

Ambient Occlusion for Self-Shadowing Objects using Ray-casting and the OpenGL Shading Language

Introduction

Ambient Occlusion is used to calculate how much light the surface of an object receives. This is very useful in making objects look more realistic in a 3D or even a 2D scene [1]. The generations of shadows through ambient occlusion techniques can greatly increase the realism of an object or character model. Calculating ambient occlusion is performed by projecting out a ray from every point to a hemisphere to calculate if an object is shadowed by another. Ray casting has been chosen to calculate shadowing on an object to take advantage of the ability to shoot multiple rays at the once to calculate dark areas more efficiently [2], [3].

Aims

- Research and implement an efficient Ray casting algorithm for ambient occlusion
- Determine which algorithm to use for self-shadowing objects
- Compare algorithms used in ambient occlusion techniques
- Implement code using OpenGL shading language to consider best practices for parallel ray casting algorithms
- Benchmark code for efficiency

Michael Thomas Rhys Johns | P147264
MComp(Hons) Computer Games Development

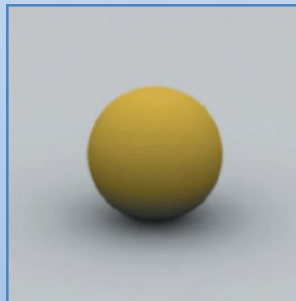


Figure 1 [4]: Sphere with Ambient Occlusion

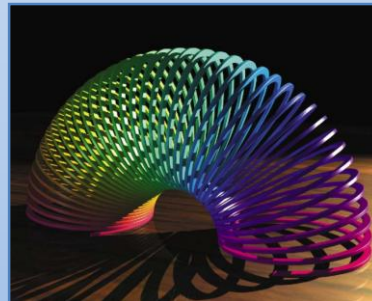


Figure 2 [4]: Slinky showing Self-Shadowing results

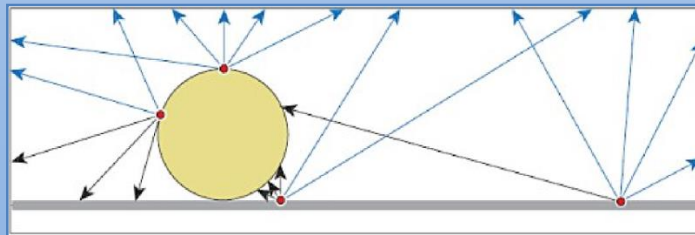


Figure 3 [4]: Ray casting from points on the sphere and a plane, to calculate ambient occlusion (black show shadow rays)

Research Methodology

This project will use the Rational Unified Process as its software engineering principle. This will allow an agile approach with a focus on development of the software for this project.

Conclusion

The aim of this project is to determine if parallel ray casting using GPU Shaders, is more efficient than a sequential approach using a CPU to calculate Ambient Occlusion. The results will be calculated by benchmarking how much time is taken for both approaches.

References

- [1]Rost., *OpenGL Shading Language*, 3rd ed. Addison-Wesley Professional, 2010, pp. 380-385.
- [2]M. Levoy, "Efficient ray tracing of volume data", *ACM Transactions on Graphics*, vol. 9, no. 3, pp. 245-261, 1990.
- [3]P. Shanmugam and O. Arikan, "Hardware accelerated ambient occlusion techniques on GPUs", *Proceedings of the 2007 symposium on Interactive 3D graphics and games - I3D '07*, 2007.
- [4]K. Suffern, *Ray tracing from the ground up*, 1st ed. Wellesley, MA: A K Peters, LTD, 2007, p. 304, p. 315.