Worksheet 7

Photon mapping is biased because we use neighbouring light paths to reconstruct the illumination in vertices along a given eye path. This leads to systematic error (bias), which is not necessarily visible as stochastic noise (variance) in the rendered image. The amount of bias versus the amount of variance is referred to as the bias-variance trade-off in particle tracing techniques like photon mapping. Improving the trade-off means that the collective sum of bias and variance is reduced. The bias comes in three forms: proximity bias, boundary bias, and topological bias. Proximity bias refers to the blurring error that may occur close to illumination edges when we use neighbouring paths. Boundary bias is a darkening near geometric edges because the geometry does not support the area used for the density estimation. Finally, topological bias is when particles on a nearby surface with a different normal contribute to the illumination in the point of interest. In this worksheet, we will work with techniques for reducing these biases.

Learning Objectives

- Use density estimation techniques to improve direct visualisation of the caustics photon map.
- Use photon differentials to achieve adaptive anisotropic density estimation.
- Use photon differential splatting to avoid storing photon data in a map.
- Improve the bias-variance trade-off in rendering techniques based on density estimation.

Density Estimation

The Cornell box with a silver sphere and a glass sphere is an excellent test scene for illustrating the different types of bias. Proximity bias is directly visible along the caustic illumination edges below the glass sphere. Boundary bias is visible along the outer rim of the Cornell box. Topological bias is visible along the inner edges of the Cornell box. We will first improve the bias-variance trade-off in general and then work with the different types of bias.

- Load the Cornell box with a silver sphere and a glass sphere. Visualize the global photon map directly (no final gathering) and adjust the number photons in the global map and the number of photons in the nearest neighbour look-ups for the global map so that the result is fairly smooth while bias is fairly obvious. (In the render framework, adjust the photon mapping parameters in `RenderEngine.cpp`.)
- Use Silverman's second order kernel (also called Simpson's kernel)

$$K\left(x = \frac{\|\boldsymbol{x} - \boldsymbol{x}_p\|}{r}\right) = \begin{cases} \frac{3}{\pi}\left(1 - x^2\right)^2 & \text{for } x^2 < 1 \\ 0 & \text{otherwise} \end{cases}$$

  instead of a uniform kernel for the irradiance estimate. (In the framework, modify the `irradiance_estimate` function in `PhotonMap.h`.)
- Topological bias appears close to geometrical edges (corners) where illumination incident on a different surface mistakenly contributes to nearby surfaces with a different surface normal. To reduce this type of bias, ensure that the direction from the estimation point to the photon is nearly perpendicular to the surface normal at the estimation point. Do this by checking a dot product. (In the framework, modify the `irradiance_estimate` function in `PhotonMap.h`.)
- Render images that illustrate the effect of your new density estimation techniques and compare them to the result that did not use these techniques.
- Suggest ways of dealing with boundary bias.

Photon Differentials (optional)

Suppose we would like to render only caustics using a particle tracing technique. In this case, we can use photon differentials. Since photon differentials provide us with means to do adaptive anisotropic density estimation without looking up the nearest neighbours, we can do splatting instead of mapping. This requires a different type of tracer for the framework.

- Switch to a tracing system set up for splatting particles to eye path vertices. (In the frameork, use `EyeDiffTracer` instead of `ParticleTracer` in the Tracer section of `RenderEngine.h` and remove the initialization argument setting the maximum number of photons in the map in `RenderEngine.cpp`. The maximum number of photons to trace will now determine the number of photons used for generating caustics.)
- Find a convenient way of executing the photon differential splatting, so that the result is added to a rendering result computed with another method. (In the framework, insert the call `tracer.splat_caustics_map()` somewhere in `RenderEngine.cpp`. Make a keyboard shortcut for it or insert it in the `pathtrace` function, for example.)
- Implement emission of photon differentials. (In the framework, implement the function `emit_diff` in `AreaLight.cpp`.)
- Render images where the caustics were rendered with photon differential splatting, while other illumination is rendered with a different technique. Try to combine with path tracing without getting a doubled caustics contribution.

Worksheet 7 Deliverables

Cornell box with silver sphere and glass sphere rendered using different density estimation techniques. Please explain the technique used for the different renderings and note the render settings. Insert images and code into your lab journal. Remember to also write a suggestion on how to deal with boundary bias.

Reading Material

The curriculum for Worksheet 7 is

- Schjøth, L. Density Estimation. In *Anisotropic Density Estimation in Global Illumination*. Part 1. PhD Thesis, University of Copenhagen, 2009.
- Frisvad, J. R., Schjøth, L., Erleben, K., and Sporring, J. Photon differential splatting for rendering caustics. *Computer Graphics Forum 33*(6), pp. 252–263, September 2014.

Additional resources uploaded to DTU Inside File Sharing:

- Walter, B., Hubbard, P. M., Shirley, P., and Greenberg, D. P. Global illumination using local linear density estimation. *ACM Transactions on Graphics 16*(3), pp. 217–259, July 1997.