UNIVERSITÄT DES SAARLANDES
Lehrstuhl für Computer Graphik
Prof. Dr. Philipp Slusallek,
Prof. Dr. Hab. Inż. Karol Myszkowski (MPII), and
Dr. Gurprit Singh (MPII)
Assistants: Julius Kilger and
Pascal Grittmann

June 7, 2019

Realistic Image Synthesis (SS 2019)
Assignment 5

Submission deadline for the exercises: June 21, 2019

Introduction

In this assignment, you will extend your simple path tracer with next event estimation and multiple importance sampling (MIS). We provide an additional test scene that can be rendered with:

```
src/arty --algo=pt ../test/cornell_box_water.yml --width=512 --height=412
```

The code that you will have to write will still be in `src/algorithms/render_pt.cpp`.

5.1 Adding Next Event Estimation 20 + 20

Let us now improve the basic path tracer from the previous assignment, by adding Next Event Estimation.

a.) Start by computing the direct illumination at each hit point. You need to trace a single shadow ray towards a random point on a random light source. If the point is not occluded, compute the contribution of the light source along the path. The lights can be accessed via the Scene object. The Light class offers a method sample_direct() to sample a point on the light source. See `src/lights.h` for details.

Hint: When debugging, it might be useful to visualize only the direct illumination. For this purpose, disable the path continuation by terminating every time after the first bounce. If you do this, you should get the image in Figure 1, middle.

b.) Because you are currently adding the values of two estimators (Next Event Estimation, and randomly hitting the light source), the image will be biased (too bright). Since Next Event Estimation usually converges much faster than randomly hitting the light source, we should use Next Event Estimation whenever possible. However, there are some special cases which Next Event Estimation cannot cover. Find those cases, and use direct hits on light sources for them. If you implement this correctly, you should get the image in Figure 1, right.

5.2 Adding Multiple Importance Sampling 10 + 30

For some lighting effects the first version of the path tracer (using only BRDF sampling) will actually converge faster than the one with Next Event Estimation (see Figure 3). Both techniques can be combined with Multiple Importance Sampling (MIS) to render both kinds of effects efficiently.

You are going to implement MIS with the balance heuristic. In your implementation, there are two techniques (BRDF sampling, Next Event), with one sample each. Thus, the weights for the two techniques are given by:

$$w_{ne}(\omega) = \frac{p_{ne}(\omega)}{p_{ne}(\omega) + p_{brdf}(\omega)}$$

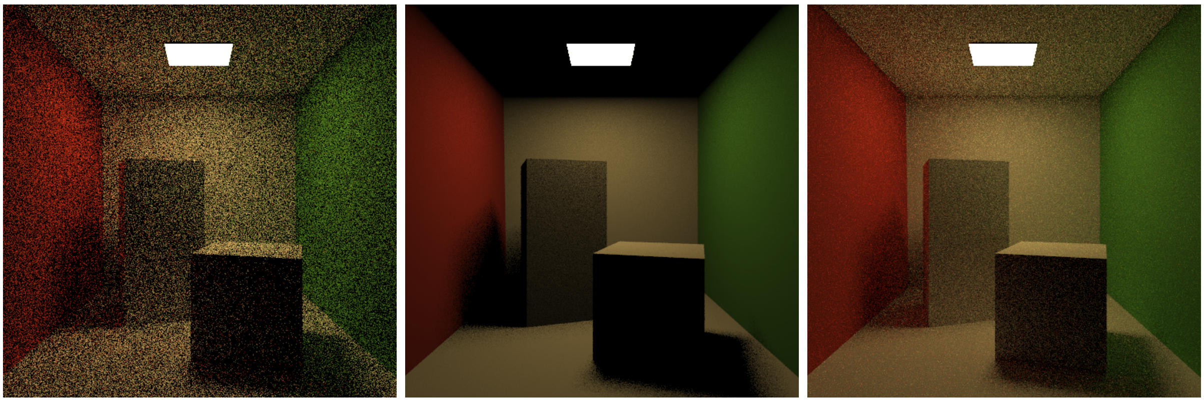

Figure 1: Images for the simple Cornell Box scene. From left to right: basic path tracer with 100 samples per pixel, direct illumination only (10 spp), and Next Event Estimation (10 spp).

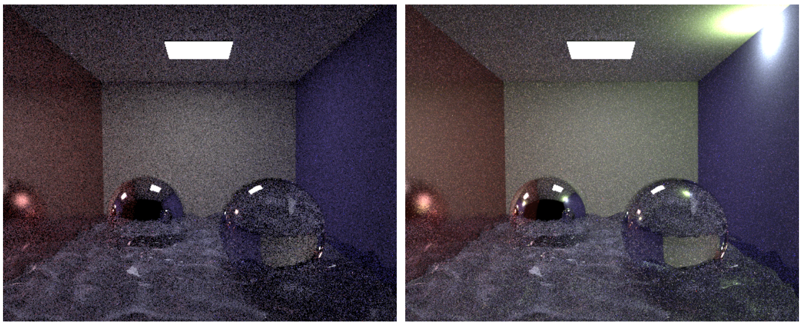

Figure 2: Scenes with materials or lights modeled with delta distributions are very challenging for a path tracer. The simple path tracer (left) cannot handle point light sources. The results are significantly improved when adding Next Event Estimation (right).

$$w_{brdf}(\omega) = \frac{p_{brdf}(\omega)}{p_{ne}(\omega) + p_{brdf}(\omega)}$$

Where ω denotes the ray that is currently considered, and ne and $brdf$ denote Next Event Estimation and BRDF sampling, respectively.

p_{ne} is the PDF (w.r.t. solid angle) for sampling the direction ω as a shadow ray. It is given by the probability of sampling the hit point x as the point on the light source, and then performing a change of variables.

$$p_{ne}(\omega) = p(x)\frac{d^2}{cos(\theta)}$$

Where d is the distance from the last point on the path to the point on the light source, and θ is the angle between the normal at x and ω. This change of variables is necessary in order to combine the two probability densities for the balance heuristic: Sampling shadow rays is done with an area density, and sampling continuation rays with a solid angle density, we hence have to express the shadow ray sampling density in solid angle (or the other way around). This ensures that the MIS weights sum to one.

The function p_{brdf} is simply the PDF for sampling the direction ω from the BRDF, and is already expressed in solid angle.

a.) Take a look at the images in figure 3. Think about the differences in the results, and write a short explanation for those. Add your explanation as a .txt or .pdf file to your submission.

b.) Implement the MIS weights in your path tracer. Keep in mind that some effects can only be sampled by directly hitting the light source, and some can only be sampled by Next Event Estimation. For those cases, make sure that the corresponding MIS weights still sum to one.

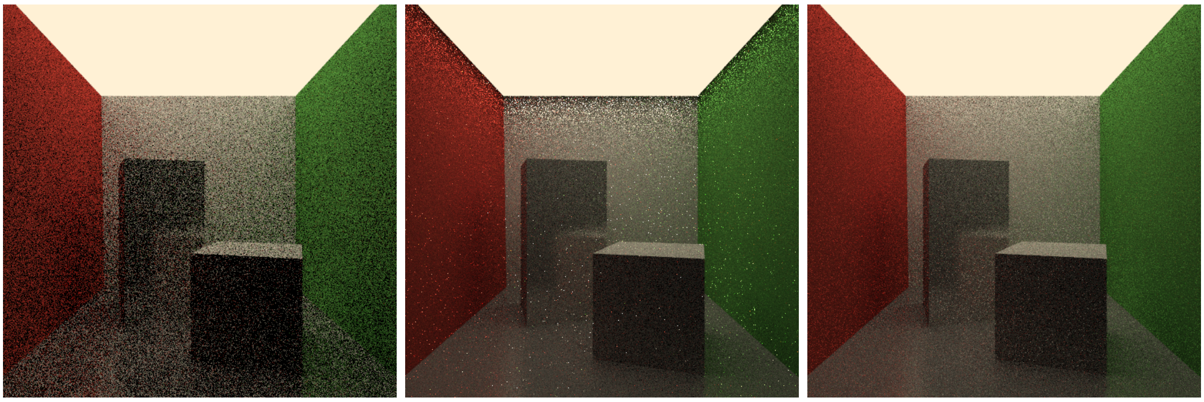

Figure 3: Images for a variation of the Cornell Box with a large light source and glossy objects. From left to right: BRDF sampling, Next Event Estimation, Multiple Importance Sampling.

Procedure of Submitting

Submit your source code per e-mail to grittmann@cg.uni-saarland.de. In case you fail to implement everything correctly, make sure to comment well your code. This will help your tutor when grading your work.

The deadline is 23:59 on June 21, 2019. Submissions late by less than two hours will be penalized with a factor of 0.8, later submissions will not be accepted. If you upload the submission, e.g., to Google Drive, you have to **provide an MD5 checksum** in your email, to verify that the file was uploaded before the deadline. If you are on a slow connection, you can also send an MD5 to ensure that the mail arrives before the deadline, and send the matching file(s) within twelve hours at no penalty.