

Tutorial 7 - Importance Sampling with PBRT

Computer Graphics

Kartic Subr and Martin Asenov
November 3, 2019

Importance sampling can make a huge difference of the quality of the rendered image. Some examples can be seen in fig.1, where importance sampling is applied based on the BSDF or the light sources.

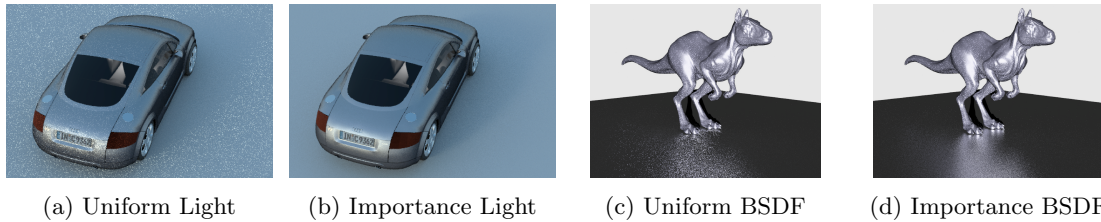


Figure 1: Possible improvement in images when using Importance Sampling. Pictures taken from PBRT book - a) and b) from [3]; c) and d) from [1]. Notice the overall reduction in noise in b) and the reduction of noise in the shadow of the kangaroo in d)

You have been provided `sampling.cpp` which can procedurally generate a scene with different number of samples and different integrators. You can compile `sample.cpp` on a DICE machine using the command:

```
g++ sampling.cpp --std=c++11 -o sampling
```

You have been provided rendered images in the data folder, which you can use directly, without having to render them again. You can use the code in `sampling.cpp` in case you want to generate more images. You have also been provided `visualize.py`, which calculates the error of the different sampled images in comparison to the reference image. You can run the Python script by executing the following command from the terminal:

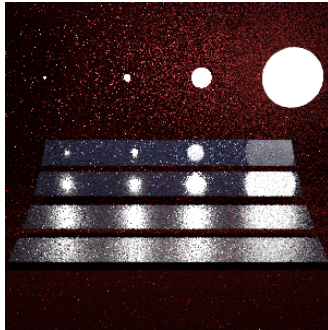
```
python3 visualize.py
```

You can read more about the different pbrt integrators and find more example rendering in [4] and [5]. In fig.2 you can find rendered sample images with importance sampling based on the BSDF, on the environment map and multiple importance sampling, and the associated error of the different integrators used with respect to the reference image.

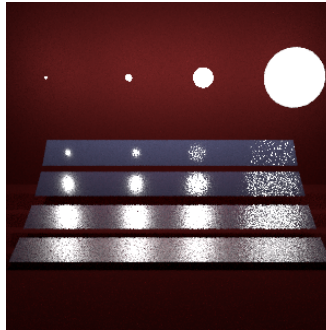
Explore the following:

1. Evaluate the performance of different integrators, with the different images from fig.3 - generate the same error graph as in fig.2. Do you see any difference in the performance? Why (not)?

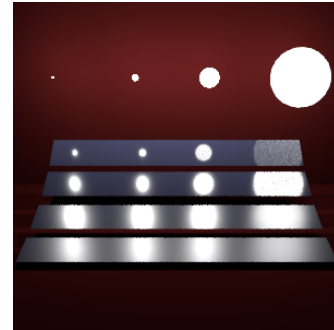
2. Why do you exhibit the specific artifacts, when using different integrators?
3. On which parts of the images do you exhibit the biggest differences when using different integrators?



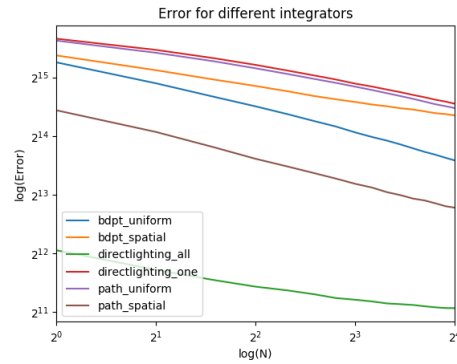
(a) Importance BSDF



(b) Importance Lights



(c) Multiple IS



(d) Sampling error

Figure 2: Possible improvement in images when using Importance Sampling. Pictures taken from PBRT book - a) and b) from [3]; c) and d) from [1]. Notice the overall reduction in noise in b) and the reduction of noise in the shadow of the kangaroo in d)

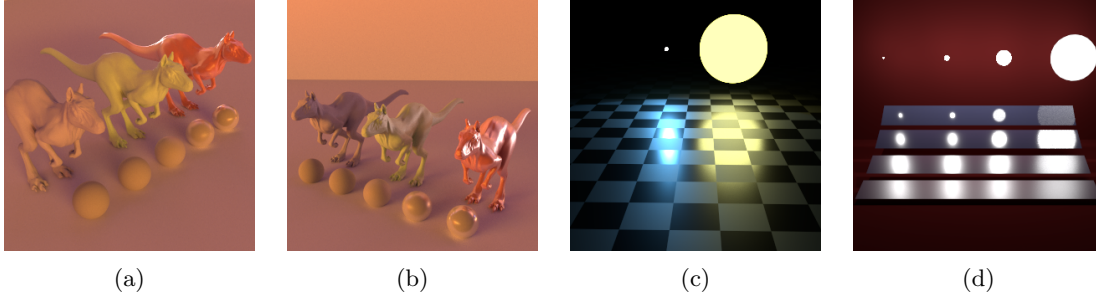


Figure 3: Explore the effect on importance sampling on those different scenes. An image with a big number of samples can be used as a reference image to evaluate the accuracy of the different sampling approaches. a), b) and c) were taken from [6]; d) is from [7] [2]

References

- [1] Bsdf importance sampling. http://www.pbr-book.org/3ed-2018/Monte_Carlo_Integration/Importance_Sampling.html.
- [2] Direct light. http://www.pbr-book.org/3ed-2018/Light_Transport_I_Surface_Reflection/Direct_Lighting.html#EstimatingtheDirectLightingIntegral.
- [3] Lights importance sampling. http://www.pbr-book.org/3ed-2018/Light_Transport_I_Surface_Reflection/Sampling_Light_Sources.html#InfiniteAreaLights.
- [4] Pbrt integrators comparison. <https://maverick.inria.fr/~Nicolas.Holzschuch/Comparison/>.
- [5] Pbrt integrators comparison 2. <https://www.janwalter.org/doc/rust/pbrt/integrators/index.html>.
- [6] Stanford assignment. <https://graphics.stanford.edu/wikis/cs348b-06/Assignment4>.
- [7] E. Veach and L. J. Guibas. Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 419–428. ACM, 1995.