# Assignment #4 – How to make light from randomness

**Assignment due date: Monday, Dec. 4, 12:00 (Noon)**

**Code to be submitted on the mathlab server by the above due date**
**This assignment can be completed individually, or by a team of 2 students**

**Student Names (Last, First)**

**Student #1:**

**Student #2:**

***Student numbers***

***Student #1:***

***Student #2:***

***Student UtorIDs***

***Student #1:***

***Student #2:***

***We hereby affirm that all the solutions we provide, both in writing and in code, for this assignment are our own. We have properly cited and noted any reference material we used to arrive at this solution, and have not shared our work with anyone else.***

_____ _____

***Student 1 signature*** ***Student 2 signature***

***(note: 3 marks penalty if any of the above information is missing)***

# Assignment #4 – How to make light from randomness

Your final assignment takes you to the current state-of-the-art in realistic computer graphics rendering. Path tracing is the most advanced method for accurate rendering that accounts for physically accurate light propagation. As such, it can handle all global illumination effects, including colour bleeding, soft shadows, caustics, and complex materials such as liquids, intervening media like smoke, and materials that exhibit sub-surface scattering.

You will implement a path tracer, and use it to render images that are rich in global illumination effects which are difficult to simulate accurately even with the advanced raytracer you worked on for A3.

***Learning Objectives – after completing this assignment you will be able to:***

Explain how the path tracing process works, and how it differs from standard Whitted ray tracing, and the more advanced distribution ray tracing.

Explain how material properties are used to determine light sampling at a surface point, and how the sampling process leads to an accurate approximation of global light transport through the scene.

Apply the idea of importance sampling, using it to reduce the amount of computation needed to approximate the illumination at a surface point.

Create a scene consisting of objects with carefully designed material properties, so as to achieve a desired visual appearance.

Provide an expert's analysis of a computer-generated image, in terms of the quality of the modeling, the richness of the illumination, and the accuracy of the light propagation method used to render it.

***Skills Developed:***

Implementing and using random sampling methods for approximating quantities that are difficult or even impossible to evaluate exactly.

Thinking of material properties in terms of light propagation and sampling distributions.

Using approximations carefully so as to reduce the variance of colour estimates, while preserving realistic global illumination effects.

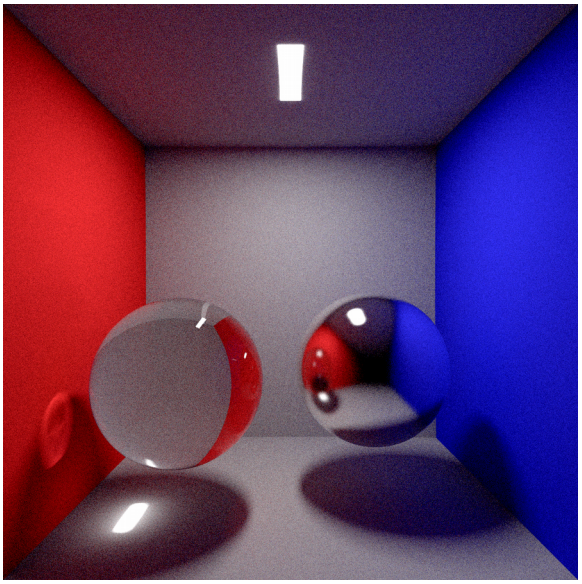
***Reference material:***

Lecture notes for all topics involved in ray tracing and path tracing.

The detailed comments in the starter code. And your implementation of the basic ray tracer (you will need code you already wrote, such as intersection testing, texture mapping, and so on)

Your course instructor! Do not wait if you run into trouble. Come to my office with any questions you may have as you work on this assignment.

***Global Illumination with Path Tracing***



*A version of the Cornell Box, a test image commonly used to test global illumination rendering methods, rendered from the solution to this assignment*

***Path Tracer [50 marks in total]***

For this assignment you will implement the core components of a path tracer. Overall, your working path tracer will be a much less complex piece of code than the advanced raytracer from A3, and yet, it will be able to render global illuminationand complex materials much more accurately.

***Required features:***

***(a) [10 marks] Path tracing core –*** Implement the ray propagation process as needed for diffuse surfaces. This includes Russian Roulette for ray propagation termination.

***(b) [5 marks] Reflection*** – Add code so your path tracer can handle reflective, and burnished reflective objects.

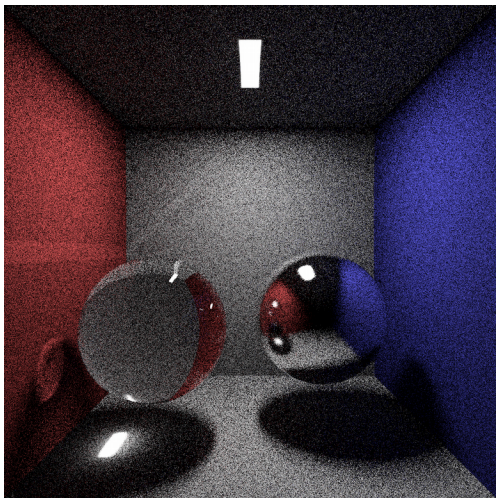
***(c) [10 marks] Refraction*** – Implement code to enable your path tracer to handle refractive objects.

***(d) [5 marks] Importance sampling –*** Modify the sampling process to use importance sampling to better approximate diffuse BRDFs

***(e) [10 marks] Explicit light sampling –*** Implement explicit light sampling to smoothly render diffuse surfaces.

***(f) [10 marks] Cool scene –*** Create and render your ultimate image, using complex objects, materials with rich properties, and showcasing global illumination.

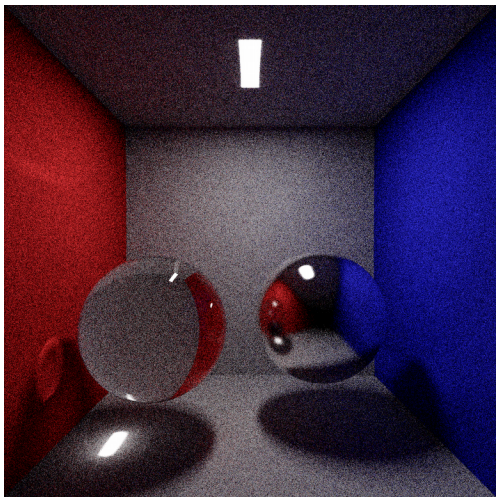
***Evolution of your path tracer***



Basic Path Tracing, including diffuse BRDF, rough specular surfaces, and refracting materials.

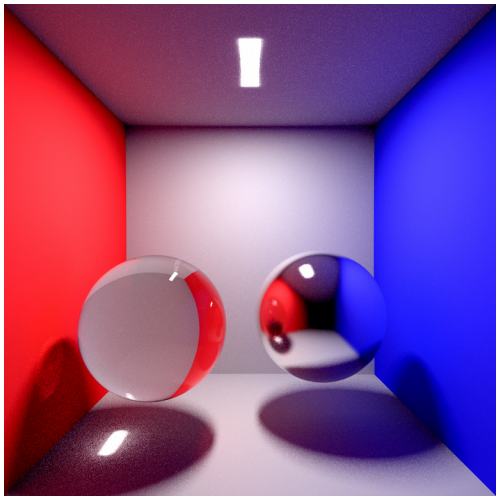
The diffuse surfaces are quite noisy since the light source is relatively small, which makes the chance of a ray hitting the light source relatively small.

However, notice the accurate rendering of caustics, and the presence of colour bleeding near the red and blue walls. The burnished metal surface shows a blurry reflection of the environment, as expected.



Improved diffuse surface rendering via importance Sampling, 1000 samples per pixel.

Here, the direction of rays bounced off diffuse materials is chosen proportional to the cosine of their angle w.r.t. the surface normal. This causes more of the sampled rays to happen along directions that have the greatest contribution to the surface's perceived colour.



Use of explicit light-source sampling to render smooth diffuse surfaces, 1000 samples per pixel.

For diffuse surfaces, the path tracer casts a ray directly toward the light source, if not blocked by another object, this will add brightness at that surface point.

As you can see this results in very smooth diffuse surfaces for the same amount of sampling.

***Crunchy Features***

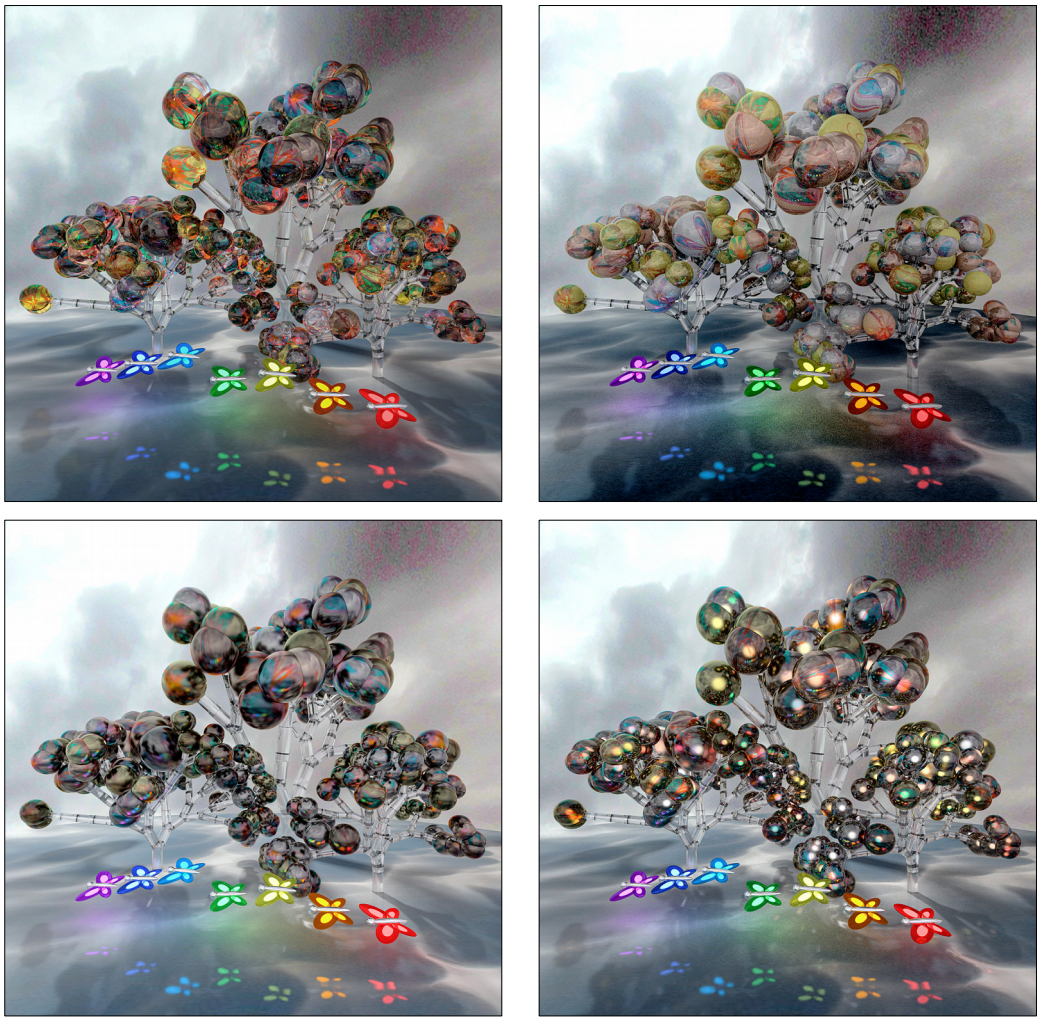
As ever, once you have a working path tracer, you can add features and render some truly impressive scenes. Some of the features you can add include:

- Depth of Field
- Hierarchical objects and procedurally generated shapes
- Texture and normal mapping
- Complex BRDFs to simulate more interesting materials.

(note that path tracing provides anti-aliasing and soft shadows by design)

***Come and talk with me if you want to implement any of these features but need a few hints about how to do this, or when you have questions as you're working on them.***

To inspire you to think in terms of what you can do with some modifications to your path tracer, have a close look at the scenes below, which were generated by tweaking the BRDF of the spheres on the trees



***What to hand in:***

- ***ALL*** your code. That means all .h and .c files as well as the compilation script
- The following images (IS is importance sampling, ES is explicit lightsource sampling)

    - Cornell Box, 512x512, 1000 samples, depth 7, no IS, no ES (***Cornell_noIS_noES.ppm***)
    - Cornell Box, 512x512, 1000 samples, depth 7, IS, no ES (***Cornell_IS_noES.ppm***)
    - Cornell Box, 512x512, 1000 samples, depth 7, IS and ES (***Cornell_IS_ES.ppm***)
    - Your final cool scene at 1024x1024, you decide the samples and depth.
      name your final scene: ***I_HAVE_CONQUERED_CG.ppm***

***Submitting your work***

Create a single compressed tar file with the name

***PathTracer_studentNumber1_studentNumber2.tgz***

From just outside your starter code directory (type it, don't copy/paste it!):

```
tar –cvfz PathTracer_11223344_55667788.tgz ./starter
submit –c cscd18f17 –a A4 –f PathTracer_11223344_55667788.tgz
```

***General advice***

You must have a fully working ray tracer from A2 to successfully complete A4. You can talk with your peers about parts of A2 that were problematic. See your TA, or come to office hours if you are stuck. However, ***you must absolutely not use any code other than your own.***

Ask questions. Don't wait if you have problems. You can drop by for hints and suggestions on how to implement any of the features of the path tracer. ***You can re-use any code you wrote for the advanced ray tracer that will help you create your cool scene, or expand your path tracer functionality.***

| Marking Scheme | |
|---|---|
| Path Tracer Implementation | 50 marks |
| No quiz! Study for the final exam! | *free* |