Exercises for the Lecture: “Architecture and Programming Models for GPUs and Coprocessors” Exercise Sheet № 6

Priv.-Doz. Dr. rer. nat. Stefan Zellmann

SS 2022

6 Parallel Programming and Ray Tracing

Ex. 6.1

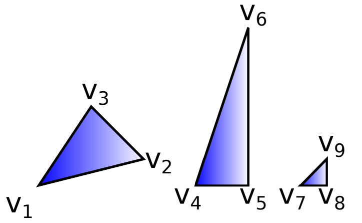
a.) Implement the serial algorithm REDUCE() from the lecture with C++11. For that you are supposed to use the file `reduce.cpp` from the template program.

b.) Implement the EREW algorithm REDUCEWORKTIME() from the lecture with C++11. For that you are allowed to use the *parallel for* construct from the file `parallel_for.h` or any other parallel-for construct from a library such as *OpenMP* or Intel’s *Threading Building Blocks*. You can make the assumption that the input length is $n = 2^k$.

Hints: In the lecture we discussed two versions of the algorithm REDUCEWORKTIME(), one of them lends itself better to a multithreading implementation than the other. As C/C++ uses zero-based indices for array accesses, in contrast to the lecture, it might be easier to use a zero-based indexing scheme to solve this exercise.

Ex. 6.2

We are given the triangles below with the following 3D vertex positions.



v_1	$(1, 1, 0)$
v_2	$(5, 2, 1)$
v_3	$(3, 4, 0)$
v_4	$(7, 1, 0)$
v_5	$(9, 1, 1)$
v_6	$(9, 7, 1)$
v_7	$(11, 1, 0)$
v_8	$(12, 1, 1)$
v_9	$(12, 2, 1)$

We previously assigned those triangles to a BVH node and now need to determine if that node should be split once more. For that, determine an optimal split plane using the *surface area heuristic* and the sweeping method along the horizontal axis. Determine the costs of every meaningful split and identify the one with the lowest costs. The costs associated with traversing the tree are the same as those associated with intersecting a ray with a triangle.

The exercise sheet will be discussed on May 18, 2022.