MESH DECIMATION FOR SHADOW TERMINATOR

Kristian Spoerer and Weibin Wang
School of Computer Science
University of Nottingham, UK

ABSTRACT

In rendering, low complexity 3D meshes can result in aliasing at the shadow terminator. This study aims to develop a novel mesh decimation algorithm that maintains a smooth shadow terminator. By concentrating on regions away from the shadow terminator, the proposed algorithm can reduce computational demands without compromising the render. We demonstrate the algorithm's effectiveness and compare it with existing methods.

KEYWORDS

Shading and Rendering, Geometric Modeling

1. INTRODUCTION

In 3D real time graphics, which can be used in videogames, medicine (Nogimori, et al., 2020), and in AR/VR (Kwak, et al., 2020), triangle meshes can be used to represent smooth surfaces for fast rendering. Higher complexity meshes appear more like the target shape but also take more time to render. Shadows help to improve realism and perception of the scene, but problems can include aliasing at the shadow terminator (Akenine-Möller, et al., 2018). When rendering the shadow using shadow mapping and simulating light rays in the fragment shader, some of the geometry near the shadow terminator will be unlit, and some lit, leading to artifacts (Jen-Yuan Chiang, et al., 2019). The problem is apparent when the 3D mesh complexity is low.

Figure 1 (a) shows a low complexity mesh where many surface points are facing away from the light source and will not be lit resulting in the incorrect shadow terminator shown in (b). Figure 1 (d) shows more complex geometry and in (e) there is no apparent problem on the shadow terminator.

(Snyder & Barr, 1987) performed early investigations to solve this problem, specifying an epsilon variable that moves the calculation point away from the low-complexity geometry, which can be implemented in the fragment shader. (Hanika, 2011) proposed a method to calculate this epsilon value based on Bezier patches. This generated sample point, *p*, is shown in Figure 1 (c). The problem with this method is that the generated point is not guaranteed to be on the intended geometry.

2. METHOD

Rather than generate sample points we propose to use actual complex geometry. Our method starts with a complex mesh and decimates it away from the shadow terminator using a decimation algorithm. (Garland & Heckbert, 1997) iteratively contracts two vertices into one vertex according to the error introduced which is the distance from the original vertex position, such that the vertices with smallest error will be contracted first. Our algorithm contracts two vertices based solely on their distance apart and is fast to compute. To calculate if a triangle is on the shadow terminator region, we use the method shown in Figure 2, which works by calculating the angle of the triangle normal compared with the light source direction to see if it is within a threshold.

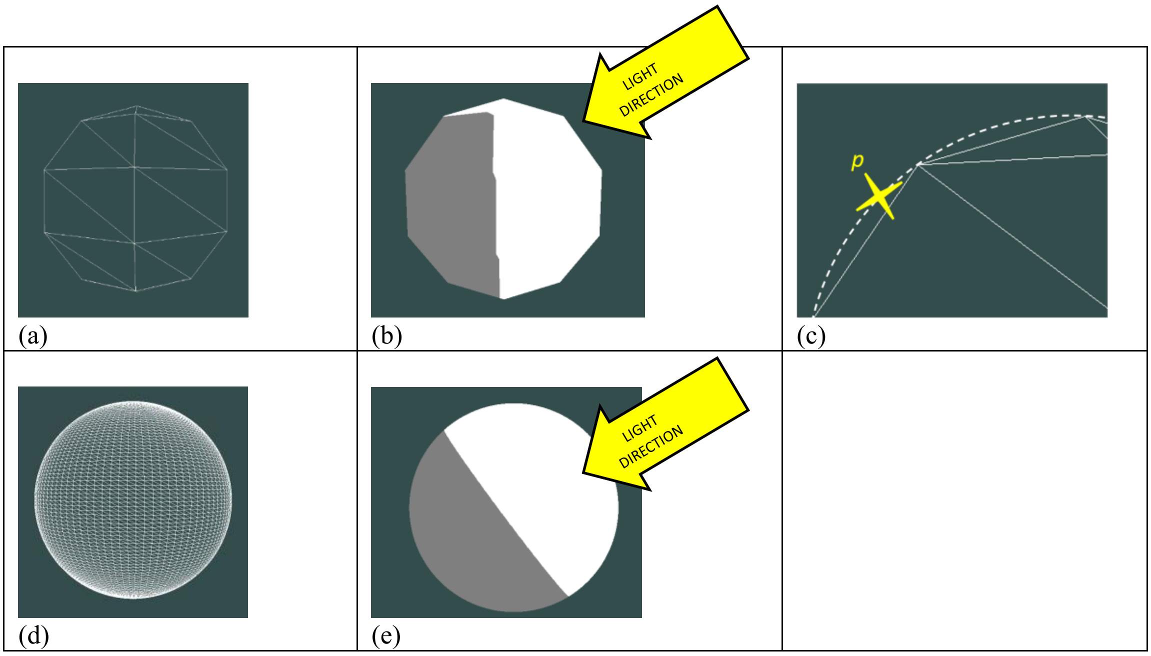

Figure 1. Showing the shadow terminator problem

```
n: normal of mesh triangle (unit vector)
l: vector pointing towards light source (unit vector)
d: dot product of n and l

angle = acos(d)
if (abs(abs(angle)-90)) <= theta
   return true
return false
```

Figure 2. Pseudocode to calculate if triangle is on shadow terminator

3. RESULTS

We tested our method using a procedurally generated sphere and a GLSL implementation of shadow mapping, with geometry decimated to 14% of original model complexity. Figure 3 illustrates the dynamic nature of the method. Each row from top to bottom shows the light source in a different position as can be seen from the position of the shadow on the flat plane above the sphere and the angle of the shadow terminator. Figure 3 can be compared with the result shown in (Figure 4-4 top row second column) in (Hanika , 2021) and as can be seen our method produces a shadow terminator which is comparable to existing state of the art results. It should be noted that the decimation cannot currently be achieved in real-time.

4. CONCLUSION

We have proposed a method for tackling aliasing problems on the shadow terminator. Our proposed method uses complex geometry rather than generated points, which we hope will produce more accurate rendering. Our method will work for local lighting and global lighting. Also, we have considered the shadow terminator region during mesh simplification. We have demonstrated its effectiveness in a dynamic scene and have also showed that it is comparable to existing results. Future work will aim to provide evidence that our method is more accurate than existing methods, will test alternative faster methods which run in real time, and will also aim to demonstrate use in practical applications.

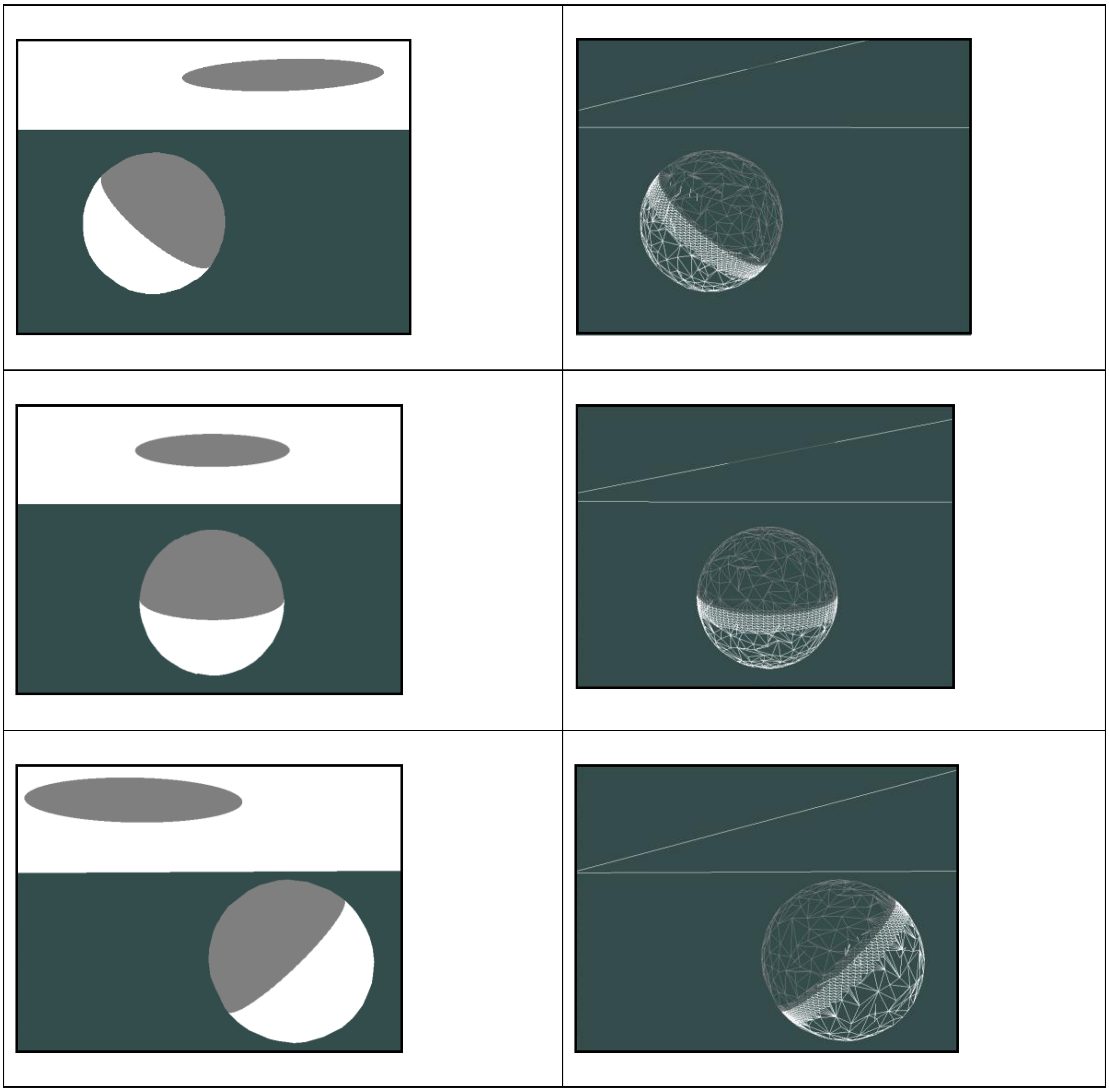

Figure 3. Results showing the dynamic nature of our method

REFERENCES

Akenine-Möller, T., Haines, E. & Hoffman , N., 2018. *Real-Time Rendering.* 4 ed. : CRC Press.

Garland, M. & Heckbert, P., 1997. Surface Simplification Using Quadric Error Metrics. *SIGGRAPH,* pp. 209–216.

Hanika , J., 2021. Hacking the Shadow Terminator. In: *Ray Tracing Gems II: Next Generation Real-Time Rendering with DXR, Vulkan, and OptiX.* Berkeley, CA: Apress , pp. 65–76.

Hanika, J., 2011. *Spectral light transport simulation using a precision-based ray tracing architecture,*: MSc Thesis.

Jen-Yuan Chiang, M., Karl Li, Y. & Burley, B., 2019. Taming the Shadow Terminator. *SIGGRAPH ,* pp. 1–2.

Kwak, S., Choe, J. & Seo, S., 2020. Harmonic rendering for visual coherence on mobile outdoor AR environment. *Multimedia Tools and Applications,* Volume 79, pp. 16141–16154.

Nogimori, Y., Seo, H. & Ueda, H., 2020. A three-dimensional computer graphics tool for congenital heart diseases. *Pediatrics International,* 62(6), pp. 738–740.

Snyder, J. & Barr, A., 1987. Ray tracing complex models containing surface tessellations. *Computer Graphics,* 21(4), pp. 119–128.