Advanced Graphics - Assignment 3

Kwinten Jacobs 7006233 Lucas Walters 5037883

February 3, 2021

For this assignment we decided to focus on improving the path tracing algorithm more while keeping it unbiased. Thus, we worked on Adaptive Sampling, as defined by M. Šik [1], and Multiple Importance sampling.

Multiple Importance sampling is implemented in PathTracer.cs. In order to test whether it work correctly Kajiya.cs is used as a ground truth as it is a simple path tracer with only Russian Roulette added there.

Adaptive Sampling is implemented in several different ways in Camera.cs:

- Contrast based pixel error estimation proposed by Mitchell [2]
- Probability distribution based pixel error estimation with three different distance functions (f-divergence):
- Square root of Kullback-Leibler distance
- Square root of Chi-square distance
- Square root of Hellinger distance

In the implementation of Rigau et al. [3], they send 8 rays per pixel (in a stratified way) initially and repeat that process for pixels where the pixel error is estimated to be too high as defined by a predefined threshold. In our implementation, instead of sampling so much initially, we compare a pixel with its neighboring pixels. For each pixel, the error is estimated over that pixel and its (up to) 8 neighboring pixels. If this error is higher than the threshold, the pixel and all its neighbors will likely receive an extra sample on the next frame. This means that whenever a firefly is visible, what could happen is that 9 extra samples will be taken at the pixel of the firefly but also more samples of the surrounnding pixels surrounding. The occurance of the extra sampling is dependent on adaptive sampling weights multiplier and which pixel error estimator is used, and therefore may not trigger in all cases. This might seem unnecessary for fireflies but for edges of objects in the scene this method seems to work rather well.

In Mitchell's implementation, clear thresholds are given for when pixel error is considered too high, based on the relative sensitivity of the human eye. These thresholds are unfortunately missing in Rigau et al.'s implementation. For our implementation we have determined these empirically based on comparisons with the contrast pixel error estimator. 0.005 for Kullback-Leibler distance, 0.007 for Chi-square distance and 0.003 for Hellinger distance.

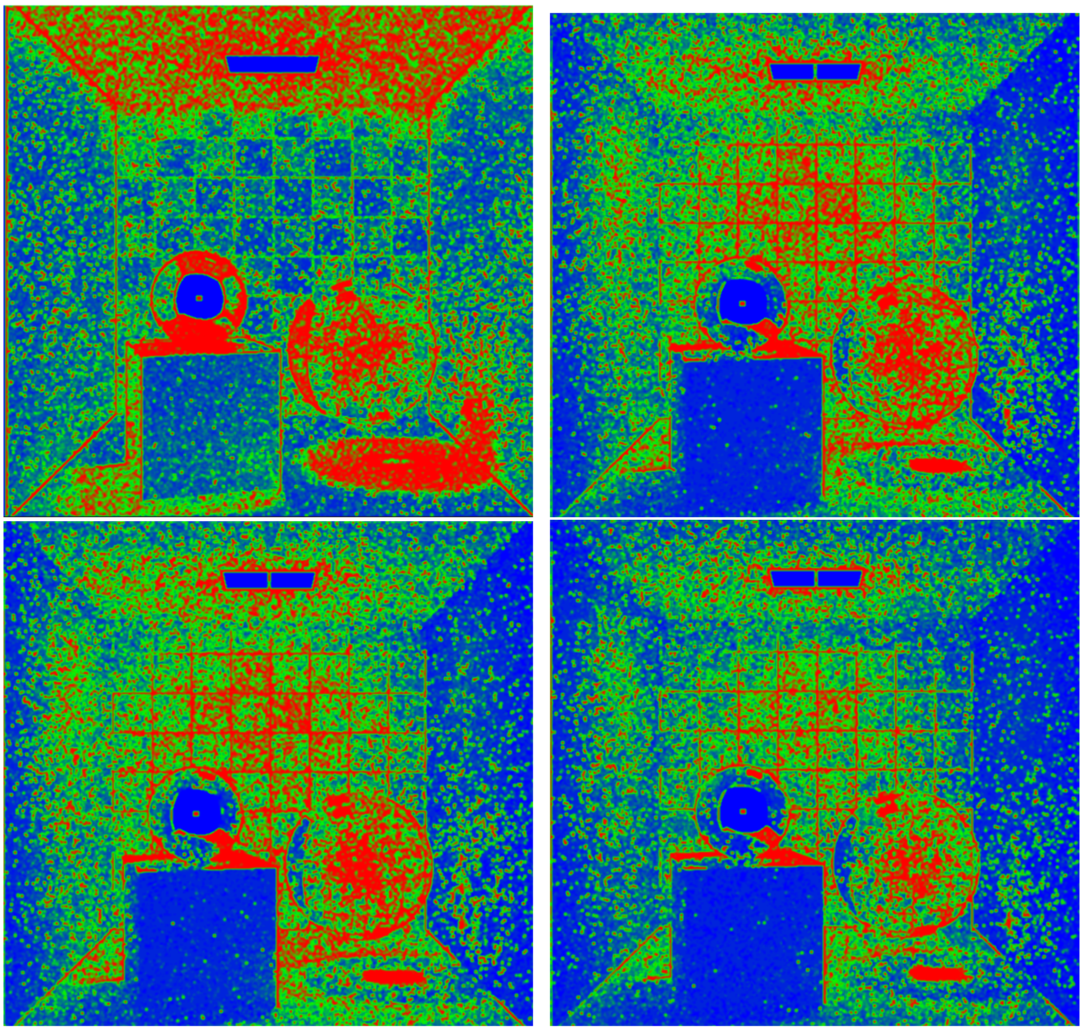

Figure 1: Heatmap of number of samples per pixel after 15 frames, red is more samples and blue is less. Top left: Contrast based pixel error estimation. The other three are pixel error estimation based on f-divergences. Top right: Square root of Kullback-Leibler distance, bottom left: Square root of Chi-square distance, bottom right: Square root of Hellinger distance.

The most interesting thing to note here is that the contrast-based solution seems to focus mostly on the indirect illumination of the ceiling while the others focus more on other parts of the scene. Another difference is that the f-divergences mostly work around the brighter areas of the checkerboard pattern in the background, while the contrast version samples the checkerboard borders quite uniformly.

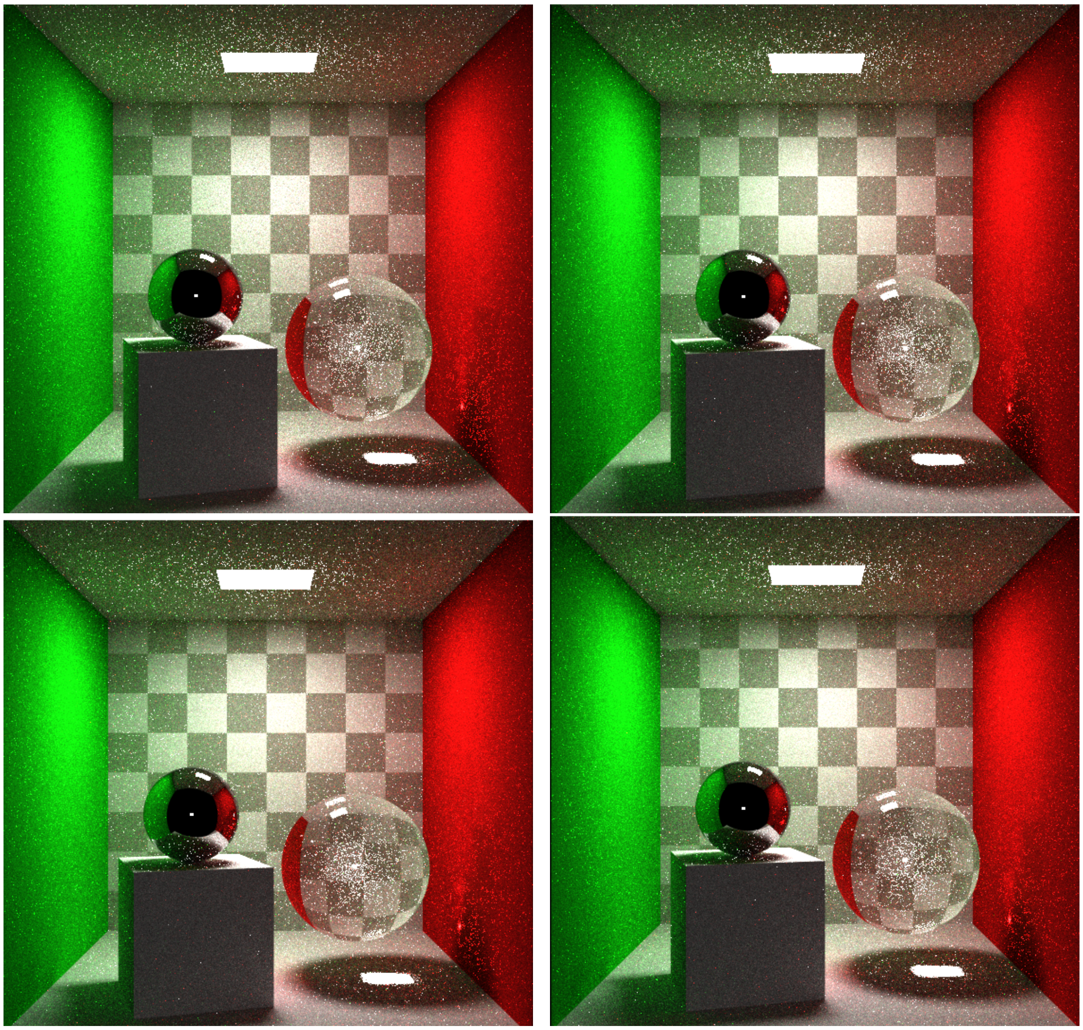

Figure 2: Actual scene after 15 frames. Top left: Contrast based pixel error estimation. The other three are pixel error estimation based on f-divergences. Top right: Square root of Kullback-Leibler distance, bottom left: Square root of Chi-square distance, bottom right: Square root of Hellinger distance.

References

[1] M. Sik, “A survey of adaptive sampling in realistic image synthesis,” 2013.

[2] D. Mitchell, “Generating antialiased images at low sampling densities,” in *SIGGRAPH ’87*, 1987.

[3] J. Rigau, M. Feixas, and M. Sbert, “Refinement criteria based on f-divergences.” 01 2003, pp. 260–269.