Advanced Graphics 2019/2020 – Assignment 3

Introduction

For this assignment, you have considerable freedom. Assignment 1 was about light transport fundamentals and setting up the framework for basic ray tracing; assignment 2 dealt with efficient ray/scene intersection (the low-level core of any ray tracing based renderer). For assignment 3 you either build on this by implementing recent research, or alternatively, you significantly improve a low-level aspect.

Assignment

Broadly speaking, there are four areas to work on, linked to the theory presented in the lectures:

1. Acceleration structures:
   a. expanding on assignment 2, build a high quality acceleration structure for interactive rendering of dynamic scenes. The code should handle rigid animation as well as deformations and quick full rebuilds, with dedicated builders for each type of animation. A real-time demo is expected to demonstrate the capabilities of your system. This option is not available if you already built a top-level BVH for assignment 2.
   b. Implement ray packet traversal, implementing (at least) the techniques proposed by Overbeck et al. in their paper “Large Ray Packets for Real-time Whitted Ray Tracing”, or the more recent work by Fuetterling et al. (“Efficient Ray Tracing Kernels for Modern CPU Architectures”). *(note: this is CPU-only.)*

2. Physically based rendering, options:
   a. Implement the paper “Practical Path Guiding for Efficient Light-Transport Simulation” by Müller et al. to improve sample quality in difficult situations.
   b. Implement ‘PNEE’ as proposed by Andreas Mikolajewski in his master thesis.
   c. Implement a volumetric path tracer. It is sufficient to support homogeneous media; extra points for non-homogeneous media.

   The physically based rendering options require the implementation of a basic path tracer first. The path tracer by itself does not yield a passing grade.

3. Filtering & reprojection, options:
   a. Implement a filter that combines reprojection and spatial filtering to improve image quality. Suggestion: use a simple scene to make reprojection worthwhile.
   b. Implement adaptive sampling (see e.g. “A Survey of Adaptive Sampling in Realistic Image Synthesis”, M. Šik).

4. GPGPU rendering, options:
   a. Implement a basic Whitted-style ray tracer or path tracer on the GPU. This must at least support a BVH to render triangle meshes. Extra points for streaming ray tracing. *Do not use Optix or other libraries for ray/scene intersection.*
   b. Experiment with efficient GPU ray traversal, e.g. an MBVH, a rope tree or a stackless traversal scheme.

If you have a different project in mind that matches the intended scope and general topic of the course, feel free to discuss this with me. Note that if you opt for an experimental project, this does not have to yield superior results, in that case the quality of the research will be assessed.
Language Notes

This assignment may be executed in a programming language of choice. Support on the implementation side will be mostly limited to C++ and C# however, and performance is expected to be optimal for C++ code. Choice of programming language will not play a role in grading.

In general, performance will not be considered, although a fast program is more satisfying.

Practical Details

The deadline for this assignment is **Monday January 27th, 23.59**. You may hand in your assignment up to 24 hours late in exchange for a 1 point penalty. The materials to submit are:

- your project, including sources and build instructions (if these are not obvious);
- a brief report, detailing implemented functionality, division of work, references and other information relevant to grading your submission.

As with the previous assignments, you may work on this assignment alone, or with one other student.

Feel free to discuss practical details on Slack. You are not supposed to share complete ray tracers there, but if everyone uses the same optimal ingredients, that would be considered ‘research’.

Tasks & Grading

Given the increased freedom for this assignment, grading is going to be somewhat subjective. Generally speaking, a passing grade (6) for this assignment requires successful implementation of a somewhat recent paper.

To obtain additional points:

1. deliver a high-quality implementation (in terms of robustness / performance);
2. implement a relatively challenging technique;
3. combine challenges;
4. implement a technique on the GPU;
5. produce an interesting demo (nice scene or animation);
6. produce ‘publishable work’, i.e. a robust LH2 core or well-crafted custom code;
7. analyze the characteristics of your work (image quality, speed, etc.).

Obviously, many other options exist. Contact me if you want to discuss an idea of which you are not sure whether it is worthwhile.

Purpose

After successfully completing this assignment, you have obtained theoretical and practical knowledge on algorithms for efficient physically based rendering. This is a solid foundation for further research in the field of graphics.

May the Light be with you,
- Jacco.