Worksheet 2

Rendered outdoor environments rarely look realistic until some sort of sky model is included. The easiest way to include a sky in a rendering is to take an image of the sky and insert it as a background texture. This approach is, however, fairly limited. And, in the long run, it would take a large database of high dynamic range sky images to make it useful as a general approach. Another solution is to physically simulate the appearance of the sky. This is, however, not a trivial task. A third option is to use an analytical model. There are CIE[1] standard models for the luminance of the sky, but they are not wavelength dependent, so they are not directly useful as appearance models. Some analytical models which include colour information have been suggested in the graphics community. We will use the Preetham sky model [Preetham et al. 1999, see reference below] and also try to set up a light source which resembles direct sunlight. With a light source resembling the sun, we can use it to more realistically insert an object in a photographed environment. This is done by inserting so-called holdout geometry, which will show the background texture unless we calculate that there should be a shadow due to an inserted object.

Learning Objectives

- Using a panoramic texture to insert a background environment map.
- Render the sky using an analytical model.
- Set up a light source corresponding to direct sunlight.
- Make the rendered sky and sunlight change with location, time, and scene orientation.

Sun and Sky and Environment Maps

One difficulty in creating a sky model is the transformation from light incident on the atmosphere at various wavelengths to the corresponding tristimulus (RGB) colour that we observe on Earth. In the following exercises, you will indirectly work with this problem when setting up a light source that resembles direct sunlight. Another challenge is to have the direct sunlight match the output from the analytical model which provides the colours of the sky. In addition, these two models should depend on the location on Earth (latitude), the day of the year and the time of the day, and the orientation of the scene. The appendix in the paper by Preetham et al. [1999] is very useful in making sunlight and skylight work together.

- Load the Stanford bunny (`bunny.obj`) into your ray tracer. Render the bunny on a textured plane with a clear sky in the background.
  (CPU framework: find the Boolean variable `use_sun_and_sky` in `RenderEngine` and set it to `true`. Texture the plane by pressing 'x' before rendering. GPU framework: load `plane.obj` together with the bunny. Use the command line interface to configure the sun and sky model.)
- Render a sequence of images from sunrise to sunset where the sun rises in front of the bunny and sets behind the bunny. Use the ordinal day (the number of the day in the year), the solar time (the time of the day with 12 being solar noon), and the latitude to find the position of the sun in the sky in spherical coordinates (θ_s, ϕ_s). The illumination of the bunny should follow the position of the sun in the sky.
  (Compute the sun position and use it in the initialization of the Preetham sky model. Then change the time of day at runtime by pressing 'h' or 'H'. CPU framework: complete the function `set_time_of_day` of the file `RenderEngine.cpp`. GPU framework: complete the function `handleSunSkyUpdate`.)
- Load a latitude-longitude panoramic environment map as background image. An example file is provided on DTU Inside File Sharing in the code subfolder. This should replace the analytic sky model.
  (CPU framework: set the variable `bgtex_filename` in the `Initialization` section of `RenderEngine.cpp`. GPU framework: use the command line interface to select an image file specifying the environment map.)

[1]Commission Internationale de l'Eclairage, https://cie.co.at/

- Once the background environment texture is loaded, select a good scene configuration and store the camera settings. (Zoom by pressing 'z' or 'Z' in the frameworks to adjust how much of the background you see behind the bunny. Use the virtual trackball attached to the mouse to position the bunny in the environment. Save your camera settings by pressing 'S'. Load the camera settings after restarting the program by pressing 'L'. In the GPU framework, use 'o' and 'i' instead.)
- Insert a holdout plane and implement a holdout shader to let the loaded object cast shadows onto the ground in the environment map. (CPU framework: implement the `shade` function in `Holdout.cpp` and comment out the green plane used with the sun and sky model in the `init_tracer` function of `RenderEngine.cpp`. Insert a holdout plane instead. GPU framework: implement `__closesthit__holdout` in `shaders.cu` and switch to the `holdout` material in `plane.obj`.)
- Adjust the input parameters for the sun and sky model so that they match the sky in the environment map. In this way, the bunny will be illuminated more realistically, and the shadows will be cast in a direction corresponding to the direction that objects in the environment cast shadows. Render an image of this scene for your lab journal.

Worksheet 2 Deliverables

Renderings of the bunny in a photographed environment and on a textured plane with an analytically defined sky. These should include the corresponding direct sunlight. The scene using the analytically defined sky should be rendered every two hours from sunrise to sunset. The sun should rise in front of the bunny and set behind the bunny. Include relevant code. Please copy everything into your lab journal.

Reading Material

The curriculum for Worksheet 2 is

- Preetham, A. J., Shirley, P., and Smits, B. A practical analytic model for daylight. In *Proceedings of ACM SIGGRAPH 1999*, pp. 91–100, 1999. https://doi.org/10.1145/311535.311545
- Reinhard, E., Ward, G., Pattanaik, S., Debevec, P., Heidrich, W., and Myszkowski, K. *High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting*, second edition, Morgan Kaufmann/Elsevier, 2010. Excerpt: Sections 2–2.5, 2.10–2.12, 3-3.3.1, and 11.3-11.4.

Alternative literature:

- Ward, G. High dynamic range image encodings. In *ACM SIGGRAPH 2004 Course Notes*, 2004. https://doi.org/10.1145/1103900.1103914
- Debevec, P. Image-based lighting. *IEEE Computer Graphics and Applications 22*(2), pp. 26–34, 2002. https://doi.org/10.1109/38.988744

Additional resources:

P Sections 5–5.3. *Color*.

- Paul Debevec has made many useful contributions to high dynamic range imaging. Check out his web page: https://www.pauldebevec.com/
- Environments in latitude-longitude panoramic format are (for example) available at
  - https://polyhaven.com/hdris
  - http://www.hdrlabs.com/sibl/archive.html
  - http://dativ.at/lightprobes/index.html
  - https://vgl.ict.usc.edu/Data/HighResProbes/