

Step 2. Implementing two versions of Ray-Tracing (openmp_ray.c, cuda_ray.cu) that utilizes CUDA and OpenMP, respectively.

- Look at the code [raytracing.cpp] for ray-tracing of random spheres, which is available on our class webpage, and modify the code for your purpose.
- You may assume the simplest form of Ray tracing that renders a scene with only spheres.
- Program input :
 - (i) openmp_ray.c: [number of threads] [output filename]
 - (ii) cuda_ray.cu: [output filename]
- Program output :
 - (i) print ray-tracing processing time of your program using OpenMP or CUDA
 - (ii) generate image file (format: .ppm or .bmp, image size: 2048X2048) that shows the rendering result

Execution example of cuda_ray.cu) > cuda_ray.exe result.ppm

CUDA ray tracing: 0.15 sec

[result.ppm] was generated.

Execution example of openmp_ray.c) > openmp_ray.exe 8 result.ppm

OpenMP (8 threads) ray tracing: 0.41 sec

[result.ppm] was generated.