CS488 A5 PROJECT DOCUMENTATION

YacRay

(***Yet Another CS488 Raytracer***)

Name: Lawson Fulton
Student ID: 20381453
User ID: ljfulton

April 3, 2015

Final Project:

Purpose :

As it stands, I have implemented uniform supersampling, reflection, and refraction as a result of A4. I have included these as objectives for my project[1] along with a number of other features I think are reasonable to implement in the allotted timeframe. My goal is to secure the 10 objective marks, and then focus my attention towards implementing more advanced features such as global illumination, depth of field, etc. More research will be required before I decide what approach to use for these additional features and will not be detailed in this proposal.

Statement :

My ideal final scene will be something simple and as photorealistic as possible that demonstrates all of my implemented features. I have not decided on a specific scene as I will hopefully come up with ideas as I work on the project. One idea I have considered is a Tim Horton's paper cup filled with coffee sitting on a desk beside a pair of glasses. The cup would demonstrate texture mapping, the table glossy reflections, the glasses refraction, etc.

I feel that a very solid grasp of the foundational techniques will allow advanced algorithms to be more accessible. Accomplishing all of the objectives in a way that consistently produces beautiful and artifact free images will be a challenge but also a great learning experience.

Through the completion of this project I will learn a great deal about the foundations of modern rendering techniques, and the kinds of challenges involved in attempting to create realistic images.

Technical Outline :

Objective one: Reflection

Implemented by calculating reflected ray using surface normal at point of intersection. Contribution from reflected ray is then recursively calculated using the original traceRay() function.

Objective two: Transparent materials with refraction

Implemented in a manner similar to reflection. Refracted ray is generated by using surface normal and Snell's law, along with the materials respective indexes of refraction. Contribution of reflection versus refraction will be calculated using the Fresnel equations.

However, there are some difficulties that you have to look out for. When calculating your ray intersections, you need to make sure to handle the case where the origin of the ray is inside the object being tested against. Also, the surface normal must be reversed if this is the case. Lastly, special care must be taken to keep track of index of refraction for the medium in which the ray is coming from, and that which it is being transmitted to.

Both reflection and refraction are controlled through a new command in the input language:

```
gr.fancy_material({dr, dg, db}, {sr, sg, sb}, shininess, reflectivity, ior, transparency,
glossiness)
```

Objective three: Soft Shadows

Implemented by sending many rays when calculating the contribution from a given light source. Rather than using a single ray from the intersection point, to the point light source, send multiple rays from the intersection point to a random distribution of points on the light source, then average the resulting values.

[1] I am taking uniform supersampling as my additional feature from A4 that will not count as an objective. However, Prof. Baranoski has allowed me to use *adaptive* supersampling as an objective.

A new light source command will be added to the input language that will be able to specify a spherical light source of a desired radius. `gr.ball_light(location, colour, falloff, radius)`

Will use [1] as a source.

Objective four: Glossy Reflections

Glossy reflections are implemented in a very similar way to soft shadows. When calculating the reflection contribution, trace many reflected rays according to some distribution and average the colour of the results. Glossiness or the amount of blurriness of the reflection will be controlled by the glossiness parameter in the new material function mentioned in objective 2.

Will also use [1] as a source.

Objective five: Anti-aliasing using adaptive supersampling

Adaptive supersampling works by dividing up each pixel into 4 regions and tracing a ray through the centre of each sub pixel. The 4 resulting colours are compared, if they differ significantly, then the pixel is further subdivided and more rays are generated. This process can proceed recursively until a maximum depth or number of rays have been traced. Finally all of the resulting colour values are averaged determining the pixel colour.

Will use [2] as a a source.

Objective six: Smooth Phong shading on triangle meshes

I will implement a .obj loader in C++ that supports specifying vertex normals for triangle meshes. Then when rendering that mesh, it is a case of interpolating the normals from the neighbouring vertices when an intersection point is found on a triangle face. I will also add a new command to the input language for my new mesh primitive type.

`gr.obj_mesh(filename)`

Objective seven: Texture Mapping

Texture mapping is done by taking an intersection point on the surface of an object and mapping it to a value $(u, v) \in [0, 1]$. This new 2d coordinate is then used to sample from a texture file using bilinear interpolation. I will also be implementing texture mapping for my new mesh type, so the obj loader will have to be capable of reading texture map coordinates from the file. The texture mapped coordinates are stored per vertex, and like normals, are interpolated to find the (u, v) value for a given intersection point.

Will use [3] as a source.

Objective eight: Bump Mapping

Bump mapping is very similar to texture mapping because we are calculating surface normals from (u, v) interpolation rather than surface colours. The difference however is that the bump map does not specify the normals explicitly, but only the surface displacement. This displacement is then used to construct a differentiable function representing the surface of the bump mapped object. At the point of intersection the derivative of this function is calculated numerically to construct the normal.

Will use [3] as a source.

Objective nine: Tone Mapping

When a raytraced image is finished, the colour intensities for each pixel must be mapped to the range $[0, 1]$ in order to be displayed. The default way of handling this is to simply clip any intensities > 1, however this can result in unsightly artifacts in images with high dynamic range. A solution is to use a more complex intensity mapping function that tries to preserve both bright and dark aspects of an image.

I will be implementing Reinhard's tone mapping operator detailed in [4].

Objective ten: Final Scene

I will be creating a simple but realistic scene as described in my purpose.

Bibliography :

1. Robert L. Cook, Thomas Porter, and Loren Carpenter. 1984. Distributed ray tracing. SIGGRAPH Comput. Graph. 18, 3 (January 1984), 137-145. DOI=10.1145/964965.808590 http://doi.acm.org/10.1145/964965.8085

2. Turner Whitted. 1980. An improved illumination model for shaded display. Commun. ACM 23, 6 (June 1980), 343-349. DOI=10.1145/358876.358882 http://doi.acm.org/10.1145/358876.358882

3. James F. Blinn. 1978. Simulation of wrinkled surfaces. In Proceedings of the 5th annual conference on Computer graphics and interactive techniques (SIGGRAPH '78). ACM, New York, NY, USA, 286-292. DOI=10.1145/800248.507101 http://doi.acm.org/10.1145/800248.507101

4. Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. 2002. Photographic tone reproduction for digital images. In Proceedings of the 29th annual conference on Computer graphics and interactive techniques (SIGGRAPH '02). ACM, New York, NY, USA, 267-276. DOI=10.1145/566570.566575 http://doi.acm.org/10.1145/566570.566575

Objectives:

Full UserID:____________________________ **Student ID:**____________________________

___ 1: Objective one: Reflective materials

___ 2: Objective two: Transparent materials with refraction

___ 3: Objective three: Soft Shadows

___ 4: Objective four: Glossy Reflections

___ 5: Objective five: Anti-aliasing using adaptive supersampling

___ 6: Objective six: Smooth Phong shading on triangle meshes

___ 7: Objective seven: Texture Mapping

___ 8: Objective eight: Bump Mapping

___ 9: Objective nine: Tone Mapping

___ 10: Objective ten: Final scene that demonstrates objectives 1 - 9

A4 extra objective: Anti-aliasing using uniform grid-based supersampling