Physically based rendering

Results

Whitted-style ray tracing vs. path tracing

Figure 1 and 2 show the differences between the results of our Whitted-style ray tracer and our path tracer. We let *WhittedRenderer* and *PathRenderer* render the exact same scene, the only difference we had to make is the fact that the first image contains a point light and the second image contain an area light. This was necessary because our Whitted-style ray tracer does not support area lights, and unfortunately our path tracer does not (yet) support point lights.

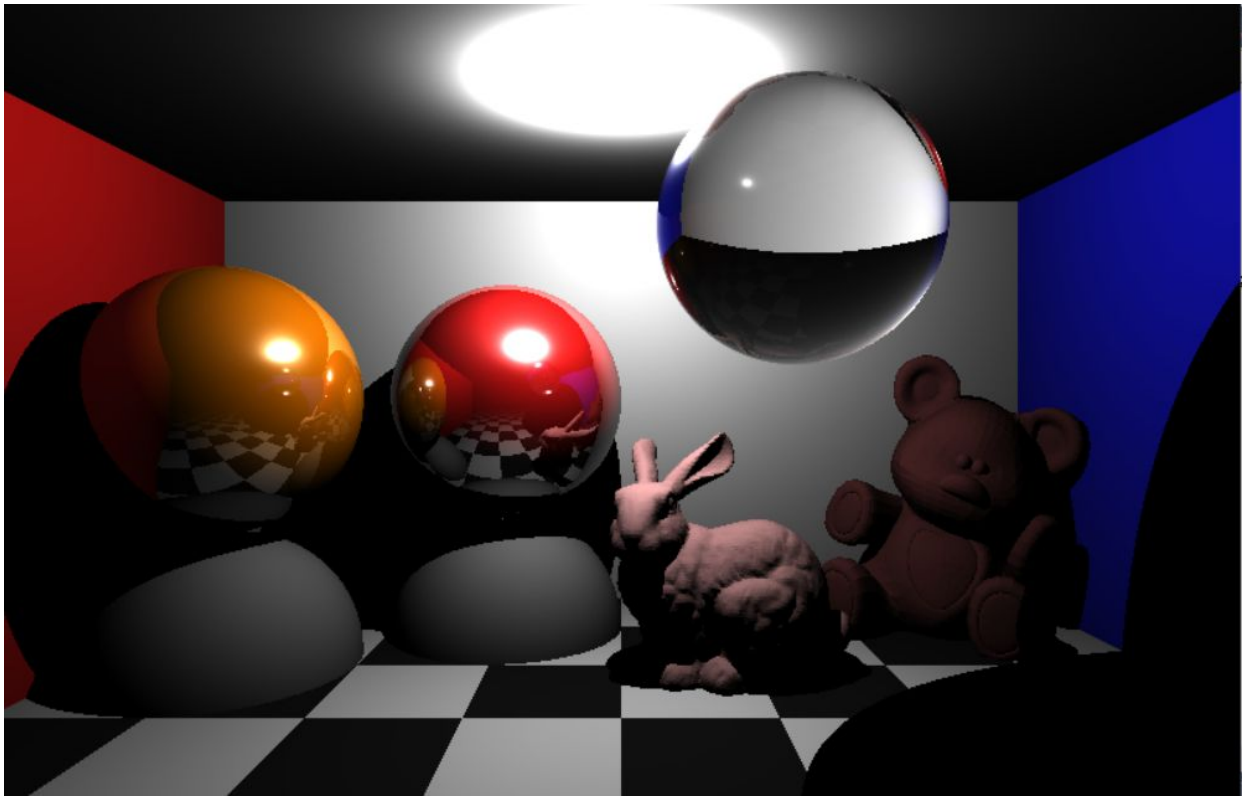

Figure 1: Scene 1 rendered with WhittedRenderer

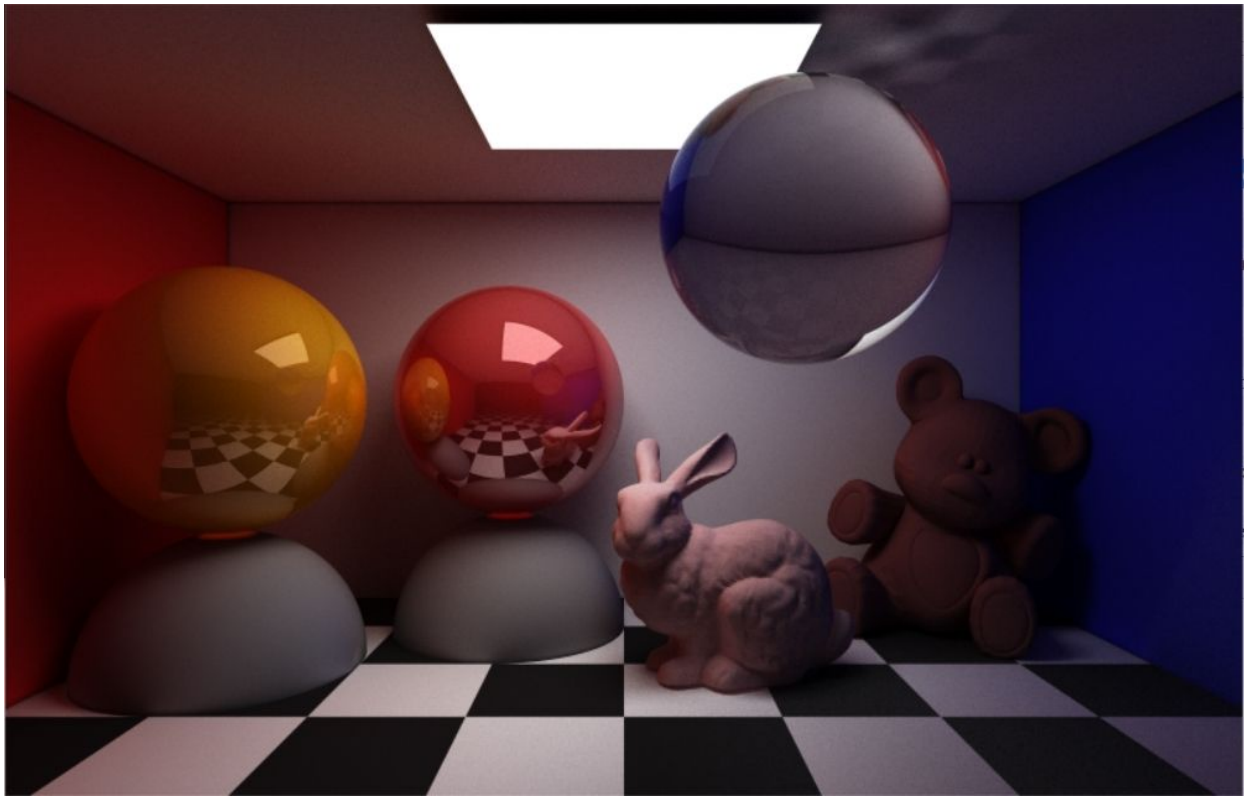

Figure 2: Scene 1 rendered with PathRenderer

Importance Sampling

Figure 3 shows the improvements we gained from Importance Sampling. For the left part of the image we used our old *diffuseReflect* function (without Importance Sampling) and for the right part of the image our *cosineWeightedDiffuseReflect* function was used (with Importance Sampling). Figure 3 is the result we get after 15 samples.

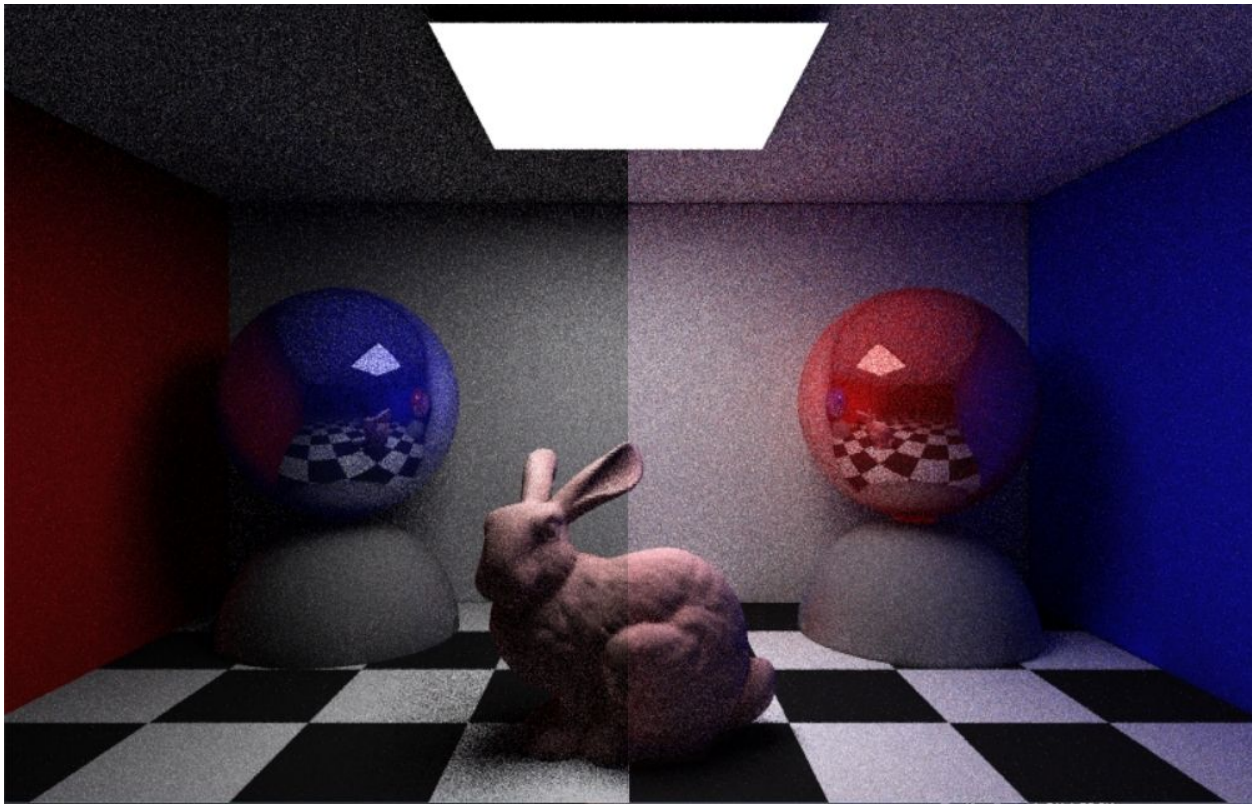

Figure 3: The result with (right) and without (left) Importance Sampling after 15 samples

Russian Roulette

Figure 4 and 5 show images that are rendered with Russian Roulette on the right side and without Russian Roulette on the left side. Figure 4 is sampled 15 times and figure 5 is sampled 100 times. As you can see, there is no (visible) difference in quality.

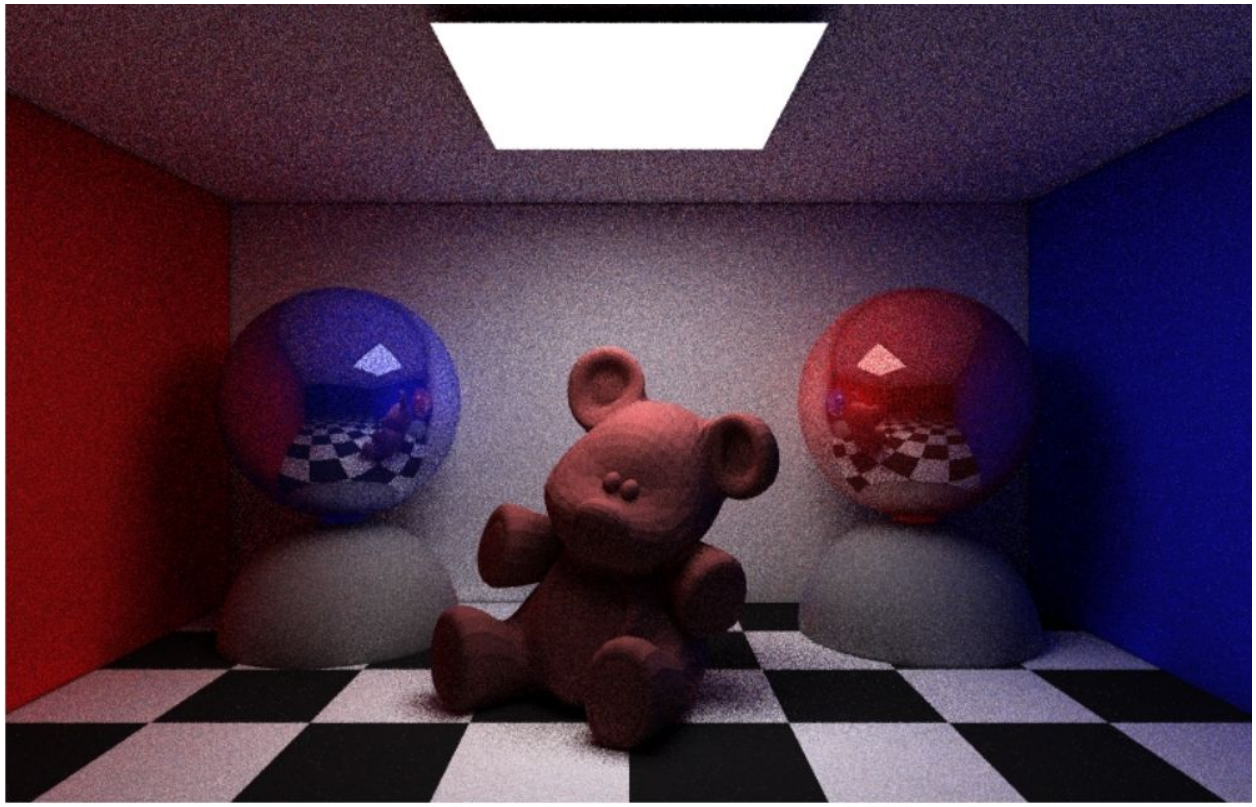

Figure 4: The result with (right) and without (left) Russian Roulette, 15 samples.

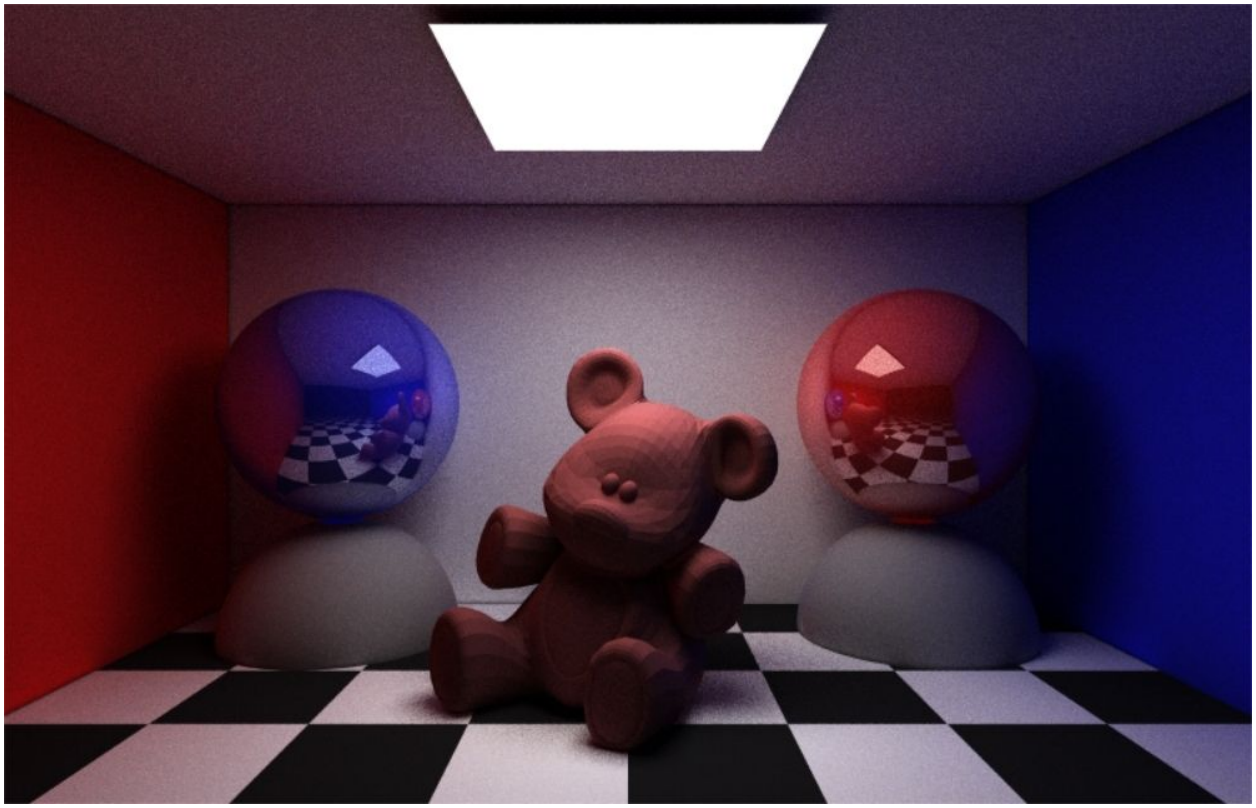

Figure 5: The result with (right) and without (left) Russian Roulette, 100 samples.

However, we can see a nice improvement in render time.

	Without Russian Roulette	**With Russian Roulette**	**Render time improvement**
15 samples	106.06 seconds	90.65 seconds	-14.5%
100 samples	730.57 seconds	607.72 seconds	-16.8%

Table 2: Render time improvements with Russian Roulette

Depth of Field

Figure 6 nicely shows our Depth of Field functionality.

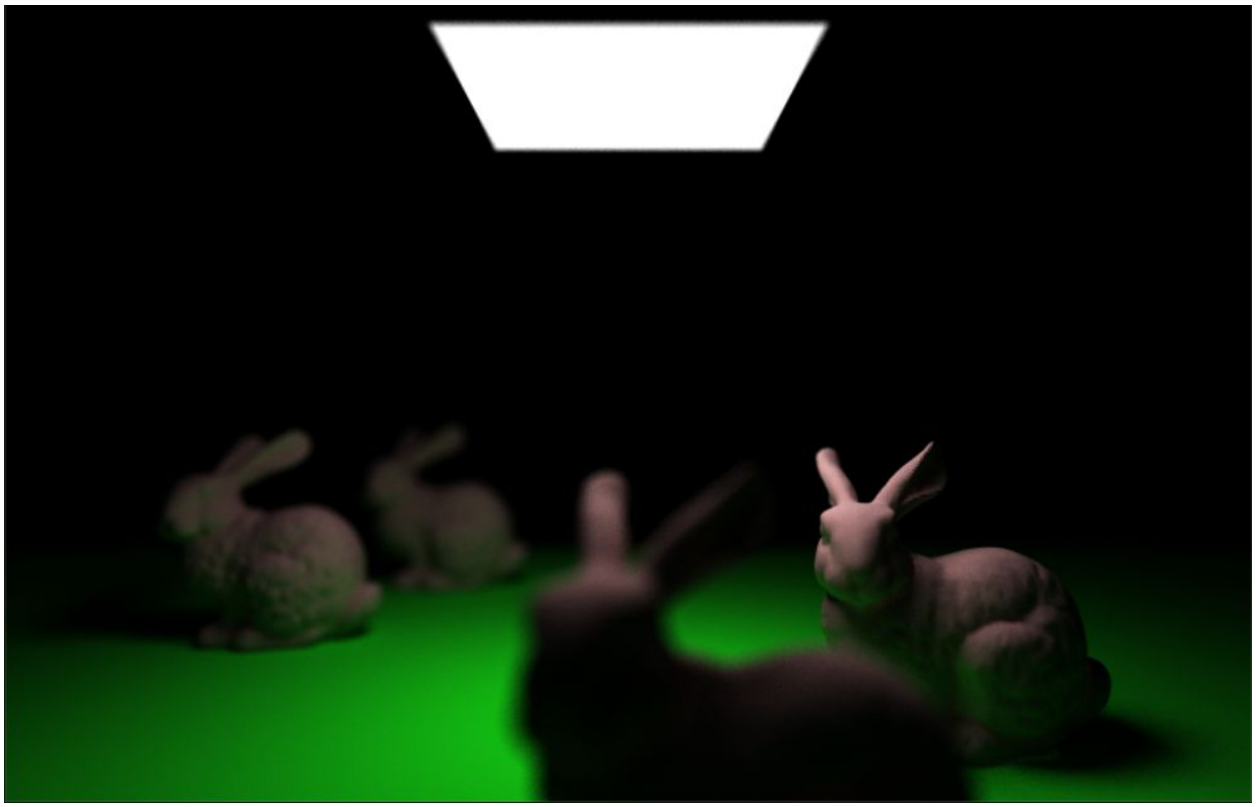

Figure 6: Field full of bunnies

Anti-aliasing

Figures 7 and 8 show the improvements we gained by adding stratified anti-aliasing to our path tracer.

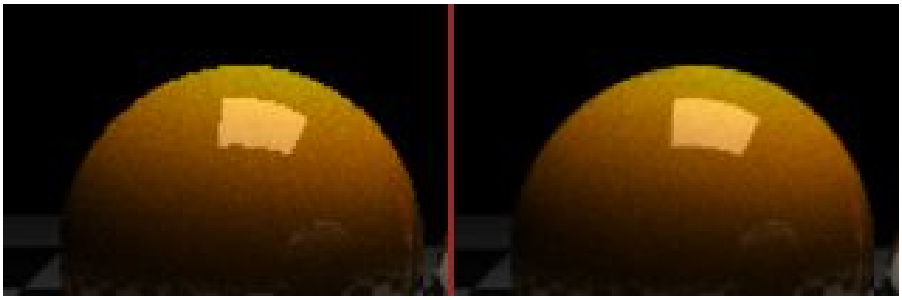

Figure 7: Comparison at sphere edge without (left) and with (right) anti-aliasing.

Figure 8: Comparison at checkerboard edge without (left) and with (right) anti-aliasing.

Final result

Finally, putting everything together...

Figure 10: Scene 2