Lab 4 - Ray Tracing

Laura Mazzuca - matr. 0000919489

09/09/2021

0 Assignments

1. Reflection rays
2. Hard shadows
3. Soft shadows

1 Reflection rays

In the **raytracer_students.cpp** file, the function $TraceRay()$ was enhanced with the computation of reflection rays. To do so, in the aforementioned function, a check is made to see if the point is reflective and if a number greater than 0 was specified for the bounce count. If so, the ray is computed with the function $reflectionRay()$ and then the $TraceRay()$ function gets recursively called to add its contribution to the answer while also multiplying it by the $reflectiveColor$ previously extracted from the material. The result can be seen in Figure 1a.

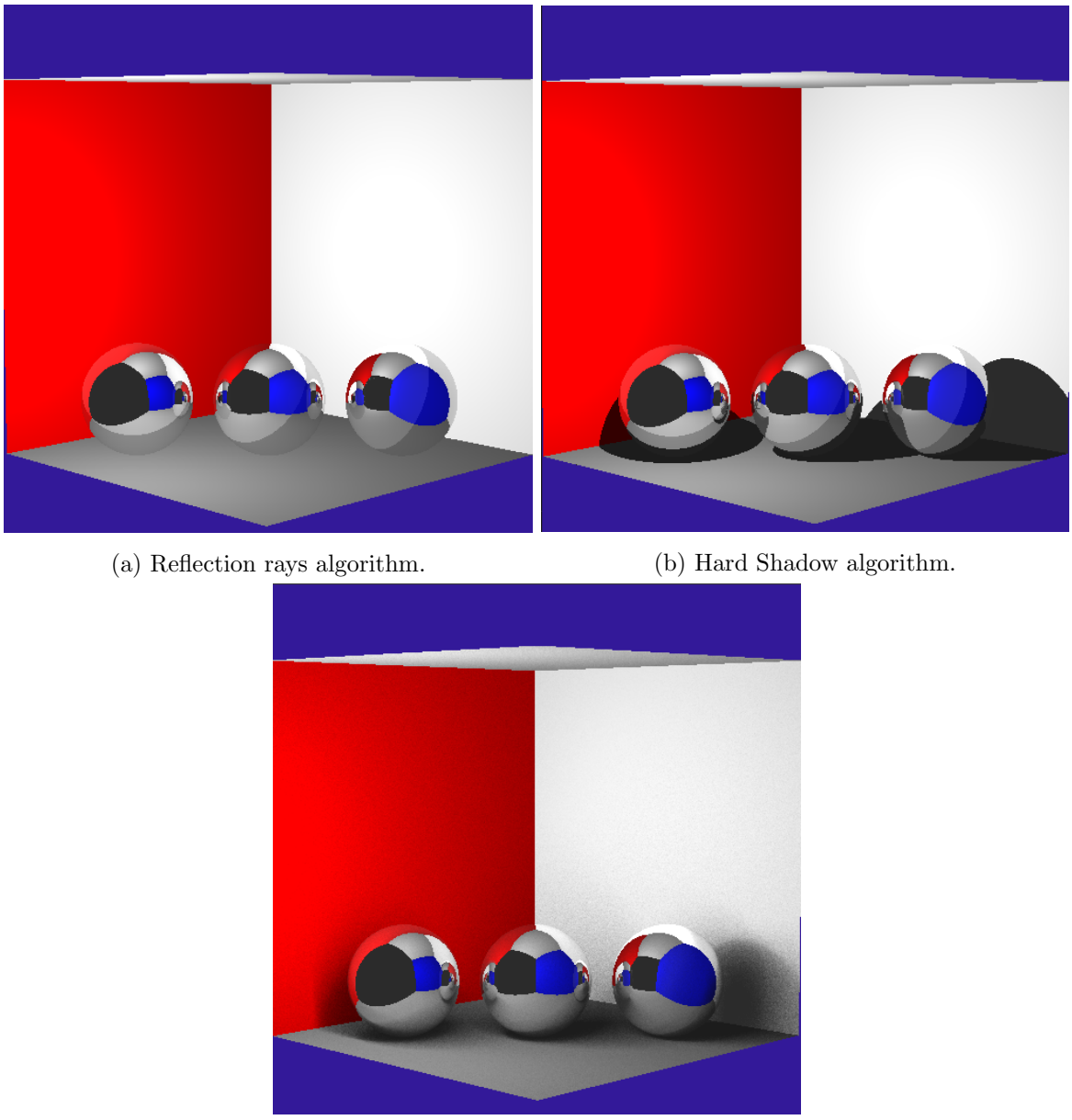
2 Hard shadows

The function $TraceRay()$ was then enhanced with the hard shadow logic. Here, we create a ray directed to the light and check the first object hit by it. If it is the light source and its luminosity is greater than 0, we compute the light's contribution with the current object's material $Shade()$ function and add it to the answer, which contains the final color of that point. The result can be seen in Figure 1b.

3 Soft shadows

Finally, the algorithm for soft shadows was implemented by sampling a number of random points on light equal to the one specified in the **-num_shadow_samples** runtime parameter. For each light

sample, the contribution is computed as described for the hard shadow algorithm, but toned down by **1/num_shadow_samples**. The result can be seen in Figure 1c.



(a) Reflection rays algorithm.

(b) Hard Shadow algorithm.

(c) Soft Shadow algorithm with 100 samples.

Figure 1: The visual results of the described solutions.