CM3202 Initial Plan - Efficient Ordered Transparency Rendering for Static 3D Meshes

Conor McSweeney (C1834704)
Supervised by Dr Xianfang Sun

February 2021

1 Project Description

In traditional rasterized 3D graphics, correct rendering of transparency in a scene is a difficult problem. Transparent pixels/triangles must be rendered from furthest to nearest in the scene to appear correct, as the result of alpha-blended transparency is order dependent.

A complete general solution requires significant real-time computation that has only become feasible on consumer-grade GPUs in the last few years with methods such as per-pixel linked lists (Yang et al., 2010). Approximate general solutions can be obtained with less computation with per-object sorting, or order-independent transparency techniques like Dual Depth Peeling (Bavoil and Myers, 2008). These methods can produce non-ideal results, requiring a compromise between performance and visual quality.

In many real-time applications such as games, the issue is typically worked around via careful design of scenes, different choice of blending functions, sparing use of transparency, or simply accepting poor visual quality.

In some cases, scenes or large parts of scenes may be rendered as a single object with a static mesh, where only the position and orientation of the object or the viewpoint is changing. This occurs often in games, scientific visualizations, etc. Dynamic scene elements such as character models are more often restricted to fully opaque geometry, which is trivial to render correctly prior to the transparent pass.

This project will look at the possibility of taking advantage of the static nature of these meshes by pre-processing them in such a way that, with minor pipeline adjustments, they can be rendered with correct ordering from all possible viewpoints. The result should be applicable to complex meshes.

2 Aims and Objectives

The two primary objectives of the project are:

- To develop a suitable algorithm to solve the above problem, proving how and where it is applicable.
- To develop an interactive application capable of demonstrating the algorithm's use cases and performance, and visualizing its operation clearly.

3 Project Deliverables

The deliverables that will be produced at the end of the project are an interactive application demonstrating the developed algorithm, and a final report describing the algorithm and evaluating the interactive application.

The interactive application should be able to render multiple transparent objects (meshes), and should run on desktop and mobile devices to evaluate the algorithm in high- and low-performance environments respectively. It may be a web application using WebGL to achieve this.

4 Work Plan

The objectives and deliverables above will be achieved through the following work plan:

Week 1 (1 Feb - 5 Feb):

- Writing initial plan.
- **Initial meeting with supervisor.**

Weeks 1 - 2 (1 Feb - 12 Feb):

- Research background and prior solutions of the problem.

Weeks 3 - 4 (15 Feb - 26 Feb):

- Create an interactive application to develop and demonstrate the proposed algorithm.
- **Milestone:** Application is functional.
- **Major review meeting with supervisor.**

Weeks 4 - 5 (22 Feb - 5 Mar):

- Implement several prior solutions to the transparency ordering problem within the application for comparison.

Weeks 6 - 7 (8 Mar - 19 Mar):

- Develop and document the proposed algorithm.
- Evaluate and prove the applicability of the algorithm.

- **Milestone:** Algorithm is fully specified.
- **Review meeting with supervisor.**

Week 8 (22 Mar - 26 Mar):

- Implement the algorithm within the application.
- Refine and improve the algorithm if necessary.
- **Milestone:** Algorithm is functional, ready to evaluate and demonstrate.

Week 9 (19 Apr - 23 Apr):

- Evaluate the performance of the algorithm in comparison with prior solutions.
- **Review meeting with supervisor.**

Weeks 10 - 12 (26 Apr - 14 May):

- Finish the write-up and finalize the report.

In addition to review meetings at milestones, there will be shorter progress meetings throughout the project.

5 References

1. Bavoil, L. and Myers, K. 2008. *Order independent transparency with dual depth peeling.* Available at: http://developer.download.nvidia.com/SDK/10.5/opengl/src/dual_depth_peeling/doc/DualDepthPeeling.pdf [Accessed: 7 February 2021].

2. Yang, J., Hensley, J., Grun, H., and Thibieroz, N. 2010. Real-time concurrent linked list construction on the GPU. *Computer Graphics Forum* 29(4), pp. 1297-1304. doi: 10.1111/j.1467-8659.2010.01725.x