Computer Graphics Assignment3

This assignment takes you through the entire pipeline for generating images. You may use any modelling program (e.g. Blender), but must use Mitsuba for rendering.

- Mitsuba: `www.mitsuba-renderer.org/`
- Blender: `www.blender.org/`
- Mitsuba Blender plug-in: `www.mitsuba-renderer.org/plugins.html`
- Mitsuba documentation: `https://www.mitsuba-renderer.org/releases/current/documentation.pdf`
- Mitsuba example scenes: `http://maverick.inria.fr/~Nicolas.Holzschuch/Comparison/`

1: Modelling (20%)

Create your own 3D scene model (preferably without participating media) that should satisfy the following constraints:

- At least one area light source.
- At least 3 smooth objects, preferably interesting (unlike sphere/cylinder).
- At least one glass (transparent) object.
- At least one dielectric material (see ch 8 of Mitsuba documentation).
- Must show visible effects due to global illumination.

2: Rendering (20%)

Render the above scene using the Mitsuba renderer as follows:

- Choose a view where all objects are visible
- Render using:
  - (a) Path tracing,
  - (b) Metropolis light transport
  - (c) Bidirectional path tracing

3: Plots (30%)

For the view chosen in 2, render a set of reference images (I_{ref1} .. I_{ref3}) using (a)-(c) with very large sample counts.

Then:

- Render (a)-(c) using N samples per pixel, where $N = 10$
- Repeat this k times, to generate k images for of (a)-(c) with $N = 10$. Using $k = 30$ will be reasonable for most scenes, but you may require more images if your scene is particularly complicated.
- Generate a mean image I_{μ_N} for each set of k images generated above.
- Calculate a variance image $I_{\sigma^2_N}$ for each pixel in I_{μ_N}
- Create a difference image $I_{D_N} = |I_{\mu_N} - I_{ref}|$
- Generate an image of the average per-pixel bias I_{B_N} from I_{D_N} taking the average of its rgb channels ($\frac{r+g+b}{3}$).
- Calculate the mean bias B_N by averaging every pixel in I_{B_N}.
- Calculate the average variance σ^2_N using a similar process for $I_{\sigma^2_N}$.

- Repeat above for $N = 64, 128, 512, 1024$. If not exactly those, numbers close to these.
- Plot (in log-log scale) the following:
  - The average bias B_N for each of the algorithms
  - The average variance σ_N^2 for each of the algorithms
- For each algorithm, compute the convergence for each pixel, and display them as a heatmap. The convergence for each pixel is the slope of the best-fit line through log-variance vs log-N. This per-pixel convergence value can be converted to a colour to render all convergence values as a heatmap image.

4: Change sampling (10%)

All the above was with random sampling. Now try any other sampling strategy and repeat 3)

5: Report (20%)

Create a short report about what you have done in the previous sections, and save is as a PDF. The report should include:

- Screenshots of the 3D model.
- 3 Rendered images for Q2.
- 3 reference images for Q3.
- 2 plots for Q3, each with 3 curves. First plot is of bias and second is variance.
- 3 heatmap images for Q3 (one for each algorithm).
- 2 plots and 1 heatmap for Q4. Similar to above.
- Describe difficulties faced in modelling

- A description of the differences in images rendered for Q2.
- An explanation of the curves in Q3 and Q4. Why are they different?
- A conclusion - what do you conclude regarding the performance of the 3 algorithms for your test scene?

Submitting your assignment:

You can submit your assignment using the `submit` command on a DICE machine:

```
submit cg cw3 ./folderWithYourAnswerFiles
```