

Randomized Coherent Sampling for Reducing Perceptual Rendering Error

Lasse Staal Toshiya Hachisuka
Aarhus University

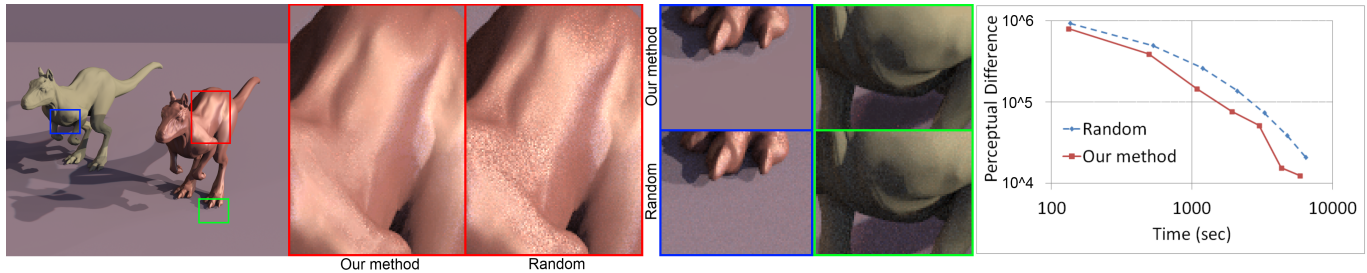


Figure 1: Left: Reference image of the killeroo model. Center top: Close-ups of the image after 8 minutes of rendering using our sampling method (RMSE 573.115). Center bottom: Equal-time comparison with random sampling (RMSE 618.509). Right: Graph of the perceptual difference of the image compared to the reference using *pdiff* [Yee 2004] with default settings.

1 Introduction

Realistic image synthesis using path tracing needs many samples to achieve noise-free images. The noise is due to the use of Monte Carlo integration in path tracing. Since Monte Carlo integration evaluates each pixel using random sampling, we obtain noisy pixels at low sample counts. Due to the random nature of Monte Carlo integration, pixel values with finite numbers of samples can be significantly different, even if their correct solutions to the rendering equation are the same.

We propose a novel sampling method which exploits the coherency among integrals of pixels without filtering. Our method does not aim to reduce RMS error of rendered images. Instead, our method makes pixels with similar integrals converge towards the correct solutions with similar trend of error. In other words, our method makes the distribution of error coherent. Figure 1 demonstrates our results. If integrals are completely the same, the image rendered by our method would be just a constant scaling from the reference solution. As the human visual system is not sensitive to absolute differences [2004], our method yields perceptually better images compared to traditional random sampling. The graph shows the perceptual error of our method compared to stratified sampling.

2 Main Idea

Instead of sampling every pixel individually, we use the same sequences of random numbers for pixels with similar integrals. This technique has already been proposed by Sadeghi et al. in coherent path tracing (CPT) [2009]. However, their main focus is to increase performance by utilizing the coherence of rays to do packet tracing. CPT suffers from structured noise in regions where the integral of the rendering equation is changing. Sadeghi et al. proposed one heuristic solution to this issue by introducing a small tile of different random number sequences over neighboring pixels. We formalize this heuristic by incorporating differences among integrals of pixels into the sampling procedure. The key difference is that we aim at rendering images with reduced perceptual error, rather than improving performance of ray tracing.

Our method generates two images; Image_R and Image_C . Image_R is generated by using regular random sampling. Using the same notation of Sadeghi et al., random sampling can be seen as use of a different random number sequence $\vec{u} = (u_1, \dots, u_n)$ per pixel [2009]. The vector $\vec{u}_{i,k}$ is used to render the k th sample for pixel x_i . CPT uses the same vector \vec{u}_k for every pixel x_i and sample k . Our method also uses the same vector over the image, but offsets it by another

weighted random number sequence for each sample. The weight indicates how different the current pixel is from its neighbors. The addition of another random number sequence reduces structured noise in regions with varying pixel integrals. The map w , containing the weights, is constructed from Image_R , by filtering the input image of path tracing. After filtering, the current pixel is compared to its neighbors in order to determine how different its pixel value is from the neighbors. The final vector used to generate the second image Image_C is:

$$\vec{u}_{i,k} = (\vec{u}_k + w_i \vec{u}_{i,k}) \bmod 1, \quad (1)$$

where $\vec{u}_{i,k}$ is the random sequence for the k th sample for pixel x_i and w_i is the weight associated with pixel x_i . If the pixel is similar to its neighbors, w_i will be 0, and $\vec{u}_{i,k}$ reduces to the coherent sequence \vec{u}_k . However, if the pixel is completely different from its neighbors, the resulting sequence will be incoherent from the neighboring pixels since the incoherent random number sequence, $\vec{u}_{i,k}$, will dominate the resulting random number sequence by Equation 1.

3 Algorithm

Our method is a two pass algorithm. In the first pass, we render Image_R using incoherent random number sequences. We then generate map w as described in the previous section. In the second pass, we utilize the resulting map to render Image_C using the vector $\vec{u}_{i,k}$ in Equation 1. The two images are combined either by addition, or a weighted function based on w , which interpolates the two images according to the differences of the integrals.

One interesting feature of our method is that the method remains unbiased since it is non-adaptive uniform sampling. This feature is in contrast to existing adaptive sampling methods where unbiasedness is usually not guaranteed due to their adaptation to previous samples. Currently the method is only explored for direct lighting. However, since we use random numbers as the basis of our method, the same approach can potentially be used for other rendering techniques such as final gathering in photon mapping. As the error of similar pixels is the same, at low sample counts, our method may still introduce structured noise. However, we see consistent reduction in perceptual error at a reasonable amount of samples.

References

- SADEGHI, I., CHEN, B., AND JENSEN, H. W. 2009. Coherent path tracing. *journal of graphics, gpu, and game tools* 14, 2, 33–43.
- YEE, H. 2004. A perceptual metric for production testing. *journal of graphics, gpu, and game tools* 9, 4, 33–40.