Prof. G. Zachmann
Maximilian Kaluschke (mxkl@uni-bremen.de)
Thomas Hudcovic (hudo@uni-bremen.de)

University of Bremen
School of Computer Science
CGVR Group
June 18, 2024

Summer Semester 2024

Assignment on Massively Parallel Algorithms - Sheet 6

Due Date June 30, 2024

Exercise 1 (Dynamic Parallelism, *8 Credits*)

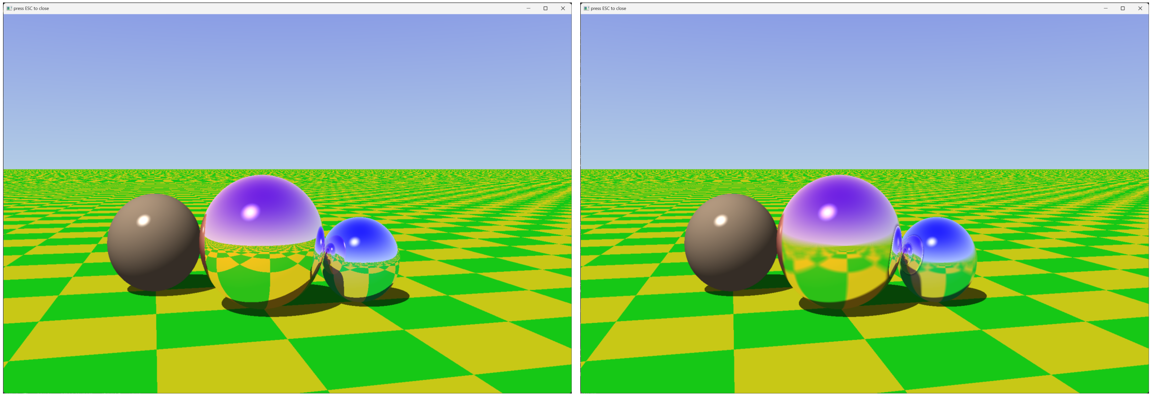

Figure 1: *Left*: Direction reflections. *Right*: Scattered reflections (with default settings).

In this assignment, your task is to enhance a parallel ray tracer with scattered reflections. The provided framework already traces a simple scene and produces an output as shown in Figure 1, left.

Your task is to implement simple, scattered reflection (see Figure 1, right). To do so, you average the color returned by multiple rays, instead of just following the direct reflection. However, during the first iteration and for scattered rays, consider the direct material color as well, since scattered rays should not cause more scattering (!) and the material color should be blended with the scattered reflection result.

In this exercise you are supposed to use dynamic parallelism (DP) for the scattered rays. Do not be discouraged if the image does not look exactly as shown here. We will (generously) consider your efforts to utilize DP during grading, even with bugs present.

Our solution takes around 17 ms to generate an image on an RTX 3090 with 16 × 16 scattering. Search for the TODO keyword in the provided source for a starting point.

- You most likely can not start a new kernel for every pixel you trace. Implement some means of limiting the number of child kernels.
- The child kernels have no access to memory from their parent. You need to implement some way for them to return their result. Remember to use atomic operations when adding the results from the child in parallel.
- Shoot the scattering rays in a dome shaped grid. You can use the thread IDs to adjust the ray direction for each scattered ray.