pbrt v3 – Direct Illumination

Luís Paulo Santos – Dep. de Informática

Universidade do Minho

April, 2022

This tutorial will briefly recapitulate fundamental concepts related to direct illumination (including the notion of stochastic sampling and Monte Carlo integration) and will study the impact on both performance and image quality of the number of samples per pixel (`spp`) and samples per light (`spl`).

The scene `cornell-T3.pbrt` is used (download it from the e-learning site). From the reference image below you can see that it is just the Cornell Box with a (very) artificial grid of polygons introduced, such that direct illumination becomes difficult to evaluate.

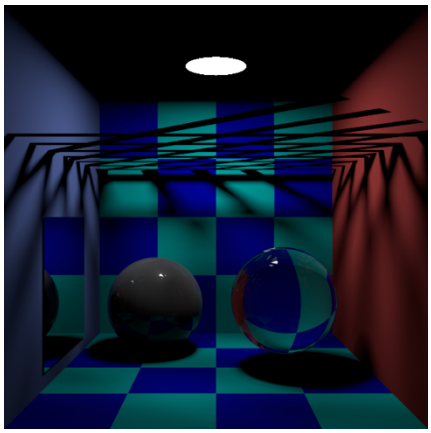

Figure 1- spp=1024, spl=4, rendering time=1209 s

Direct Illumination: recap

The evaluation of direct illumination implies sending shadow rays from the point being shaded to points on the surface of the light source. These shadow rays evaluate whether the light source id visible from that point along that direction. The contribution of the light source to the illumination of the point is proportional to the number of rays that do not intersect any geometry along the way.

Clearly the more shadow rays are shot, the higher the image quality but the rendering time also increases.

The most fundamental setup consists on shooting a single ray per pixel (`spp`=1) and then, at the intersection point, shoot one single ray per light source (`spl`=1) with directions stochastically selected over the solid angle subtended by the light source, as depicted below.

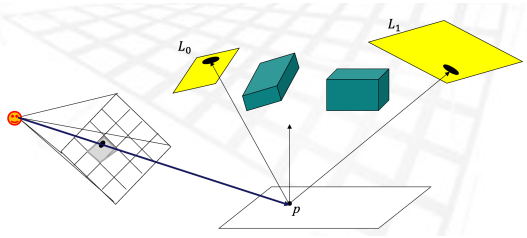

Figure 2- Direct lighting diagram: spp=1, spl=1

The above approach can result:

- in aliasing due to the very low image plane sampling rate (`spp`=1)
- noise due to the light sources very low sampling rate (`spl`=1)

One obvious solution is to increase the number of shadow rays per light source. This will reduce noise in the shadows. To render this scene we are using `pbrt's` "`directlighting`" integrator. This integrator checks whether each light source definition has a parameter called "`nsamples`". We can see in the first area light source definition how to adjust it by changing this parameter:

```
# 1st Area Light Source
TransformBegin
Translate 0 0 0.999
Rotate 180 1 0 0
AttributeBegin
  AreaLightSource "area"
       "color L"    [25 25 25]
       "integer nsamples" [1]
       #"integer nsamples" [4]
  Shape "disk" "float radius" [0.2]
AttributeEnd
TransformEnd
```

This solution, illustrated in the figure below, will reduce noise in the shadows, but will not reduce aliasing, since the image plane sampling rate doesn't change:

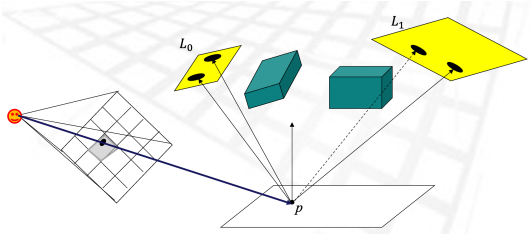

Figure 3 - Direct lighting diagram: spp=1, spl=2

A more appropriate solution might be to increase the number of primary rays per plane, therefore increasing the image plane sampling rate (`spp`):

```
#Sampler "stratified" "integer xsamples" [1] "integer ysamples"
[1] "bool jitter" ["false"]
Sampler "stratified" "integer xsamples" [2] "integer ysamples" [2]
"bool jitter" ["false"]
```

Note that each primary ray will trigger the process of sampling the light source. The parameter "`nsamples`" can therefore be set to 1, avoiding an exponential explosion on the number of shadow rays. This is illustrated in the next figure:

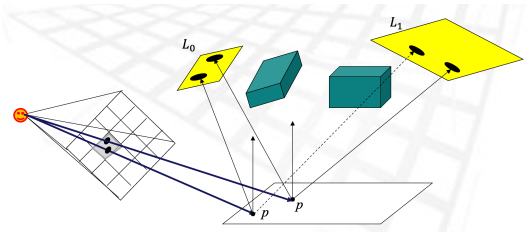

Figure 4 - Direct lighting diagram: spp=2, spl=1

Finally, up to now we had all light sources being sampled at each intersection point. An alternative is to stochastically select only one of the light sources and shoot a shadow ray towards that single light source. If it contributes (if there is no occluding geometry between the point p and the light source) then its emitted radiance is divided by the probability with which the light source was selected. The "`directlighting`" integrator supports both strategies (sampling all or just one light source) by changing the strategy parameter:

```
Integrator "directlighting" "string strategy" "all"
#Integrator "directlighting" "string strategy" "one"
```

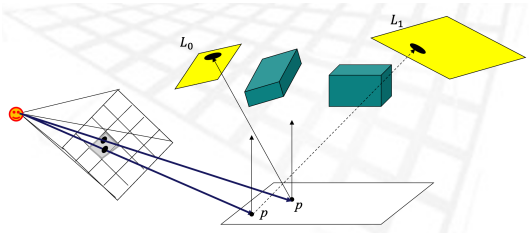

Figure 5 - spp=2, spl=1, sampling only one light source per shading point

Single point light source

If you just downloaded the file `cornell-T3.pbrt` from the e-learning site, it should be prepared to this first section. Verify that:

- the Film component specifies the correct image filename:
  `Film "image" "string filename" ["cornell-T3-1point-spp1-all.png"]`

  the Sampler specifies `spp`=1:

  `Sampler "stratified" "integer xsamples" [1] "integer ysamples" [1] "bool jitter" ["true"]`
- the integrator will sample all light sources (just one in this initial section):
  `Integrator "directlighting" "string strategy" "all"`
- The only light source which is not commented is the first point light source (there are 2 more point light sources and 2 more area light sources, but these must all be commented out:

```
# 1st Point Light Source
AttributeBegin
  LightSource "point"
        "color I" [10 10 10]
        "point from" [0 0 0.99999]
```

```
AttributeEnd
```

Run `pbrt` and visualize the image. Also fill the rendering time in Table 1.

- The shadows have very sharp edges. This are referred to as hard shadows. Given that the light source is an infinitesimally small point do you think hard shadows can be avoided?
  The image is very noisy. Increase the image plane sampling rate to `spp`=4. It is suggested that you also change the image filename:
  `Film "image" "string filename" ["cornell-T3-1point-spp4-all.png"] "integer xresolution" [600]`
- `Sampler "stratified" "integer xsamples" [2] "integer ysamples" [2] "bool jitter"["true"]`

Run `pbrt` and visualize the image. Also fill the rendering time in Table 1

- Can you comment on the noise? And the rendering time?
- Does it make any sense to increase `spl` for a point light source?

Table 1- Single point light source

spp	Time (s)
1	
4	

Multiple point light sources

Now remove the comments for the second and third point light sources:

```
# 2nd Point Light Source
AttributeBegin
  LightSource "point"
        "color I" [10 10 10]
        "point from" [0.5 0 0.99999]
AttributeEnd

# 3rd Point Light Source
AttributeBegin
  LightSource "point"
        "color I" [10 10 10]
        "point from" [-0.5 0 0.99999]
AttributeEnd
```

Also change the image file name and set `spp` to 1 again:

```
Film "image" "string filename" ["cornell-T3-3point-spp1-all.png"]
Sampler "stratified" "integer xsamples" [1] "integer ysamples" [1] "bool jitter" ["true"]
```

Run `pbrt` and visualize the image. Also fill the rendering time in Table 2.

- Can you still perceive that all shadows are hard shadows? That is the nature of point light sources.

Set `spp` to 4 and change the image filename:

```
Film "image" "string filename" ["cornell-T3-3point-spp4-all.png"]
Sampler "stratified" "integer xsamples" [2] "integer ysamples" [2] "bool jitter" ["true"]
```

Run `pbrt` and visualize the image. Also fill the rendering time in Table 2.

- Comment the noise and the rendering time comparing with the previous case.

As explained before for each ray intersection (we are using 4 rays per pixel now) shadow rays are sent towards all light sources; these are 3 shadow rays per intersection. We can reduce rendering times by telling the integrator to randomly select one light source out of all existing

light sources and shoot a shadow ray only for that one. Change the integrator strategy and the filename:

```
Film "image" "string filename" ["cornell-T3-3point-spp4-one.png"]
Integrator "directlighting" "string strategy" "one"
```

Run `pbrt` and visualize the image. Fill the rendering time in Table 2.

- Comment the noise and the rendering time comparing with the previous case. The increased noise is more perceivable everywhere in the image or only in specific places? Why?

Table 2 - Multiple point light sources

strategy	spp	Time (s)
all	1	
	4	
one	4	

Multiple area light sources

Comment the three point light sources and remove the comments from the 2 area light sources.

Now go through the same three steps as the in the previous section, registering the respective rendering times in Table 3. Here are the important changes in the scene description file for each of the three steps:

1. `spp`=1, sample all light sources:
   ```
   Film "image" "string filename" ["cornell-T3-2disk-spp1-all.png"]
   Sampler "stratified" "integer xsamples" [1] "integer ysamples" [1] "bool jitter" ["true"]
   Integrator "directlighting" "string strategy" "all"
   ```
2. `spp`=4, sample all light sources:
   ```
   Film "image" "string filename" ["cornell-T3-2disk-spp4-all.png"]
   Sampler "stratified" "integer xsamples" [2] "integer ysamples" [2] "bool jitter" ["true"]
   Integrator "directlighting" "string strategy" "all"
   ```
3. `spp`=4, sample one light source:
   ```
   Film "image" "string filename" ["cornell-T3-2disk-spp4-one.png"]
   Sampler "stratified" "integer xsamples" [2] "integer ysamples" [2] "bool jitter" ["true"]
   Integrator "directlighting" "string strategy" "one"
   ```

Compare the rendering times and the perceivable noise in the three images.

Table 3- Multiple area light sources

strategy	spp	Time (s)
all	1	
	4	
one	4	

To verify that the process actually converges (i.e. noise diminuishes) try rendering the image with `spp`=64:

```
Film "image" "string filename" ["cornell-T3-2disk-spp64-one.png"]
Sampler "stratified" "integer xsamples" [8] "integer ysamples" [8] "bool jitter" ["true"]
```