Week 8

In the exercises for last week, you computed and stored the expansion coefficients for a spherical harmonics representation of the radiance transfer in a scene. The next step is to reconstruct lighting in real-time using this precomputed radiance transfer (PRT). One particularly nice thing about the spherical harmonics representation is that it is easy to place the scene in different lighting environments. It is, for example, possible to place the scene in a natural lighting environment by expanding a high dynamic range environment map in spherical harmonics. This is a popular application which is included in the following exercises.

Precomputed Radiance Transfer

A directional light is simple to represent in spherical harmonics. The expansion coefficients are simply the spherical harmonics basis functions for the direction toward the light. This means that we can easily compute the expansion coefficients for a directional light in real-time. And, consequently, we can interact with the light (rotate the direction of the light). The goal of the exercises in this section is to have real-time global illumination with interactive camera and directional light.

- First make sure that you can reconstruct lighting in a shader using your precomputed expansion coefficients. Implement a shader that reconstructs lighting in the same way that your ray tracer reconstructed the lighting after computing the expansion coefficients. (In the `realtime` project of the course framework, implement the `compute_sh` function in `SHOmni.cpp` and the fragment shader in `RadianceTransfer.cpp`.)
- Load the scene that resembles the scene in **SH** (`shl_room.obj`) and load the best expansion coefficients you obtained in Week 7 into your real-time shading program. Reconstruct lighting and store a sample image. (In the framework, the expansion coefficients are loaded from the file `scene.prt` when a `RadianceTransfer` shader is selected. Press '4' on the keyboard to select the shader that uses the omnidirectional SH light.)
- Next step is to reconstruct the lighting using a directional light instead of a unit background illumination. Implement a directional light that uploads this different set of coefficients to the shader which reconstructs the lighting. (In the framework, implement the `compute_sh` function in `SHDirectional.cpp`.)
- Make sure that your program is able to rotate the directional light interactively. Store three different images where the light comes from different directions. (In the framework, press '3' on the keyboard to select the shader that uses the directional SH light. Use the arrow keys to rotate the directional light.)

High Dynamic Range Imaging

To use high dynamic range environment maps, a loader for a high dynamic range image format is needed. And once the map is loaded, it must be used in the right way. The following exercises are about handling high dynamic range environment maps.

- Go to Paul Debevec's web page (see references below) and choose a light probe that you would like to use as your environment map. Download it in .hdr format and load it into your ray tracer. (In the `pathtrace` project of the framework, set the `bgtex_filename` variable in the `Globals` section of `pathtrace.cpp`.)
- The .hdr format stores high dynamic range data in the RGBE format. Implement the conversion of RGBE to floating point data. (In the framework, implement the `convert` function in `HDRTexture.cpp`.)

- A light probe is a transformed photo of mirror sphere. To use it as an environment map, you have to project a direction to a set of uv-coordinates that can be used to make a look-up in the texture. For the light probes at Paul Debevec's web page, this is done using the angular map. Implement the angular map. (In the framework, implement the `project_direction` function in `SphereTexture.cpp`.)
- Ensure that your path tracer makes a look-up into the environment map when no geometry is encountered. Load the scene that resembles the scene in **SH** (`shl_room.obj`) and path trace a reference image. Store the reference image and the view it was taken from.
- Load the same environment map into your real-time shading program and make sure that the program has the same functionality for handling high dynamic range textures. (In the `realtime` project of the framework, set the `bgtex_filename` variable in the `Globals` section of `realtime.cpp` and insert the code from the previous exercises into `HDRTexture.cpp` and `SphereTexture.cpp`.)
- Implement a shader which uses the angular map to render the high dynamic range background environment in the same way that it is rendered in the ray tracer. (In the framework, implement the fragment shader in `EnvSphere.cpp`.)
- Modify your lighting reconstruction such that it uses the high dynamic range environment map instead of unit background illumination. (In the framework, modify the `compute_sh` function in `SHOmni.cpp`.)
- Finally, load the view that you used for the reference image and render the corresponding real-time PRT approximation. Store the resulting image.

Week 8 Deliverables

Five images showing the scene (`shl_room.obj`) in different lighting environments: A shader reconstruction of the preview image from Week 7; three real-time PRT images with different directional lights; a path traced reference image with illumination from a high dynamic range environment map; and a real-time PRT approximation to the reference image. Evaluate PRT as a real-time global illumination rendering technique.

Reading Material

The curriculum for Week 8 is

SH Pages 36–46. *Spherical Harmonic Lighting: The Gritty Details*.
HD *High Dynamic Range Imaging*.
R Section 8.5. *Final Imaging Pipeline Stages*. Optional.

Please note that the u and v coordinates are incorrectly swapped around in the formula for the angular map presented in **HD**.

Additional resources:

- Peter Pike Sloan is the father of Precomputed Radiance Transfer (PRT). Check out his publications: http://www.ppsloan.org/publications/
- Paul Debevec has made many useful contributions to high dynamic range imaging. Check out his web page: http://www.debevec.org/
- Captured light probes are available at http://www.debevec.org/Probes/