Jump to...

project #1
project #2
final project
assignments

project #3: Environment Lights

Assigned: 2016/11/24
Due: 2016/12/15 2:00pm

Project description

pbrt2 uses important sampling to render scenes with environment lights. Another alternative is to convert the environment light into a set of point light sources. During rendering, you can either sample these lights or use them all.

There are a few algorithms to approximate an environment light as a constellation of light sources such as the methods proposed by Kollig and Keller [2] and Ostromoukhive et al.[3]. For this project, we recommend the median cut algorithm proposed by Debevec [1] for its simplicity. This method recursively subdivides the light probe image into equal-energy regions. Finally, a representive point light is created for each region at its energy centroid. In the example of the following image, the Grace Cathedral light probe is subdivided into 64 regions with roughly equal energy values and 64 point lights are created to represent the environment light. To implement the algorithm efficiently, you could use sum area table. Since the environment light uses a latitude-longitude mapping, you need to compensate distortion caused by such a mapping. Read the 1-page paper for these details. The TA of previous semester had a slide on the project and it is worth of a read.

A light probe image is subdivided into 64 equal-energy regions. A point light is created for each region at its centroid.

The procedure to create a new class is similar to what you have done for project #1. We suggest that you copy the source files for InfiniteAreaLight, infinite.h and infinite.cpp. You could name the new class MedianCutEnvironmentLight. During construction, you need to apply the Median Cut algorithm to create nSamples virtual point lights. During rendering, you need to rewrite the Sample_L function. A reasonable way is to choose one point light from the lights you have created

with a uniform random number. Since the virtual lights represent equal-energy areas, they should have roughly the same chance to be selected. Once you have picked up a light, return its direction, energy and related information.

We have provided a test scene file. You are asked to compare pbrt's environment light implementation using importance sampling and your median cut implemenation under two probe images, grace_latlong.exr and grace-new_latlong.exr (they are in scenes/textures). Render each scene for each probe image with four different numbers of light samples, 4, 16, 64 and 256. Other than the probe image and the number of light samples, please do not change other settings. Note that your images might be darker or brighter overall than pbrt's. It is fine because it can usually ve overcome by tone mapping. Use programs such as Photomatix could solve the problem. Turn in the rendered images (there will be 16 images, two methods, two probe images and four different numbers of samples) and record the time spent on each one. Please write a html report to include your images and statistics.

Submission

You need to submit the source code files that you have created or modified, and a html report on what you have accomplished. The report should at least include 16 renderings mentioned above. Follow the instruction for submission.

Reference

1. P. Debevec, A Median Cut Algorithm for Light Probe Sampling, SIGGRAPH 2006 Sketches and Applications.
2. K. Viriyothai and P. Debevec, Variance Minimization Light Probe Sampling, SIGGRAPH 2009 Sketches and Applications.
3. T. Kollig and A. Keller, Efficient Illumination by High Dynamic Range Images, EGSR 2003.
4. V. Ostromoukhov, C. Donohue, and P.-M. Jodoin, Fast Hierarchical Importance Sampling with Blue Noise Properties, SIGGRAPH 2004.