Int. Computer Symposium, Dec. 15-17, 2004, Taipei, Taiwan.

Real-Time Shadow Rendering Using Projective Texture

Jyun-Ming Chen, Kai-Pin Tung
Department of Computer Science and Engineering, Tatung University, Taiwan
jmchen@cse.ttu.edu.tw, g9106038@ms2.ttu.edu.tw

Abstract- *Shadow rendering is a complicated count in the GPU (Graphic Process Unit), it can simply and vivid render the model's shadow in the scene, it will improve on the scene realistic image and provide artistic placement for the user.*
This paper emphasizes on the shadow rendering in the scene, using Projection Texture Shadow to render the model's shadow in the scene. Texture image be the model's shadow has simply and vivid features to reduce the complexity of the shadow rendering. In addition, in order to enhance the realistic shadow, except shadows' color rendering, soft shadow rendering, and consider the projected shadows in the multiple models and determine the model in the scene should be shadowed or not.

Keywords: Shadow, Projective Texture Mapping, Projective Texture Shadow, Computer Graphics.

1. Introduction

Shadow can make the 3D graphic more believable, and knows the position and the movement of the model. And the scene with the shadow can improve on the scene realistic image and provide artistic placement for the user. Although the shadow has these features, but it always abandon due to difficult rendering. Hence, simply and vivid render the shadow of the model in the scene is my study of thesis.

In general, the shadow rendering algorithms are Projection Shadow, Shadow Map, Stencil Shadow Volume and Projective Texture Shadow. Projection Shadow, render the shadow in the plane and can not render the shadow in the terrain. It should render the shadow in each plane. Shadow Map and Stencil Shadow Volume can render the shadow in the terrain, but can not easy implement. Projective Texture Shadow, take the model in the scene as shadow texture, is a simply way to implement and has vivid shadow. This research chooses this algorithm to render the shadow.

2. Shadow Rendering

The procedures of the shadow rendering are rendering the model in the scene from the light position and getting the shadow texture, using the shadow texture map the model coordinate in the scene and finish the shadow rendering. The Fig.1 shows the procedures of the shadow rendering. Figure 2 shows the techniques of the texture mapping the model coordinate.

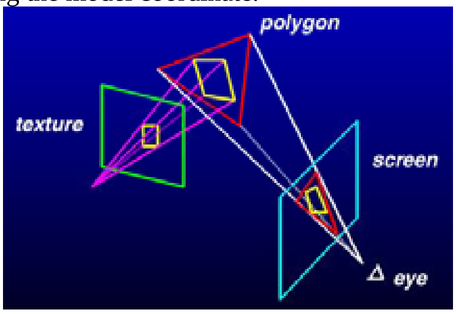

Figure 1.the procedures of the shadow rendering [6]

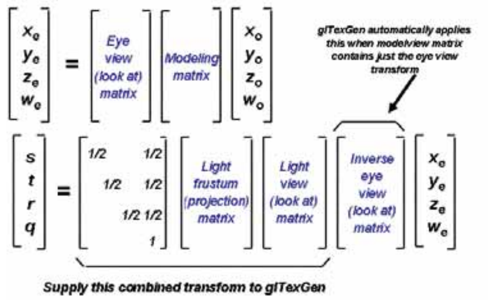

Figure 2.the texture coordinates equation [8]

The color of the shadow is hyaline black in the real world. And, the receiver can show the shadow and the texture of the receiver in the same time. The Fig.3 shows the color of the shadow rendering.

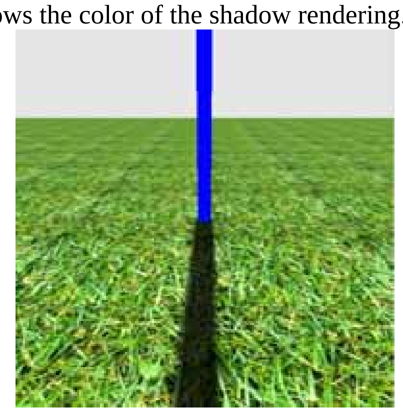

Figure 3.the color of the shadow rendering

Soft shadow is the shadow of the model in the scene by using the area light, and renders the shadow in the plane or the terrain. One of the soft shadow rendering techniques is dividing the area light to multiple point light. Render the shadow each point light; the shadows will be the shadow in the area light. And the intersections of the shadow are blacker than the color of the shadow. The Fig.4 shows the soft shadow in multiple lights.

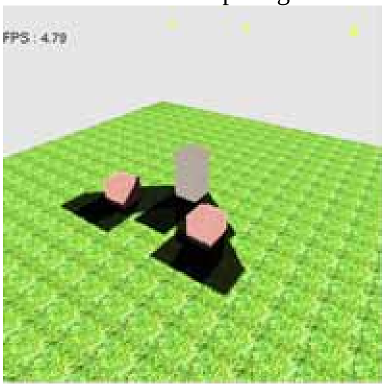

Figure 4.the soft shadow in multiple lights.

The second way of the soft shadow rendering techniques is doing smoothing. The hard shadow texture becomes the soft shadow texture by using the smoothing. Figure 5 is the procedure. Texel of the soft shadow texture is the average value of the eight Texel values adding from the hard shadow texture. Figure 5 is the result of the soft shadow by using smoothing.

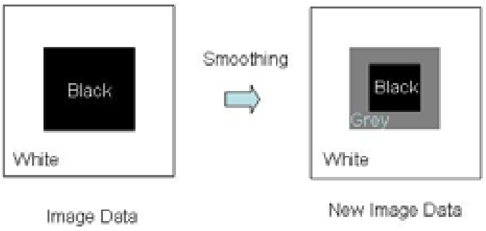

Figure 5.the hard shadow texture becomes the soft shadow texture

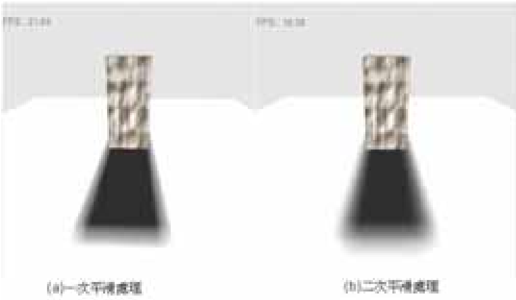

Figure 6.the effect of the soft shadow by smoothing numbers.

3. Shadow rendering in the multiple models

All the models in the scene will be the occluder or the receiver. Figure 7 is the situation of multiple occluders and multiple receivers. So, the shadow rendering procedures will divide to two parts. First, render the shadow of all models in the scene from the light position. Second, render the shadow of the model in the receiver models. Figure 8 shows the scene without the shadows. Figure 9 shows the procedure of the first part. Figure 10 shows the procedure of the second part. Figure 11 shows the scene with the shadows.

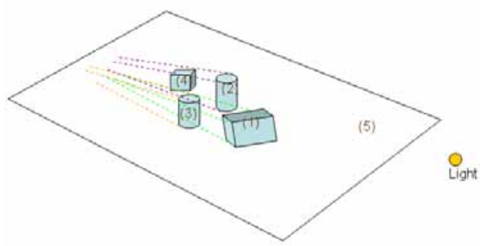

Figure 7.multiple occluders and multiple receivers

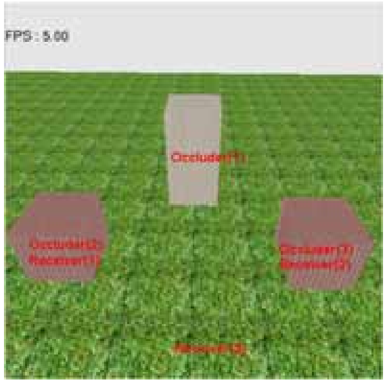

Figure 8.the scene without the shadows

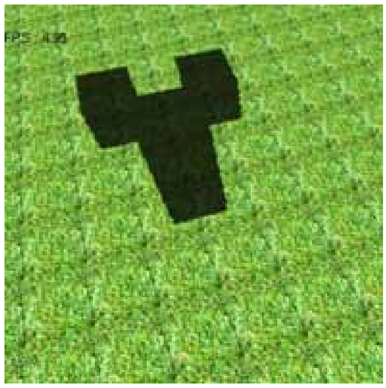

Figure 9.the procedure of the first part

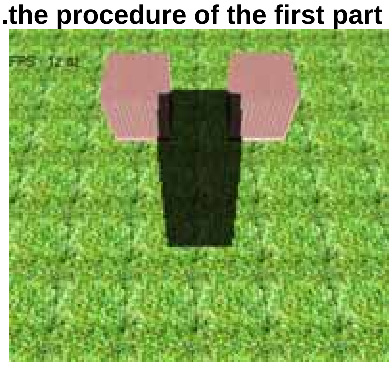

Figure 10.the procedure of the first part

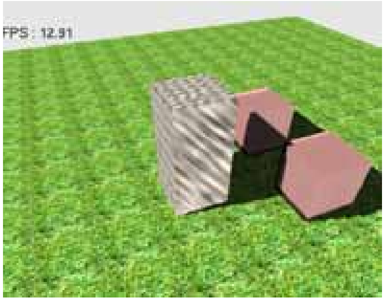

Figure 11.the scene with the shadows.

Then, we proposed the model in the scene should be shadowed when the model in the shadow frustum of another model, avoid rendering the shadow of all models from the light position. Rendering the shadow of all models is unneeded procedure for some models are not occluders or the receivers.

In order to simply describe the models, we are bounding them with the bounding sphere. And using the bounding sphere replaces the model in the scene. All the bounding spheres of the models are occluders in the point light. So, the light will generate the infinite cone according to the occluder from the light position. The cone has two compositions; one is the cone including the light, and the other is the truncated cone. The shadow frustum of the occluder is the scope of the truncated cone. We can determine the model in the scene should be shadowed in point light and the direction light. Figure 12 is the definition of the point light, occluder and the shadow frustum.

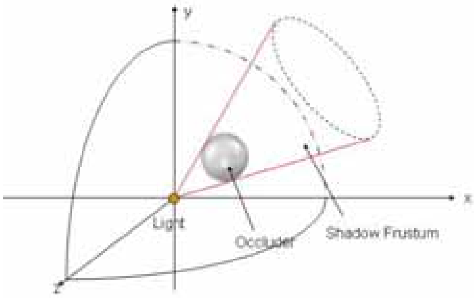

Figure 12.the point light, occluder, and the shadow frustum.

All the bounding spheres of the models are occluders in the point light. So, we can compare them in pairs to determine the model in the scene should be shadowed or not. We define some parameters to be the compare parameters. The parameters are the radius of the model bounding sphere, r_i, the half-angle of the cone, θ_i and the angle of the cone central shaft, ϕ. Figure 13 shows the clear ideas about these parameters assume the bounding spheres of the models and the light in the same plane. After the definitions, we have three cases to determine the model in the scene should be shadowed or not by using the half-angle of the cone, θ_i and the angle of the cone central shaft, ϕ. Because we compare the situation in pairs, so we can get two half-angle of the cone, θ_1 and θ_2. In case 1: the ϕ angle is bigger than the angle of θ_1 adds θ_2. Figure 14 shows the two models in the scene should not be shadowed. In case 2: the ϕ angle is equal to the angle of θ_1 adds θ_2. Figure 15 shows the two models in the scene should not be shadowed. In case 3: the ϕ angle is lesser than the angle of θ_1 adds θ_2. Figure 16 shows the two models in the scene should be shadowed. And then, we can determine the occluder and the receiver in the models by using the distance from the point light to the centre of a circle, d_i. The distance is small, the model is occluder. Otherwise, the model is the receiver. Figure 17 shows the parameter to determine the occluder and the receiver.

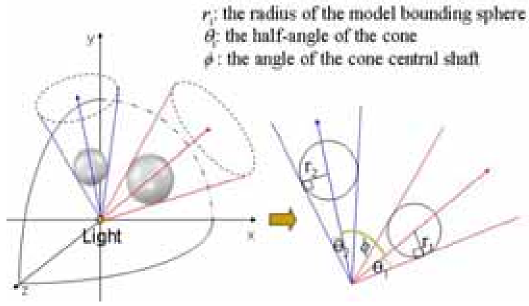

Figure 13.the compare parameters

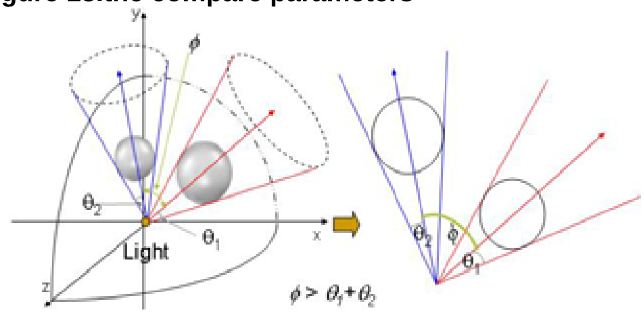

Figure 14.models should not be shadowed.

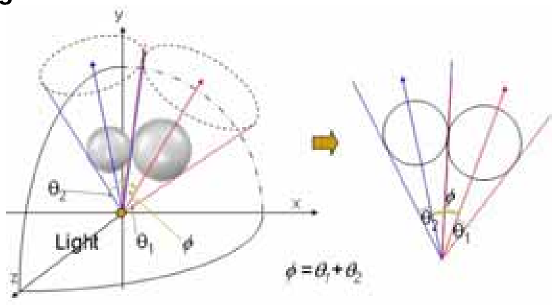

Figure 15.models should not be shadowed.

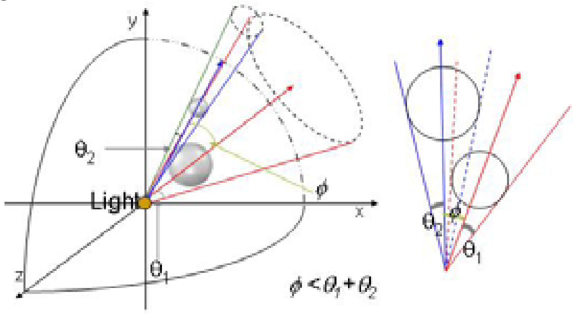

Figure 16.models should be shadowed.

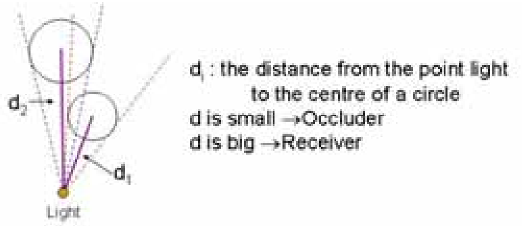

Figure 17.the parameter to determine the occluder and the receiver.

Direction light projects the model bounding sphere to the corresponding plane with the light direction. Figure 18 shows the light direction is –x axis, so the projection plane is y-z plane. We can determine the model in the scene should be shadowed or not by comparing the intersections of the model bounding sphere projection in pairs. Figure 19 shows the three situations of rendering the shadow or not. Figure 18 (a) shows the projections are not intersecting, the two models in the scene should not be shadowed. Figure 18 (b), (c) shows the projections have intersections; the two models in the scene should be shadowed. We can use the signed distance to determine the occluder and the receiver of two models in the projection plane. The signed distance is the dot product of the vector p_i (the direction of the model bounding sphere) multiples the normal n (the normal of the projection plane). Figure 20 shows the signed distance and determine the occluder and the receiver of two models.

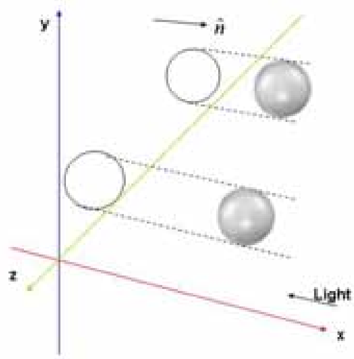

Figure 18.light directions, projection plane

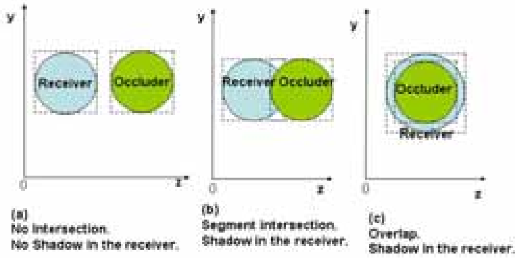

Figure 19.three situations of rendering the shadow or not

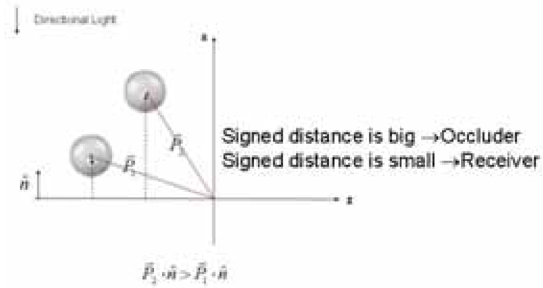

Figure 20.determine the occluder and the receiver by signed distance

4. Implementation and discussion

The working platform of this research is as follows: Operating System is Microsoft Windows XP Home Edition, CPU is Pentium 4 2.4G, Memory is 256MB SDRAM; Display Chip is Sis 650 32MB on board, and the Program Development Environment is Microsoft Visual C++ 6.0 and OpenGL.

Figure 21 and Fig.22 show the MD2 models (the model in the Quake [2] game) with their shadows in the plane and the terrain and the multiple receivers. Figure 23 shows the MD2 model with its shadows in the plane and the terrain with the multiple lights. Figure 24 shows the MD2 model with its soft shadow in the plane and the terrain. The soft shadows are generated by smoothing.

In addition, the MD2 models can render their shadows in the moving objects. The procedures are the same as the chapter 3 described. First, render the shadow of all models in the scene from the light position. Second, render the shadow of the model in the receiver models. The result is Fig.25.

The MD2 model will load the texture which is defined by the MD2 data structure. And this feature fails the shadow of the occluder rendering in the MD2 model, when the receiver model is the MD2 model. Because the MD2 textures will cover the shadow texture, so the MD2 model can not render the shadow texture.

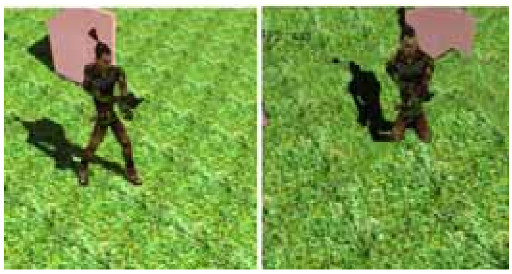

Figure 21.the MD2 models in the plane and the terrain

Figure 22.the MD2 models in the multiple receivers

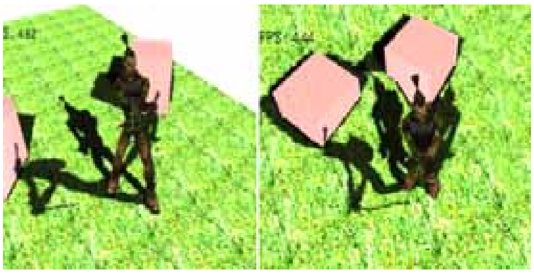

Figure 23.the MD2 models in the plane and the terrain with multiple lights

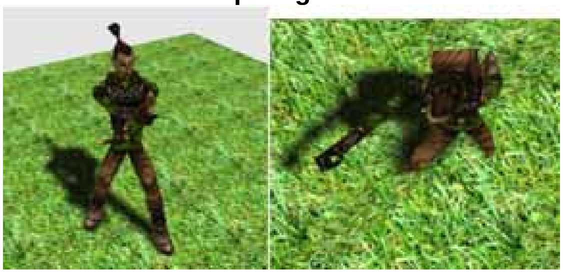

Figure 24.the MD2 models with its soft shadow in the plane and the terrain

Figure 25.the shadow of the MD2 model in the plane, terrain and the moving object

Using the shadow texture to be the shadow of the model will generate the aliasing. Because the shadow texture is generated by the perspective projection, so the boundaries of the texture have aliasing. Figure 26 is the aliasing. And the model sizes are always bigger than the texture sizes, when the model in the scene loading the texture. So the texture should be enlarged to match the model. The aliasing will be serious by using the small texture sizes to match the model than using the larger texture sizes to match the model. So, improve on the aliasing in the texture, we should use the larger texture sizes to match the model. Figure 27 is the compare of the texture sizes, and we can know the larger texture sizes will get the better realistic image.

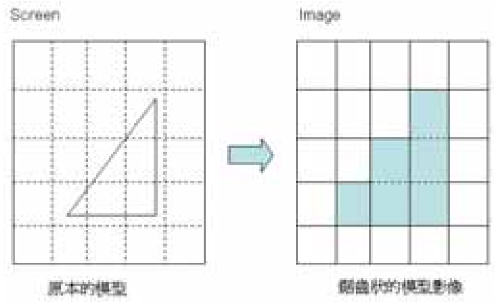

Figure 26.aliasing

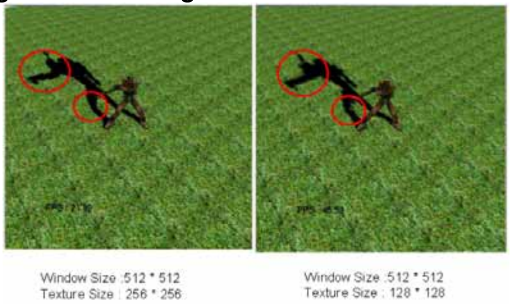

Figure 27.the compare of the texture sizes

5. Conclusion and Future work

The scene with the shadow can improve on the scene realistic image and provide artistic placement for the user. And the shadow can make the 3D graphic more believable, and knows the position and the movement of the model. We discuss the shadow rendering algorithms including the Projection Shadow, Shadow Map, Stencil Shadow Volume and Projective Texture Shadow. In addition, we discuss the shadow rendering in the plane, in the terrain and soft shadow rendering. And then, we use the projective texture shadow render the shadow in the multiple models and determine the model in the scene should be shadowed or not. These parts are less discussion in the others paper, we focus in these parts to complete the shortage parts.

We render the shadow from the light position in this research. And these procedures are very slowly, but we can make sure the shadow of all models in the scene could be rendered correct. In future, we can use some accelerating algorithms, like frustum culling or the occlusion culling, to accelerate the rendering procedures. In the part of determine the model in the scene should be shadowed or not, we can let all models in order in the scene. Then, compare all models projections at once, not compare all models projections in pairs.

In future, we can render the reflection. Mirror、water and all models which reflect the light always can render shadow, but it called reflection not shadow. The difference between shadow and reflection are that shadow describes the model’s shape and motion, and the shadow’s color is transparent black, But reflection reflect the environmental scene.

References

[1] Cass Everitt, "Projective Texture Mapping," NVIDIA SDK White Paper, http://developer.nvidia.com/object/Projective_Texture_Mapping.html, 2001.

[2] GameBase, http://www.gamebase.com.tw

[3] GameDev.net, http://www.gamedev.net/columns/hardcore/shadowvolume/default.asp

[4] Lance Williams, "Casting Curved Shadows on Curved Surfaces," SIGGRAPH 78, 1978

[5] Mark J. Kilgard, "Shadow Mapping and Shadow Volumes," SIGGRAPH 02, 2002

[6] Mark J. Kilgard, "Shadow Mapping," CEDEC 2001 Tokyo, Japan, http://developer.nvidia.com/object/cedec_shadowmap.html, 2001

[7] Mark Segal, et al., "Fast shadows and lighting effects using texture mapping," SIGGRAPH 92, pages 249-252.

[8] Mark J. Kilgard, "Shadow Mapping with Today's OpenGL Hardware," NVIDIA SDK White Paper, 2001

[9] Michael I. Gold, "Projective Textures," NVIDIA Corporation GDC99 http://developer.nvidia.com/object/projective_textures.html, 2000

[10] Tomas Akenine Moller Eric Haines, "Real-Time Rendering Second Edition," A K Peters, 2002, Pages 248-275

[11] William T. Reeves, "Rendering Antialiased Shadows with Depth Maps". SIGGRAPH 87, pages 283-291