

Final Project Proposal: Ray Tracer
CS 488

Patrick Dobson
pjdobson
20384008

1.0 – Topics

- Recursive ray-tracing (reflection and refraction)
- Calculating UV coordinates for texture mapping and bump mapping
- Acceleration techniques (spatial subdivision, bounding boxes, distributed ray tracing)
- Ray tracing CSG objects
- Soft shadows

2.0 - Statement

The final scene will be a game of scrabble with a mirror and clear glass sitting on the table. In order to do the mirror, there will need to be reflections. Texture mapping will need to be used to texture the scrabble pieces, board and table. Bump mapping will be used on the board to give it the illusion that the board has slots to keep the letters in place. Refraction will be necessary to bend rays travelling through the clear glass. CSG can be used to create the cup (using cylinders and a torus to round off the edges at the top of the glass). Soft shadows will be used to bring the scene closer to photo realism.

3.0 - Technical Outline

In order to implement the new primitives an equation describing their surface must be known. For cones and cylinders these are in the course notes. The equation for the torus can easily be found online.

3.1 – Bump Mapping

The technique to be used involves taking the height field and generating a normal map from it. The formula to calculate the normal for each pixel is as follows:

$$\vec{N} = \frac{\langle H_s - H_y, H_s - H_x, 1 \rangle}{\sqrt{(H_s - H_y)^2 + (H_s - H_x)^2 + 1}}$$

Where H_s is the height of the source pixel, H_y is the height of the pixel directly above and H_x is the height of the pixel directly to the right.

3.2 – Refraction

The reflection coefficient from dielectric materials can be approximated using Schlick's approximation^{[1][2]}.

$$R(\theta) = R_0 + (1 - R_0)(1 - \cos(\theta))^5$$

$$R_0 = \left(\frac{n_1 - n_2}{n_1 + n_2} \right)^2$$

The current index of refraction must be kept track of throughout the lighting calculations

3.3 – Input language

gr.cylinder(name) – Creates a cylinder at (0,0,0) with a radius and length of 1 unit.

gr.cone(name, zmin, zmax) – Creates a cone with the tip at (0,0,0) wrapped around the z-axis, where the slope of the edges is 1. Zmin and zmax correspond to the start and stop values for creating a truncated cone.

gr.torus(name, rin, rout) – Creates a torus at (0,0,0) on the XY-plane. Rin corresponds to the radius from (0,0,0) to the circumference of the ring. Rout corresponds to the radius of the ring.

gr.csg(name, op, prim1, prim2) – Creates a new CSG object. Op specifies the boolean operator to use to combine the primitives ('u'=union, 'i'=intersection, 'd'=difference). Prim1 and prim2 correspond to the left and right operands, respectively. These must be primitives or other CSG objects.

```
gr.mesh(name, {
    {v1x, v1y, v1z},
    ...
    {vnx, vny, vnz}
}, {
    {p1l, ... p1m},
    ...
    {pn1, ... pnm},
}, {
    {u1, v1},
    ...
    {un, vn }
})
```

} – Creates a mesh (same as before). The uv coordinates are the texture coordinates for each corresponding vertice (and are optional).

gr.material(name) – Creates a material with the specified name.

material:set_diffuse(r,g,b) – Sets the diffuse color for the material.

material::set_diffuse(img) – Sets the texture for the diffuse color. Img is the location of the texture.

material::set_specular(r,g,b) – Sets the specular color for the material.

material::set_specular(img) – Sets the texture for the specular color. Img is the location of the texture.

material::set_shininess(c) – Sets the shininess of the material.

material::set_reflection(c) – Sets the reflective coefficient of the material.

material::set_refraction(c) – Sets the refraction index for the material.

Bibliography

[1] – Peter-Pike J. Sloan, Interactive Ray Tracing
<http://www.ppsloan.org/publications/rtrt99.pdf>
[refraction and reflection with dielectrics]

[2] - http://en.wikipedia.org/wiki/Schlick's_approximation
[more details about Schlick's approximation]

[3] – NVIDIA, The CG Tutorial
http://http.developer.nvidia.com/CgTutorial/cg_tutorial_chapter08.html
[bump mapping]

[4] - <http://paulbourke.net/dataformats/obj/minobj.html>
[more OBJ format information for texture mapping, vertex normals, etc.]

Objectives

Name: Patrick Dobson

UserID: pjdobson

Student ID: 20384008

- ___ 1) Ray tracer supports new primitives (in addition to boxes/spheres): Cylinder, cone, torus
- ___ 2) Ray tracer able to perform CSG operations.
- ___ 3) Texture mapping for primitives and meshes.
- ___ 4) Bump mapping for primitives and meshes using Height Fields^[3].
- ___ 5) Reflection (with dielectric materials)
- ___ 6) Refraction (with dielectric materials)
- ___ 7) Spatial subdivision using a regular grid to accelerate ray tracing
- ___ 8) Soft shadows using stratified random sampling
- ___ 9) Anti-aliasing using adaptive super-sampling
- ___ 10) Render of final scene using the ray tracer.

* The extra objective for A4 was Anti-aliasing using only super-sampling.

Declaration:

I have read the statements regarding cheating in the CS488/688 handouts. I affirm with my signature that I have worked out my own solution to this assignment, and the code I am handing in is my own.

Signature: