Computing Visibility Functions Using Polygon Intersection Algorithms

Author: Marc Ruijs **Contact:** m.j.a.ruijs@student.tudelft.nl TU Delft **Responsible Professor:** Ricardo Marroquim **Supervisor:** Yang Chen

1 Introduction

- In computer graphics, determining the amount of surface area that is *visible from a viewpoint* is a relevant problem [1].
- Approximative algorithms exist, but these are inherently inaccurate.
- This research provides an exact algorithm, such that the accuracy of approximative methods can be evaluated.

2 Research Question

Are approximative visibility algorithms accurate enough to justify their usage?

4 Results

Zenith	Our algorithm	Approximated method
90°	100%	99.1%
80°	100%	99.2%
70°	95.1%	94.7%
60°	86.8%	86.9%
50°	79.2%	79.5%
40°	38.5%	71.8%
30°	62.9%	63.9%
20°	55.4%	55.6%
10°	40.2%	49.3%

Table 1: The outputs for our algorithm, as well as an approximative one. Results for varying viewing angles are listed

5 Conclusion

- Results show that approximated outputs are not identical to the outputs of our algorithm, so for applications that require maximum accuracy, an exact method is still recommended.
- However, the differences between them are small, and for any situation that does not demand maximum accuracy, using an approximative method is justified.

6 Limitations

Outliers in the results are caused by floating-point errors, which cause the algorithm to make mistakes.

References

[1] T. Akenine-Mller, E.Haines, and N. Hoffman, *Real-Time Rendering, Fourth Edition,* 4th USA: A. K. Peters, Ltd., 2018, pp. 327-338.

3 Methodology

Figure 1 shows a surface of which we want to find the visible area, which is done following these steps:

1. Project surface triangles onto our 2-dimensional view-space, as shown in Figure 2.
2. Combine all triangles into one shape. The area of this shape equals the visible surface area, and is shown in Figure 3.
3. Triangulation of this shape results in a collection of triangles that take up the same area, as shown in Figure 4.
4. Calculate the area for each triangle and add them up.

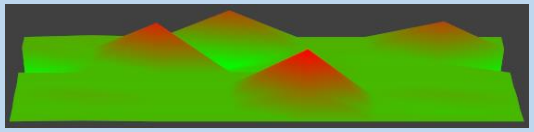

Figure 1: The surface of which we want to calculate the visible area

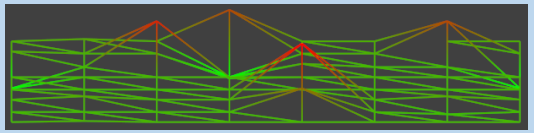

Figure 2: The surface triangles projected to our view-space

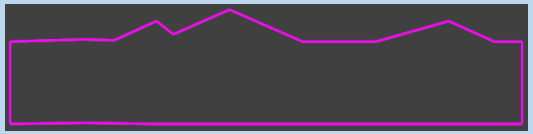

Figure 3: The shape that is the result of combining all projected triangles

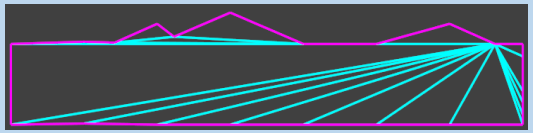

Figure 4: The shape is split up into triangles by the triangulation algorithm