Shadow Texture Atlas

Kyoung-su Oh [1], Tae-gyu Ryu[1]

[1] Department of Media, University of Soongsil, Korea
{oks, mordor}@ssu.ac.kr

Abstract. In this paper, we present a new method for creating shadow texture atlas with which we can represent the self-shadow. Shadow texture atlas is a texture atlas with shadow information in each texel. Shadow texture is effective to represent high-quality shadow using the graphics hardware because it stores shadow information as color unlike shadow map which stores depth. However, it cannot represent the self-shadow and it takes a long time to create a shadow texture. To overcome these problems we present shadow texture atlas method. Shadow texture atlas can also represent self-shadow information. Moreover, we use shadow map to create shadow texture atlas instead of drawing all shadow casters. Our experimental result shows that our method performs at linear time and its speed is similar to the traditional shadow map method. Furthermore in comparison with shadow map method, our method can use mipmap filtering which is hard to achieve with shadow map method.

1 Introduction

Shadow is not common in computer games on account of the expensive costs. Most widely used real-time shadow generation methods are shadow map, shadow texture, etc.

In shadow map method, we render scene from the light's view, and use the depth map to determine which surfaces lies in shadow [1]. However, it can not utilize graphics hardware features such as texture filtering because unlike filtering of color values, filtering of depth values may result in wrong shadow. There were some research to overcome this shortcoming such as percentage closer filtering[2] and variance shadow map[3]. Percentage closer filtering filters the result of the depth comparison. If the shadow map area covered by a pixel is large percentage closer filtering needs many depth comparison. Variance shadow map stores moments of depth distribution function and it uses the Chebyshev's inequality to get upper bound of the probability that the center of each pixel is in shadow. If variance of a pixel is high, light bleeding artifact can occur.

Shadow texture method maps the texture image storing shadow information on rendered objects [4]. Shadow information is represented by texel color, therefore, it can present high-quality shadow effect using hardware accelerated image filtering. However, the projective shadow texture method can not support the self-shadow and it takes a long time to create a shadow texture.

In this paper, we present a shadow texture atlas method. In this method, shadow information is stored in shadow texture atlas that is shadow texture in the form of texture atlas. And we use shadow map when generating shadow texture atlas. Using shadow map, it is possible to generate shadow texture atlas in linear time to the number of objects in the scene. Advantages of shadow texture atlas method compared with traditional shadow texture methods are support of self-shadow and fast generation time. And advantage over shadow map method is enhancement of image quality using hardware accelerated filtering.

2 Shadow texture atlas method

Shadow texture atlas method includes next three steps. First, we create the shadow map. Second, we create atlas of shadow textures for each object with shadow map. And last, we render the scene with the shadow texture atlas.

2.1 Shadow map creation

To create a shadow map, we perform Z-buffer algorithm with a light source as a view point. As a result, the distance of the closest model to light source is stored in depth buffer. These values are copied to shadow map.

2.2 Shadow texture atlas creation for each object

With planar projection, we cannot store correct shadow information for concave objects because some positions may be projected to the same texture coordinate. With texture atlas[5], we can map every position on 3D model surface to different texture coordinates. We create shadow texture atlas for each object by rendering.

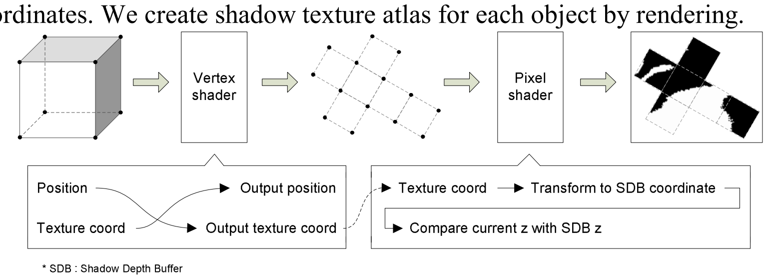

Fig. 1. Shadow texture atlas creation

Fig. 1 is a rendering process. The vertex shader outputs shadow texture atlas coordinate as its position. And it also output model space coordinate. Then model space coordinate are interpolated and passed to pixel shader by graphics hardware. Consequently, a pixel shader is transferred a three-dimensional position (value) corresponding to a pixel.

The pixel shader transform model space coordinate into shadow map coordinate. And it compares the z (depth) value of it with a z value obtained from a shadow map. If current pixel's z value is larger than stored value in shadow map then shadow color is assigned. Otherwise illuminated color is output.

2.3 Rendering with shadow texture atlas

With stored position and atlas texture coordinate in the each vertex, we can render shadowed objects by mapping shadow texture atlas to each model. By hardware mip-mapping, shadow texture decreases aliasing artifacts. In particular aliasing artifacts which are significant during texture minification are reduced.

3 Implementation and results

We implemented the shadow texture atlas method on a system with Athlon64(3000+), Nvidia 7800GTX. Test model is a geometry image model of the Stanford Bunny.

GU's [6] geometry image generation method minimizes the shape transform of each triangle and incision of 3D model. Because each point on 3D model is mapped to a unique 2D position on the geometry image, we can use geometry image as texture atlas.

We tested with 512x512 size of the shadow map. Test model has 66049 vertices. The size of the shadow texture atlas is 256x256. Fig.2 is comparison of our method with shadow map method. This comparison shows the effect of mipmap filtering.

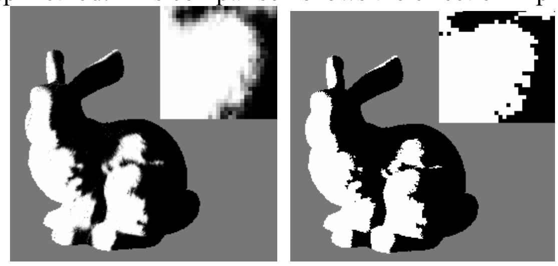

Fig. 2. Rendered output and : (a) shadow texture atlas (left) (b) shadow map (right)

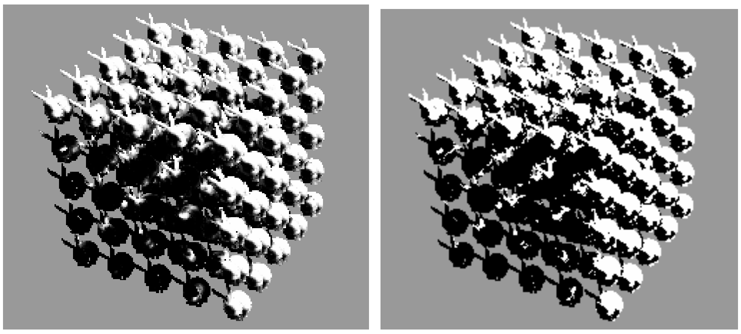

Fig. 3. Rendered output. (a) atlas shadow texture (left) (b) shadow map (right)

A 3D model used in Fig. 3 is a low resolution geometry image which has 4096 vertices. Moreover, we use a shadow texture atlas of 64x64 size in for each objects.

Performances of rendering Fig. 2, 3 are listed in Table 1. Performance of our method is similar to that of shadow map method.

Table 1. The frame rate

Scene	Shadow texture atlas	Shadow map
Fig. 6	123.46 fps	126.20 fps
Fig. 7	16.30 fps	18.28 fps

4 Conclusions

Shadow texture method has difficulty in representing self-shadows and it takes a pretty long time to create a shadow texture. We presented shadow texture method that uses shadow texture atlas. Our method can create shadow texture representing self shadow in linear time. Advantage compared with shadow map method is image quality when viewer is far from the shadowed models.

5 Acknowledgments

This work was supported by the Korea Research Foundation Grant (KRF-2004-005-D00198).

References

1. Heidmann, T: Real shadows real time, IRIS Universe, 18, 28–31, 1991
2. Reeves, W., Salesin, D., and Cook, R.: Rendering antialiased shadows with depth maps. ACM SIGGRAPH Computer Graphics, v. 21 n4, 283–291, 1987.
3. William Donnelly, Andrew Lauritzen: Variance Shadow Maps. Variance shadow maps, Proc. of symposium on Interactive 3D graphics and games, 161-165, 2006
4. Mark Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran, Paul Haeberli: Fast shadows and lighting effects using texture mapping, ACM SIGGRAPH Computer Graphics, v.26 n.2, p.249-252, July 1992
5. Bruno Lévy, Sylvain Petitjean, Nicolas Ray, Jérome Maillot: Least squares conformal maps for automatic texture atlas generation, ACM Transactions on Graphics (TOG), v.21 n.3, July 2002
6. Xianfeng Gu, Steven J. Gortler, Hugues Hoppe: Geometry images, Proceedings of the 29th annual conference on Computer graphics and interactive techniques, July 23-26, 2002