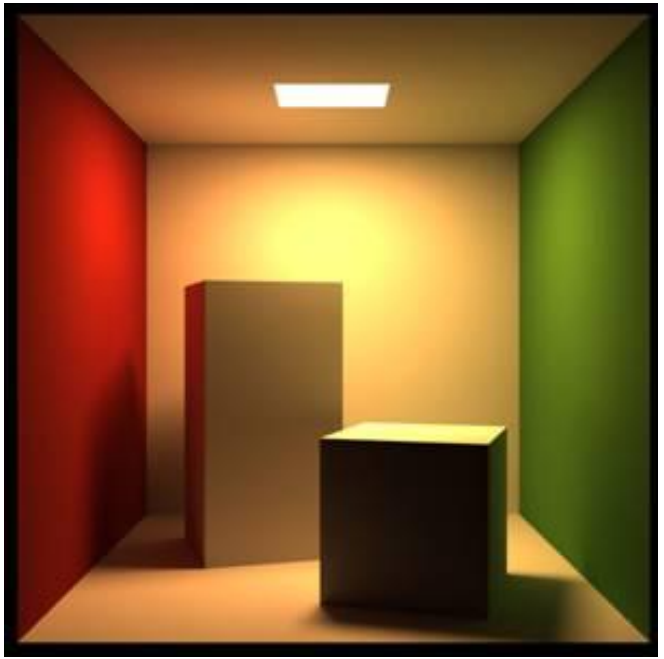
Assignment 2 - Ray Tracing

In this assignment, you will be building a ray tracer. At the end of this assignment, your ray tracer should be able to minimally handle opaque surfaces with lighting, shadows, reflections and texture mapping.



Sending out rays

- The first step is to uniformly send out rays from the camera location. You will need to use backwards ray tracing where rays are sent from the camera, one ray per pixel.
- When generating your scene, use a perspective projection "camera" with changable camera properties.

Assignment 2 – Ray Tracing

Intersection Testing

- The next step would be to write code for intersections and texture mapping. The mathematical solutions for the intersection and texture mapping code are provided in the lecture notes.
- In order to perform Phong shading, normals are required. For triangles, you can interpolate the x, y, and z coordinates of the normals at each vertex, and then normalize their length. We recommend using barycentric coordinates for interpolation of triangles. For spheres, the normal is simple to calculate from the center of the sphere.
- You are also required to implement texture mapping for triangles and spheres. For triangles, interpolate the texture coordinate using barycentric coordinates. For spheres, we recommend using spherical coordinates for texture mapping.

Assignment 2 – Ray Tracing

Illumination

- The third step in the process would be to implement the illumination equations. The illumination equation is given in the lecture notes.
- Note that in the ideal ray tracer, there should be no ambient effect. That would make the area the light cannot reach completely dark.
- For materials with a non-zero specular component, you need to recurse (upto a maximum depth of at least 3).

Your program should:

- Ray trace and correctly draw a **sphere object** in the scene
- Ray trace and correctly draw a **cube object** in the scene
- Ray trace and correctly draw a **model object** in the scene
- Correctly perform **Phong shading** by supporting ambient, diffuse, and specular lighting in your renderer. Interpolate the normals of the vertices of the polygons to determine the normals at each pixel to be used for shading.
- **Display shadows** by not illuminating surfaces by lights from which they are occluded.
- Correctly perform **recursive reflection** (up to 3 iterations) by creating a shiny surface and bounce rays off of it.
- Read in **material properties** of an object in the scene, including the image files and texture coordinates, and add textures to your raytraced triangle objects and model objects. Use barycentric coordinates to calculate the material properties and texture coordinates at each pixel. Texture mapping for model object is not required.
- Be reasonably commented and written in an understandable manner.
- Be summitted along with the **executable file** (if possible), corresponding **model object files,** JPEG files displaying your ray tracing **screenshots** and a **readme** describing the finished and unfinished work. Included screenshots must demonstrate all of the visible features of your ray tracer (shadows, textures, etc., but not bounding volumes). Put snapshot files in a sub-directory called images. This assignment does not need any report documents.
- Be summitted in .zip format, named hw2_StuID_StuName.

Assignment 2- Ray Tracing

Reference Book:

An Introduction to Ray Tracing.

Andrew S. Glassner, editor, Academic Press, 1989.

ISBN 0-12-286160-4.

Deadline: Dec 30th 23:59, together with the report of course project.

Send to email: computergraphics18@163.com