Advanced Graphics 2021/2022 – 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) and/or path tracing. 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 five areas to work on, linked to the theory presented in the lectures:

1. Acceleration structures:
    a. Expanding on assignment 2, implement the paper "Spatial Splits in Bounding Volume Hierarchies" by Stich at al. (click for download). *Difficulty: medium/hard.*
    b. Implement Overbeck et al.'s "Large Ray Packets for Real-time Whitted Ray Tracing". The implementation must include both traversal types and must approach the gains reported by the authors. Warning: this requires SIMD programming. You may find use for some old lecture slides and a SIMD tutorial.
    c. *No other options – feel free to propose something interesting!*

2. Physically based rendering, options:
    a. Implement a basic but correct bidirectional path tracer, as described by Veach in his Ph.D. thesis (as well as in many other sources). Warning: *hard*.
    b. Implement photon mapping, as described by Henrik Wann Jensen ("Global Illumination using Photon Maps", 1996). Your implementation should correctly handle caustics. Verify your result against ground truth produced by a path tracer. *Difficulty: easy – medium.*
    c. Implement a basic but correct volumetric path tracer. It is sufficient to support homogeneous media; extra points for non-homogeneous media. *Difficulty: medium.*
    d. *Other options: feel free to propose something interesting.*

    Most of the physically based rendering options require the implementation of a path tracer first. The path tracer itself does not contribute to your 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. *Difficulty: easy.*
    b. Implement Adaptive Sampling (see e.g. "A Survey of Adaptive Sampling in Realistic Image Synthesis", M. Šik. *Difficulty: easy.*
    c. Implement a recent paper on the topic of filtering (see 5).

4. GPGPU rendering, options:

    a. Implement a Whitted-style ray tracer and/or path tracer on the GPU. This must at least use 'wavefront' / 'streaming'. The renderer must support a BVH to render triangle meshes. Important: do not use RTX/Optix or other libraries for ray/scene intersection. *Difficulty: easy, if you know GPGPU.*

5. Implement a recent paper in the field of graphics. Some options:

    a. "Path Guiding in Production", by Vorba et al., 2019, see: https://cgg.mff.cuni.cz/~jirka/path-guiding-in-production/2019/index.htm Path guiding is a class of 'learning algorithms', which estimate importance of directions over the hemisphere based on earlier transport results. *Difficulty: medium.*
    b. "Direct Ray Tracing of Smoothed and Displacement Mapped Triangles", Smits et al., 2000. The paper describes a method for directly rendering displacement maps in a ray tracer. The method is computationally expensive, which made it impractical in 2000. Now, in 2020, things have changed. *Difficulty: I think hard.*
    c. "Massively Parallel Path Space Filtering", https://arxiv.org/pdf/1902.05942.pdf , Binder et al., 2019. This approach uses world space storage rather than screen space storage for temporal accumulation. *Difficulty: medium – hard.*
    d. "Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting", Bitterli et al., 2020. As discussed in the slides: a method for importance sampling thousands of lights. *Difficulty: medium, but potentially a lot of work.*

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.

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.

Practical Details

The deadline for this assignment is **Wednesday February 2nd, 17.00**. 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 Teams. 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 (in terms of scene or animation);
6. analyze the characteristics of your implementation (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.