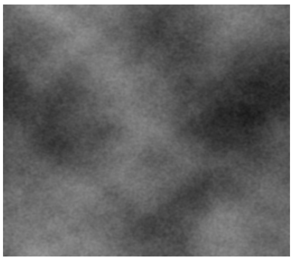

Fig. 2: Noise texture

3. The Related Material

The actual concrete subclass which will exist for a texture is dependent on the rendering system in use. This class represents the commonalities, and is the one 'used' by programmers even though the real implementation could be different in reality [6]. Texture objects are created through the 'create' method of the TextureManager concrete subclass.

In the material scripts, a material unite named ‘light shaft’ is defined in this paper. The texture used here should call the high level shading language material programs which could realize the depth map. Also on the texture unite, observers can choose the image according their wishes. The light texture is shown as Fig.3.

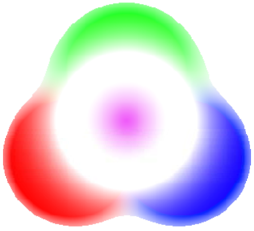

Fig. 3: Light texture

4. Experiments

Using our method, the volumetric lighting is rendered vividly in OGRE. The light could not penetrate the object. The final result is shown as Fig.4. The changing color light is shown as Fig.5.

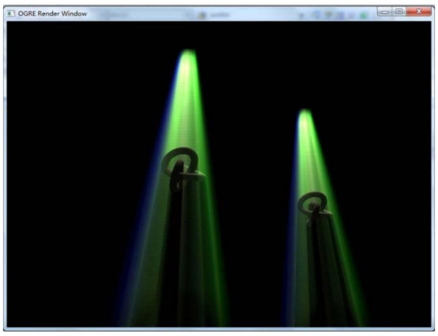

Fig.4 : Final result

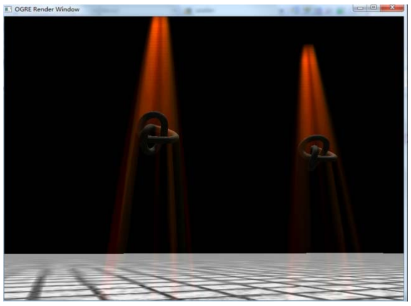

Fig. 5: Color changed lights

5. Conclution

Though the proposed method, a realistic volumetric lighting was stimulated in OGRE scene. With this vivid light beam, a wide variety of lighting effects in 3D rendering can be realized easily, such as the car lights, the stage lights, the god rays and so on.

In the other hand, because of the depth map material, we can imitate the real lighting when the light is blocked by some objects. In order to change the colors, the textures of the material could be transformed by the related functions. In addition, orientation and rotation are supported through setting the scene node.

6. Acknowledgment

This work is supported by the Science and Technology Innovation project of Ministry of Culture (No.2010-10), the University Level Research project of Communication University of China (No.XNG1008), and the National University Student Innovation Test Plan (No.G2010033).

7. References

[1] Hoffman, N., and K. Mitchell. "Methods for Dynamic, Photorealistic Terrain Lighting." *In* Game Programming Gems *3*, edited by D. Treglia, 2002, pp. 433–443.

[2] Charles River, Media.Edward,Yourdon, Carl Argila. Practical Object-Oriented Software Engineering Tutorials, PRENTICE HALL, 1998, 6 .

[3] Jensen, H. W., P. H. Christensen. "Efficient Simulation of Light Transport in Scenes with Participating Media Using Photon Maps." In Proceedings of SIGGRAPH, 1998, pp. 311–320.

[4] http://www.ogre3d.org/

[5] Dobashi, Y., T. Yamamoto, T. Nishita. "Interactive Rendering of Atmospheric Scattering Effects Using Graphics Hardware." Graphics Hardware, 2002, pp.1-10..

[6] http://www.ogre3d.org/docs/api/html/classOgre