COSC 363: Computer Graphics

Assignment 2

Max. Marks: 20
Due: 11:55pm, Friday 31st May 2019

Aim

In this assignment you will implement a ray tracer that can handle different types of geometric objects and global illumination features, and demonstrate its capability in enhancing the visual realism of a rendered scene.

1. The Basic Ray Tracer

In labs 7, 8, you will implement a simple ray tracer that can handle scenes containing planes and spheres. You will also implement methods for generating shadows and reflections. This assignment builds upon that ray tracer. As a minimum, your ray tracer should include the following features/objects.

(a) Lighting: The scene must include at least one light source and also diffuse, specular reflections generated by the source.

(b) Shadows.

(c) Reflections: the scene must include at least one reflective sphere.

(d) The scene must include at least one box (cube or other parallelepiped). The box need not be defined as a single object - it could be constructed using five or six separate planes.

(e) There must be at least one planar surface in the scene that has a pattern or a "texture" on it (e.g., a chequered floor).

OpenGL functions should be used only for rendering the ray traced image.

2. Extensions

This assignment will be marked out of 20, with 16 marks for the ray tracer and 4 marks for a report (see below). With just the above features, your ray tracer would earn at most 9 marks out of 16, and the report 2 marks out of 4. To get more than that, you need to implement additional features. Some possible features, and the approximate marks they would each gain if implemented correctly (up to a maximum of 16 marks) are as follows.

- Primitives other than a plane, sphere or box [eg., cone(1 mark), cylinder (1 mark), tetrahedron (1 mark), torus (2 marks)].
- Multiple light sources including multiple shadows generated by them: 1 mark
- Spotlight: 1 mark
- Transparent object: 1 mark (Note: A transparent object should <u>not</u> be modelled as a special case of a refractive surface with $\eta_1 = \eta_2 = 1$)
- Refraction of light through an object : 1 mark
- Anti-aliasing: 1-2 marks
- An object transformed using rotation or shear transformation: 1-2 marks

- A non-planar object textured using an image: 1 mark
- A non-planar object textured using a procedural pattern: 1-2 marks
- Fog: 1-2 marks
- Any other feature (eg: acceleration algorithms such as bounding volumes, advanced illumination models, camera motion, constructive solid geometry): 1 – 5 marks, depending on the level of complexity.

The marks associated with each feature should be taken to be indicative of the time and/or effort required to implement that a feature. The 9 marks for the minimum requirements are relatively easy to get when compared to the marks gained for some of the extension features. The above list should not be taken as a list of the only features that can be implemented.

3. Report (Max. marks: 4, No. of pages: 2 – 5)

The report should describe your ray tracer, including both its successes and its failures. It should include at least one image showing the ray tracer's capabilities. You may include additional images demonstrating various features of your implementation. The report should clearly outline both the mathematical and the implementation aspects of all extra features you have implemented in the ray tracer, other than those listed as minimum requirements. For example, if you have implemented anti-aliasing in your ray tracer, you should describe its purpose, the effect it produces on the output (include figures showing the rendering with and without anti-aliasing), and how it is implemented. Similarly, if you have used procedural textures, provide the mathematical equations used for mapping coordinates to colour values. All resources and references used in your work must be cited in the report.

If your program takes a long time (>30 secs) to generate the output, give an estimate of the time taken by the program to run on lab machines. Ray tracing can be computationally very expensive. The time taken to ray trace an image using a simple "brute force" algorithm is proportional to the number of pixels times the number of scene objects. To keep run times to a minimum, you should do most of your development with small images (approx 500x500 pixels), simple scenes, and low levels of recursion.

Use of any code segments, data or images from the Internet or other resources should be acknowledged in the report. Please include the details of the build process. You may also (optionally) include a list of references.

4. Assignment Submission (by 11:55pm Friday 31st May 2019)

Submit electronically (using *Learn*), the source code and any other supplementary files needed to run the program. Please also submit your report in PDF format. The files may all be packaged together and submitted as a single .zip file.

5. Miscellaneous

1. Check regularly on the *Learn* system forums for spec updates and clarifications.
2. You may submit up to one week late for a 15% penalty.
3. This is not a group project. Each student must individually develop the program. In particular, students are not permitted to share program source code in any way. Standard departmental regulations regarding dishonest practices apply.