3D Programming
2023/2024

CPU RayTracer & GPU PathTracer Project Report

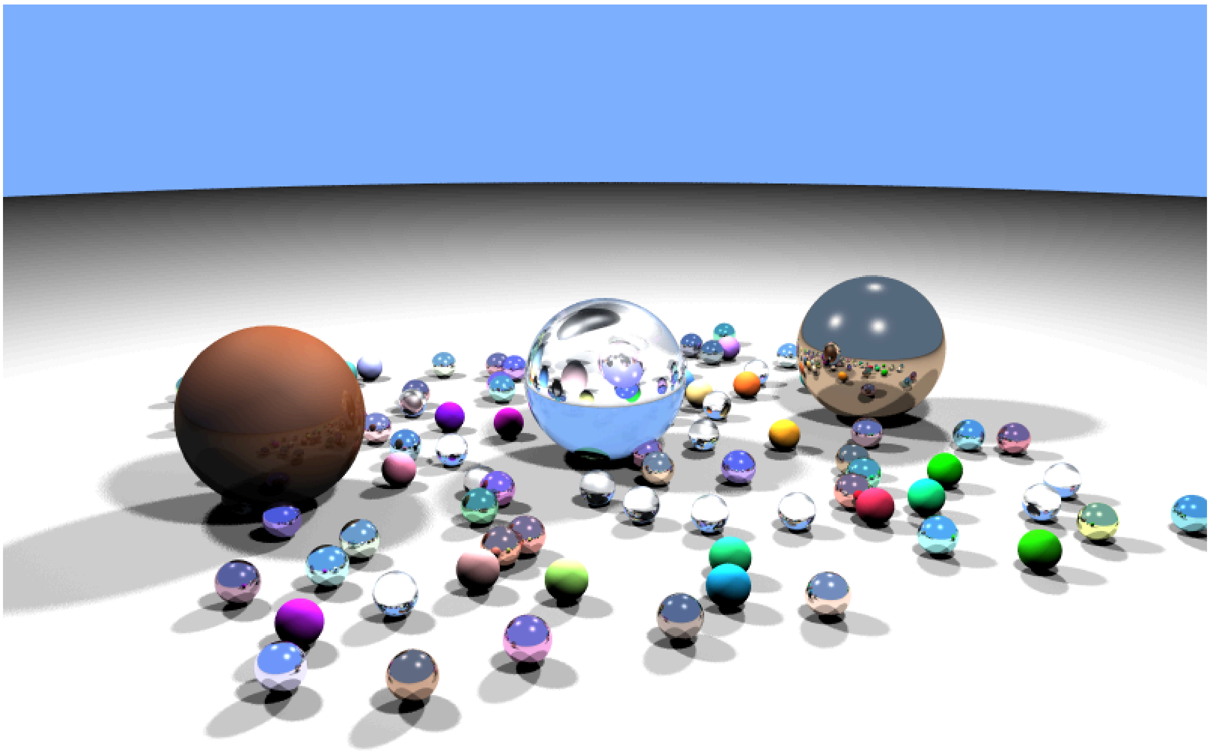

Fig. 0 - The output of our CPU Raytracer for the random scene.

Group 8
94188 Vasco Correia
95581 Gonçalo Guerreiro
95686 Vasco Sebastião

CPU RayTracer

In this part of the project, we performed a phased implementation of the RayTracer. First, we started with a naive approach by implementing the Turner Whitted Ray Tracing algorithm covered in the class. This part contemplated features such as ray intersections with spheres, triangles, and axis-aligned bounding boxes, local Blinn-Phong reflection model, multiple source lights, hard shadows, and global color component by implementing the mirror reflection and refraction with Schlick approximation of Fresnel Equations for dielectric materials.

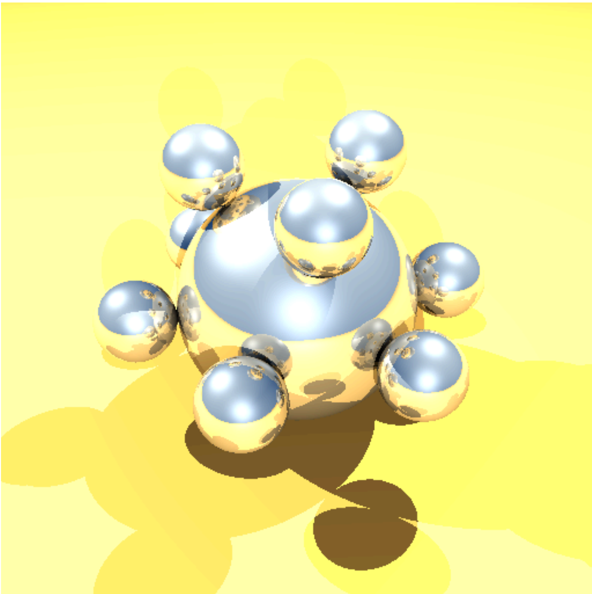

Fig. 1 - Output of our implementation given the “balls_low.p3f” file.

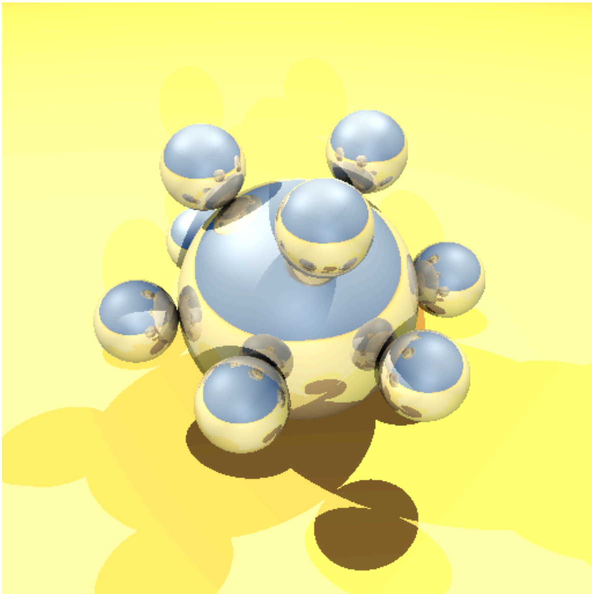

Fig. 2 - The provided result sample.

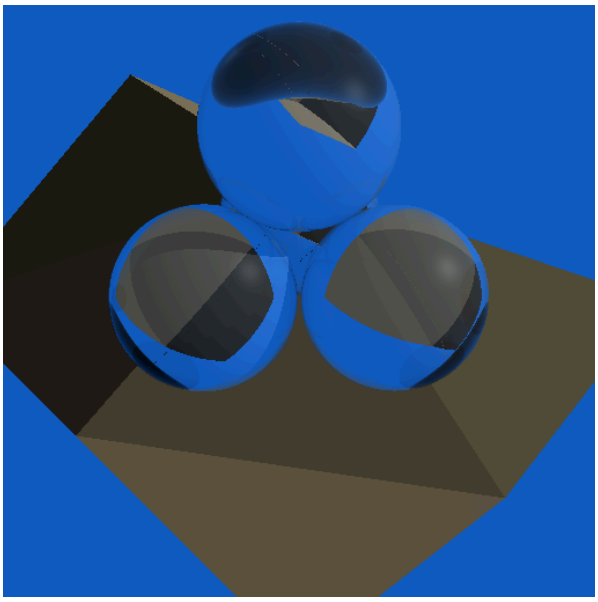

Fig. 3 - Output of our implementation given the “mount_low.p3f” file.

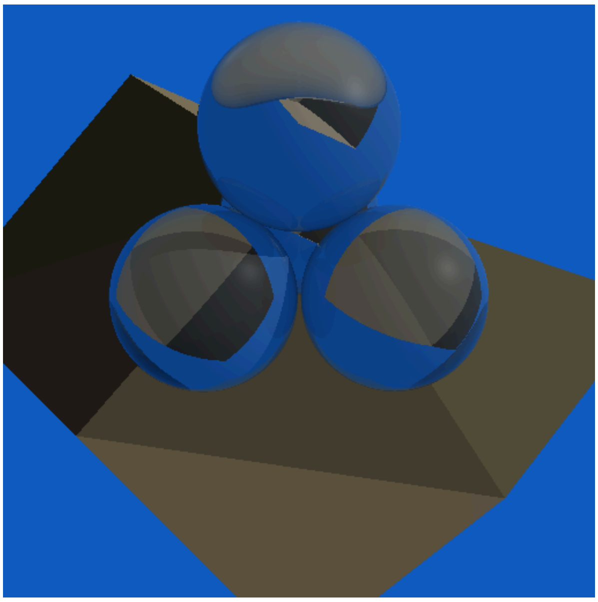

Fig. 4 - The provided result sample.

As can be seen from figures 1 to 4 our results are very similar to the expected ones, we can only point out a few differences being the brighter specular areas on the first image and a darker color on the refraction of the top sphere in the second image.
The next step was to implement the distribution ray tracing which involved the techniques of anti-aliasing, soft shadows, depth of field, and fuzzy reflections.

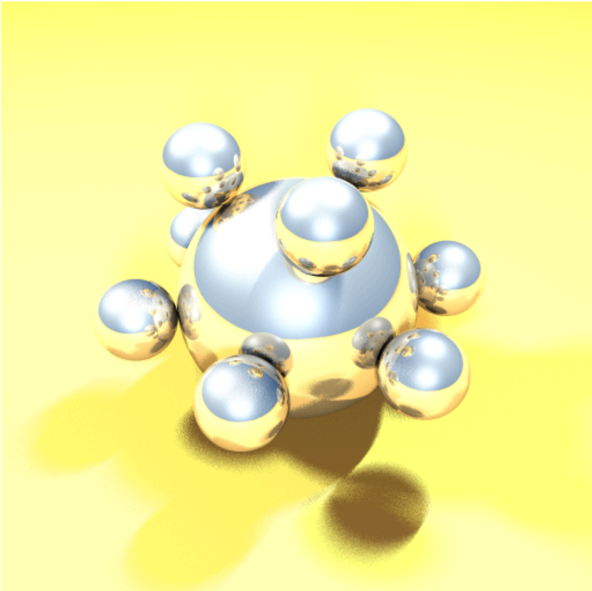

Fig. 5 - Output of our implementation given the "balls_low.p3f" file, with anti-aliasing and soft shadows (16 samples per pixel).

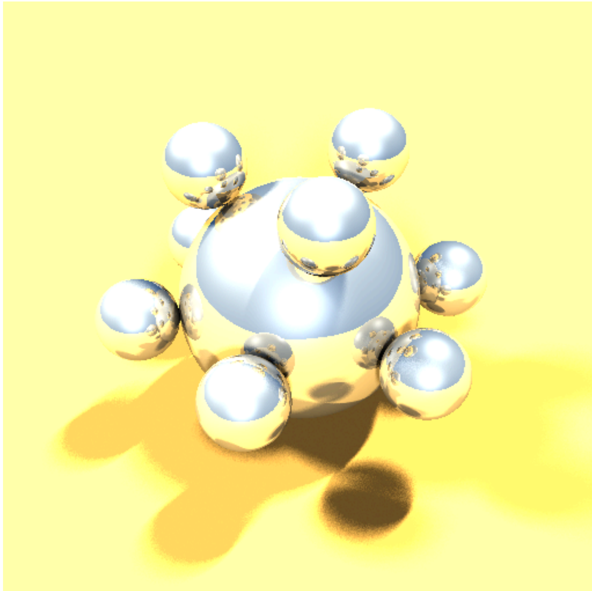

Fig. 6 - Output of our implementation given the "balls_low.p3f" file, with **no** anti-aliasing and soft shadows (16 samples per light).

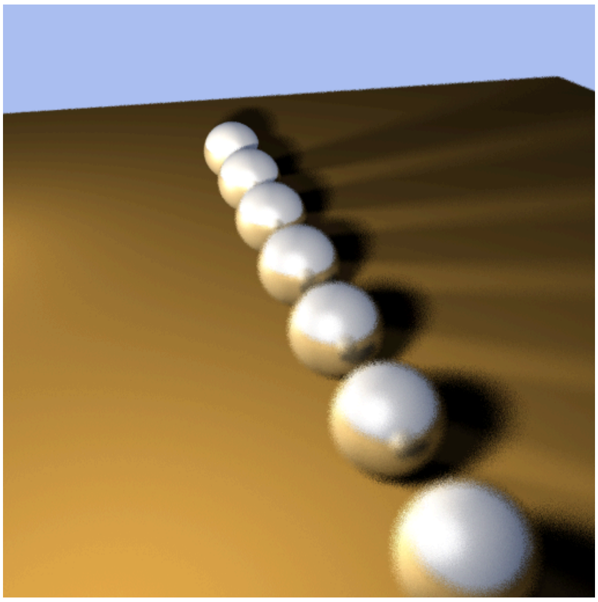

Fig. 7 - Output of our implementation given the "dof.p3f" file, with anti-aliasing, soft shadows and depth of field (16 samples per pixel, 12 aperture).

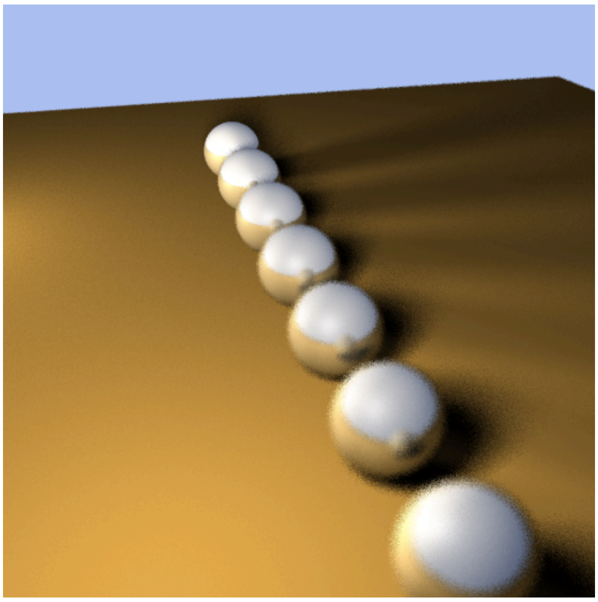

Fig. 8 - The provided result sample.

As can be seen from images 5 and 6, the anti-aliasing and soft shadows work well. Comparing images 7 and 8 we can see the depth of field is also good, only noting some differences in the shadows of the spheres. Image 9 shows the fuzzy reflections are well implemented. Images 10 and 11 are very similar except that our image has soft shadows.

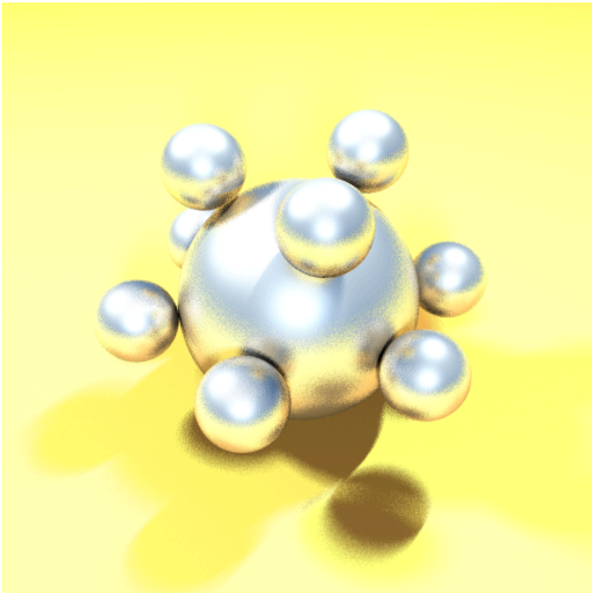

Fig. 9 - Output of our implementation given the "balls_low.p3f" file, with anti-aliasing, soft shadows and fuzzy reflections (16 samples per pixel, 0.3 roughness).

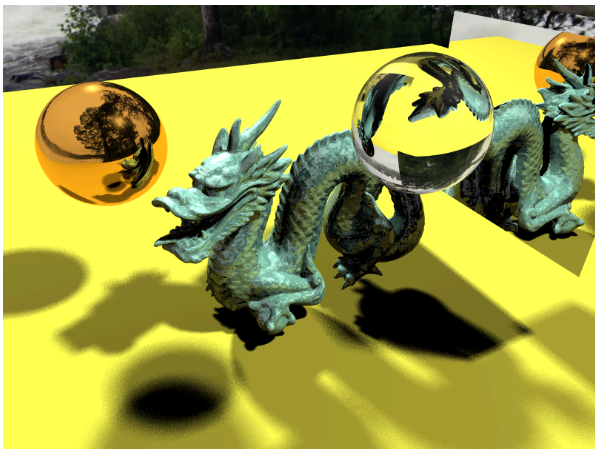

Fig. 10 - Output of our implementation given the "dragon.p3f" file, with anti-aliasing, soft shadows (16 samples per pixel).

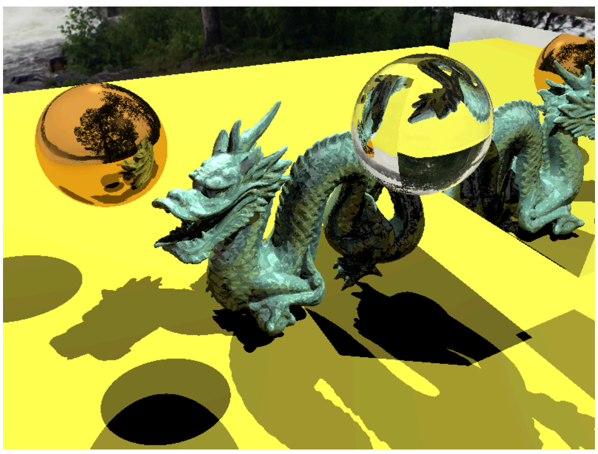

Fig. 11 - The provided result sample.

To conclude the CPU RayTracer we implemented two different acceleration structures, Uniform Grid Integration and Bounding Volume Hierarchy (BVH), that optimized the algorithm. Running the "dragon.p3f" file in non-interactive mode (with **no** anti-aliasing and hard shadows) takes on average (N = 5) 25,89 seconds using BVH, 38,87 seconds using the Uniform Grid and 18653,27 seconds without acceleration structures.

GPU PathTracer

In the second part of the project we essentially ported the implemented algorithm with a few changes to the GLSL language to implement a shader.

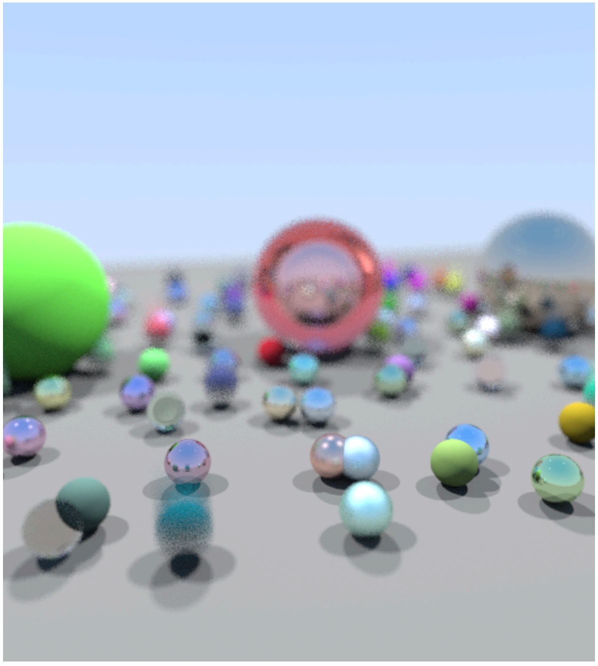

Fig. 12 - Screenshot of the render provided by our implementation of the shader with DOF.

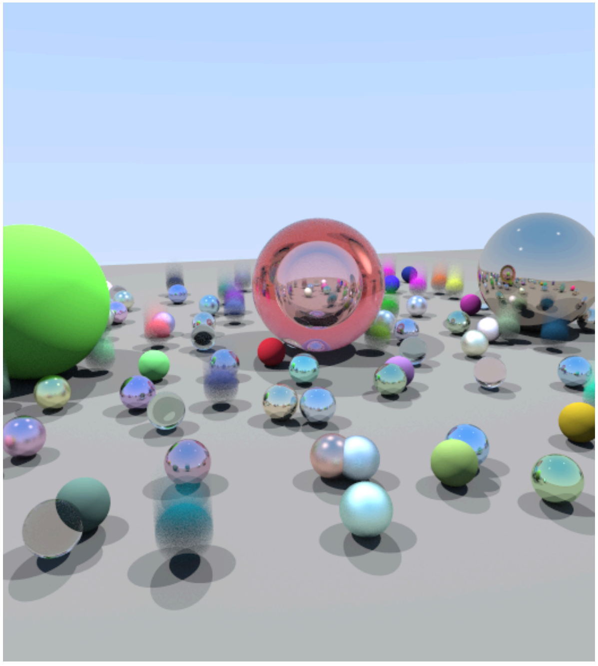

Fig. 13 - Screenshot of the render provided by our implementation of the shader without DOF.

In these images, we can see that features such as the ray intersections with spheres and triangles, calculation of local color, multiple source lights, hard Shadows, global color, mirror and fuzzy specular reflections for metallic objects, reflection and refraction with Schlick approximation of Fresnel equations for dielectric transparent materials, and depth-of-field, were well translated from the CPU RayTracer to the GPU PathTracer. Additionally, we also adequately implemented Beer's law for light absorption inside a transparent dielectric and motion blur with spheres.