Randomized Coherent Sampling for Reducing Perceptual Rendering Error

Lasse Staal (staal@cs.au.dk) AARHUS UNIVERSITY Toshiya Hachisuka (toshiya@cs.au.dk)

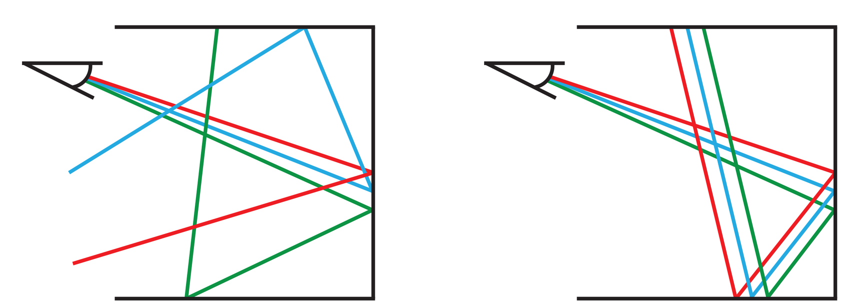
Problem

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.

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.

Related Work

Coherent Path Tracing (CPT) [Sadeghi et al.] increases the performance of Path Tracing through packet tracing. CPT uses the same sequence of random numbers for all pixels. This results in coherent paths between pixels which allows for efficient packet tracing compared to traditional sampling.



Left: Standard Path Tracing, no coherency between secondary rays. **Right:** CPT, highly coherent secondary rays.

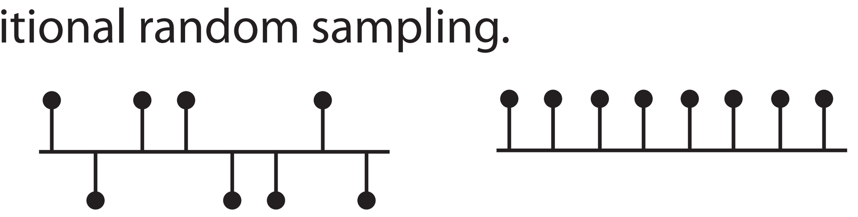
The method suffers from structured noise in regions with changing pixel integrals, which the authors alleviate through a heuristic solution.

Our Approach

Observation: Coherent paths in regions of the image where pixel values are similar, results in pixel values which converge towards the correct solutions with similar trend of error. In other words, this makes the distribution of error coherent.

We propose a novel, lightweight, sampling method which exploits this coherency amongst integrals of pixels without filtering to reduce the perceptual error of an image. Our method does not aim to reduce the absolute RMS error of rendered images.

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 [Yee, 2004], our method yields perceptually better images compared to tradiditional random sampling.



Left: The absolute error of pixel values varies between neighboring pixels in tradtional renderings. **Right:** Coherent error, with the same absolute error while reducing the perceputal error of the region.

Main idea

Random sampling can be formalized as using a different random number sequence per sample *k*, per pixel *i*:

$$\vec{u}_{i,k} = (u_1, \ldots, u_n)$$

CPT uses the same sequence $\vec{u}_k$ for the k'th sample for all pixels. We combine these sequences using a weight per pixel. This weight indicates how different a pixel is to its neighbors. A low weight results in a sequence which is coherent with its neighbors.

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

Constructing the pixel weights

To estimate the weight for each pixel, we use a an image of the scene rendered by other means. The weights are constructed as described to the right. This construction is independent of scene complexity as it is done purely in image space.

Before storing the final weight, it is tested against a threshold *t*. In practice a difference of 0 will not be obtained and the threshold is a way to set a lower limit on pixel difference. Finally *c* is a parameter controlling the strength of the incoherent sequence.

Construction:

- The image is filtered to reduce the amount of noise from the rendering.
- Find the maximum relative difference *d* between the pixel and its neighbors in RGB color space.
- The final weight for the i'th pixel is:

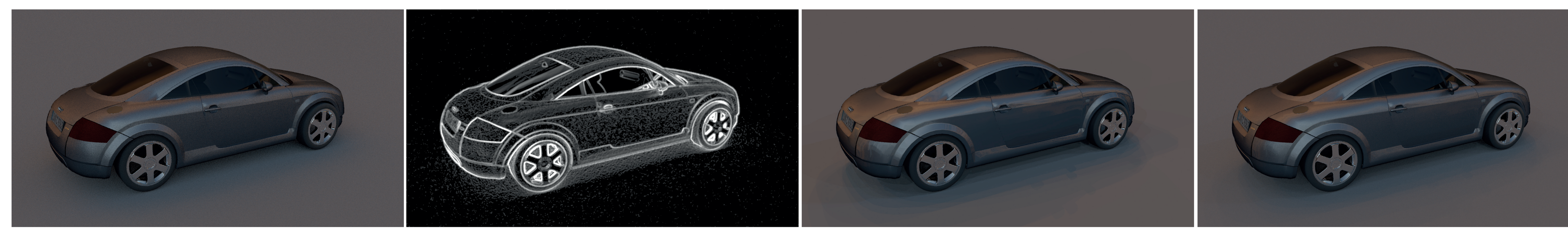
$$w_i = \begin{cases} c \cdot d & \text{if } d \geq t \\ 0 & \text{if } d < t \end{cases}$$

Algorithm Overview

The algorithm itself is done in 2 passes. First some intitial samples are generated.

The samples are used to construct the weights used in the second pass.

The two passes are combined into the final image by.



Step 1: Render image R using traditional incoherent random number sequences.

Step 2: Construct the difference weights ***w*** from image R. Here it is represented as a grayscale image.

Step 3: Render image C using the weighted coherent random sequence using the vector $\vec{uc}_{i,k}$ for the k'th sample and i'th pixel.

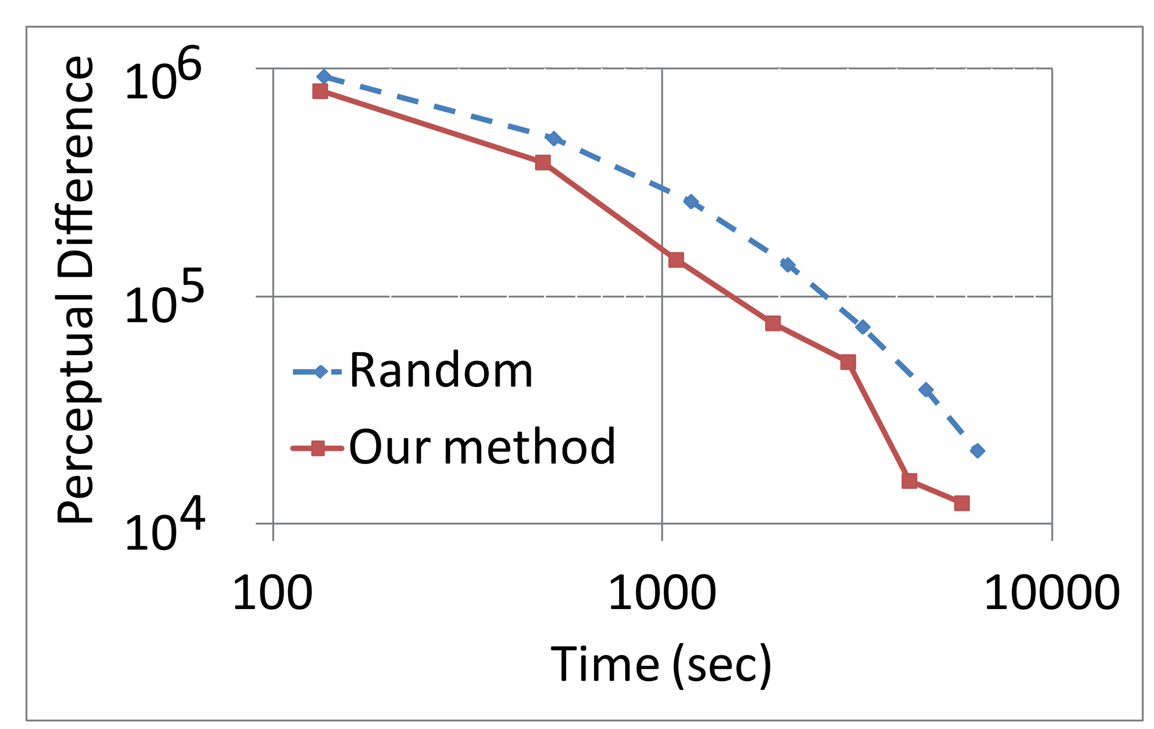
Step 4: Combine R and C using either addition or a weighted function, which interpolates the two images based on ***w***.

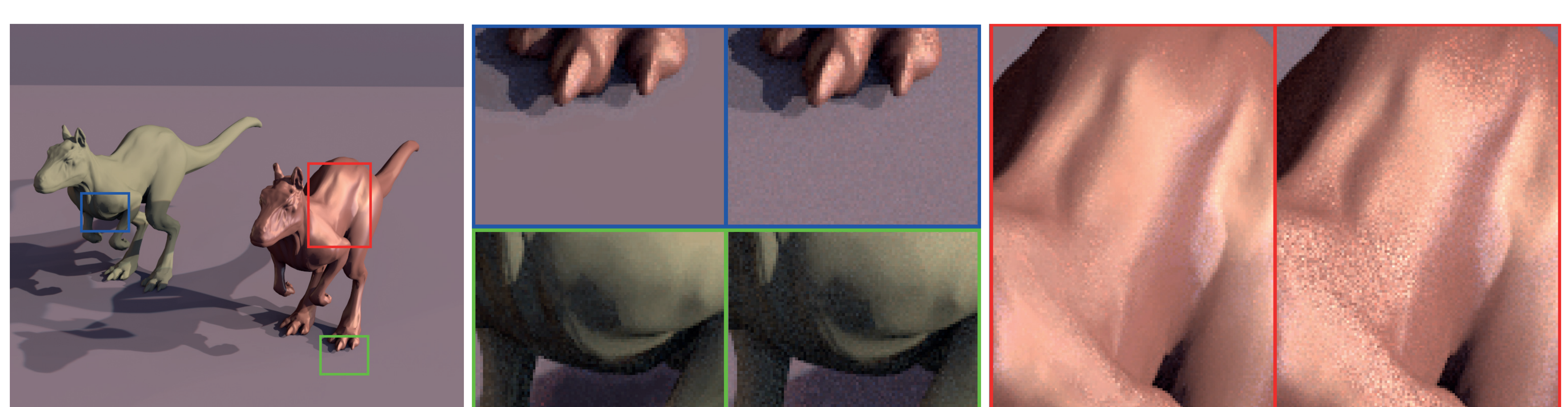
Experimental Results

Below: A rendering of the Killeroo model. The reference image is shown, with 3 highlighted regions where we compare our method to traditional random sampling.

- Equal time comparison after 8 minutes of rendering.
- Similar RMSE, at 573.1 (Our) and 618.5 (Random).
- Observable reduction of perceived error.

Left: Graph of the perceptual difference of the image compared to the reference using pdiff [Yee, 2004] with default settings.





Reference Our Method Random Our Method Random

Results

- The method is unbiased since it is non-adaptive uniform sampling.
- On average, reduces the perceptual error at equal time comparisons to traditional random sampling.
- The method can suffer from structural error in some regions and depends more on good random number sequences compared to traditional random sampling.

Future Work

- Our method is only explored for direct lighting. As it is unbiased and based on random numbers, it can potentially be applied to other rendering techniques such as final gathering in photon-mapping.
- Further exploration on how the structural error can be further reduced at lower sample counts, based on the weights in the difference map.
- Currently the method has several tuning parameters, some of which have similar effects on the method. How to combine these can be explored.