

Rendering

CS 775: Advanced Computer Graphics - Assignment 1

Due Date: 8/2/2014 (Part 1), 15/2/2014 (Part 2)

1 Description

In this assignment you will implement a rendering system in two parts. The first part will consist of implementing a simple recursive ray-tracer. The second part will enhance this with more advanced illumination capability. Given below are details for part 1. Details for part 2 will be added later.

2 To do

1. Implement a recursive ray-tracer
 - (a) Implement ray-object intersections with the following object types: spheres, cylinder, cones and triangles.
 - (b) Implement affine transformations of objects.
 - (c) Implement the Phong Illumination Model with point lights.
 - (d) Implement computation of reflected rays, transmitted rays.
 - (e) Implement shadows.
 - (f) Implement uniform super-sampling for anti-aliasing.
 - (g) Specify the scene in a scene file and read it from that file. See minimum scene spec in the marking scheme.
 - (h) Specify a synthetic camera in the file as explained in class and construct the appropriate VCS to WCS matrix for the primary rays.
 - (i) Write the rendered image to a file in any format (PPM is a simple format. See <http://netpbm.sourceforge.net/doc/ppm.html>).
 - (j) The depth of recursion may be specified in the scene file apriori.

3 Method of Submission

1. A Tar-Gzipped archive of the complete source code (and only source code). It should compile on any Ubuntu 12.04 system.

2. A html report page on the assignment that should contain some details about what you implemented and images of the results that you generated.
3. A README file with a declaration that the assignment has been done by you and not plagiarized from anywhere, and a statement specifying if you are using any *late days* or not.
4. Submission will be through the submission portal, which will be enabled later.

4 Marking

The assignment will be marked as follows:

- Creating a scene, read from a scene file, with at least 6 objects consisting at least of 1 cone, 1 sphere, 1 cylinder and 1 object made of triangles. You are free to make more than one scene with the various objects - 10 marks
- Implementing correct ray casting (1 level ray-tracing with correct camera and image setup) - 15 marks
- Implementing correct reflection, refraction and shadows - 15 marks
- Implementing correct anti-aliasing - 10 marks
- HTML report with images of rendered results - 5 marks
- Viva during Demo - 15 marks
- Total : 80

5 Late submission policy

1. Each student gets 3 late days in total over the entire duration of the course. If a student chooses to use a late day for a submission the late submission penalty will not apply (for that day only).
2. This means, for example, if you turn in three assignments one day late, they could all be counted as on-time. Or, if you turn in a single assignment four days late, it could be considered only one day late.
3. Late days are not divisible; as soon as a submission is 1 minute late, you must use a full late day. If you are working on a task in a group, then every late day you take will cost each partner one late day.

4. When submitting an assignment, a student must state whether she is using free late days, and if so, how many. This should be stated in the README file. If you do not state it, we will assume that you are going to use the late day elsewhere and want to take the usual penalty scheme.
5. The usual penalty scheme will follow a policy of graceful degradation with a 25% penalty for each day's delay (i.e., zero marks if the assignment is more than three days late after the due date.)
6. If the late days have been all used up, then the penalty scheme will apply for each additional late day for the assignment submission.
7. Unused late days are not available for cashing, or for donation to a needy soul.