Brian Curless, 318 Sieg
Head TA: Craig S. Kaplan

Homework 4: Raytracing, Texture Mapping, and Curves

Received: Wednesday, 18 November 1998
Due: Wednesday, 2 December 1998

DIRECTIONS

Please provide short written answers to the questions in the space provided. If you require extra space, you may staple additional pages to the back of your assignment. Feel free to talk over the problems with classmates, but please answer the questions on your own.

Name: ____

Problem 1 - Raytracing

(a) Our method of raytracing involves tracing all rays from the eye. This is also known as "backward ray tracing". What are the major problems associated with this method?

(b) If instead we decide to trace all rays from each light source, would we solve all of the above problems? What new problems arise by using this method of "forward ray tracing"?

(c) In class, we described how we could perform pixel anti-aliasing with jittered supersampling. Why is jittering better than casting rays with a purely random distribution? Give an example to illustrate your point.

(d) Distributed raytracing allows us to achieve more realistic results than simpler methods. For the following uses of distributed raytracing, briefly describe how they are implemented. Indicate when only a few samples per pixel will suffice and also indicate when many samples would instead be necessary to model the effect accurately (i.e. to avoid aliasing in other dimensions). List extreme cases where applicable. The first one is done as an example.

Pixel Anti-aliasing:

Randomly distribute a number of rays over the sub-areas of the pixel (jittering). Average the intensity contributions from these rays to determine the overall pixel intensity. Few samples would suffice for an area of uniform color. Many samples would be needed for a hard edge.

Motion blur:

Depth of Field:

Glossy Reflections:

(e) You just bought a commercial rendering package, but you didn't buy the source code or documentation of the underlying algorithms.

I. As your final rendering, the system can raytrace your scene. You've been told that the termination criteria for tracing rays is either

(1) to always trace rays to a recursive depth of 4 or
(2) to adaptively terminate based on ray contribution.

Describe a scene that can be used to discern which method is used. Be specific about the details of the scene and what you would be looking for.

II. As a preliminary step, scenes are made entirely of polygons and are rendered using either:

(1) Phong interpolated shading or
(2) Gouraud interpolated shading.

Describe a scene in which the difference between these two shading schemes would be noticeable and explain how you would identify which system is being used.

(f) Why does Gouraud interpolation yield Mach banding? Give an example. Will Phong interpolation solve this problem for diffuse surfaces?

Problem 2 - Texture anti-aliasing

(a) For each of the following types of texture maps, list the amount of time (in Big O notation) that it takes to return a value for a k by k square of texels that is getting mapped to a pixel p in screen space. For each type of texture map, also list the amount of storage that the preprocessed texture map takes up in big O notation. You may assume that the original image that represents the texture map is r pixels wide and r pixels high. Justify your answers.

i. Brute Force

ii. Summed Area Tables

iii. Mip Maps

(b) Now assume that you want to compute an exact area average over a texel-aligned rectangle of dimensions m by k. How does the worst case complexity change for each of the above cases? Justify your answers.

i. Brute Force

ii. Summed Area Tables

iii. Mip Maps

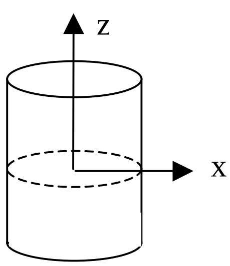
Problem 3 – Procedural Solid Textures

(a) One problem with the standard method of texture mapping is that textures appear to be stuck on objects in the same manner that wallpaper is put on a wall. In many cases, a surface doesn't look like it is really made up of the material that is texture mapped onto it, for example, consider a wood texture map. One method that is used to alleviate the "wood veneer" problem is to use a solid, or procedural, texture map.

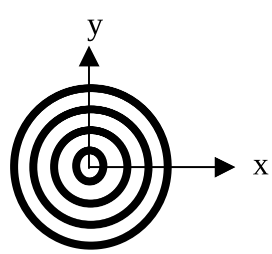
A procedural texture map answers the question: What is the color of the surface at this point?

Fill out the following function with very general pseudocode. Your code will return the answer to the question: What is the color of the wood at this point on the surface? Assume that you have values for the age of the wood, and the two colors for each type of ring in: age, darkRing, and lightRing. The Point passed into the function contains the x, y, and z coordinates of the point on the surface in object space.

Here is an example of a horizontal slice of a cylinder with our wood shader applied to it:



Original Cylinder



Horizontal slice

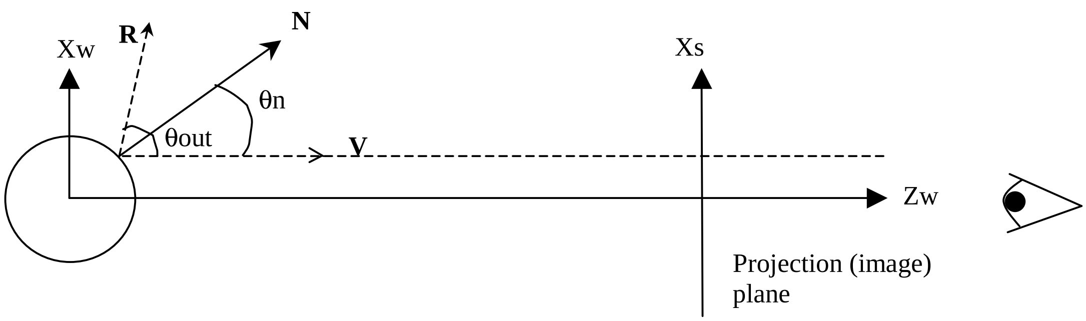
```
color wood (Point pt) {

}
```

(b) As the illustration above shows, this pattern is somewhat boring. It looks too perfect. How would you modify the code to introduce a less 'sterile' appearance, specifically, is there a way to introduce some 'wiggles' into the rings?

(c) Let's say that during the seasons that yield light rings, the tree was absorbing silver from some strange atmospheric conditions. How would you modify the above function so that light colored rings were shinier than dark colored rings?

Problem 4 – Environment Mapping

One method of environment mapping (reflection mapping) involves using a 'gazing ball' to capture an image of the surroundings. The idea is to place a chrome sphere in an actual environment, take a photograph of the sphere, and warp the resulting image into an environment map. Let's examine this in two dimensions. Below is a diagram of our setup. Assume that the viewer is infinitely far away so that rays from the viewer are parallel. All angles are measured in <u>world</u> coordinates with respect to the Z-axis.



(a) For a ray of given Xs, determine the point of intersection (in world coordinates) with the two dimensional gazing ball. Assume that the gazing ball is a unit circle centered at the origin in world coordinates.

(b) Calculate the normal vector at the point of intersection as a function of Xs.

(c) Calculate θn (the angle between the normal and the Z-axis) as a function of Xs.

(d) Calculate θout (the angle between the reflected ray **R** and the Z-axis) as a function of Xs.

(e) Plot θout versus Xs. In what regions do small changes in image coordinates map to large changes in reflected angles?

(f) If we were to use θout to index into an environment map, what kind of errors can we expect when raytracing?

Problem 5 - Curves

(a) True/False. Label each statement as “true” or “false”. Please justify your answers.

______ A Bezier curve of order 2 lies in a plane.

______ A Bezier curve of order 1 is a point.

______ A Catmull-Rom spline is a piecewise parametric polynomial curve.

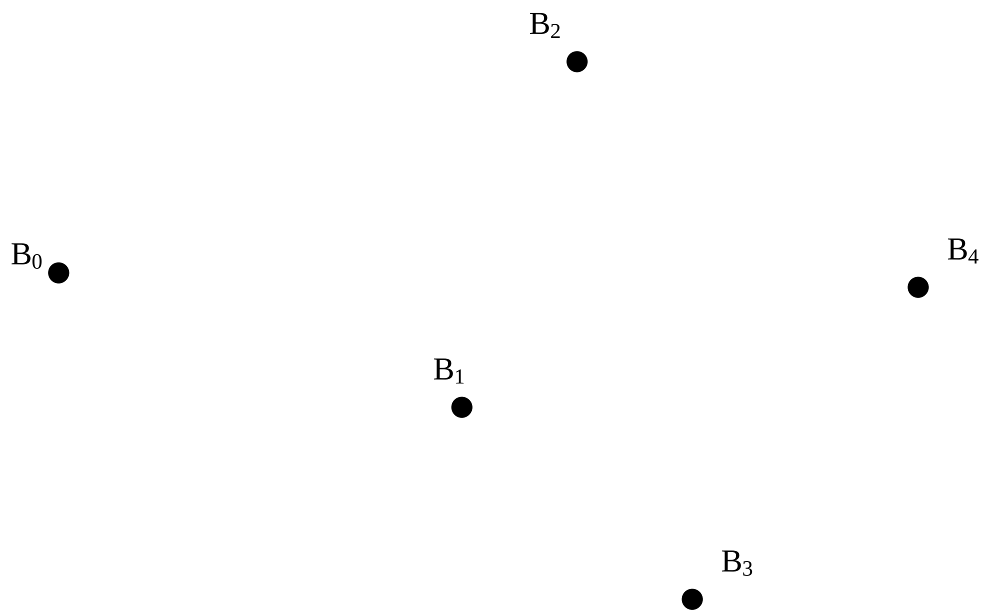
______ Catmull-Rom splines are C2 continuous.

______ A Catmull-Rom spline lies within the convex hull of its control points.

(b) Describe how Catmull-Rom splines are different from B-splines in terms of standard characteristics (continuity, what points are interpolated, control...) What application can you think of where one type of spline would be clearly preferential to the other?

(c) When displaying Bezier curves, there is a special case (i.e. a configuration of control points) where our method of recursive subdivision does unnecessary work. What is this case, and how can we remedy it?

(d) The points {B0, B1, B2, B3, B4 } shown below are control points for a cubic B-spline. Construct the Bezier control points {V0, V1, V2, V3 } that are derived from the B-spline on the diagram below as carefully as you can. After you have found the Bezier control points, carefully construct the resulting spline (be sure to construct a few points to help your approximation).



(e) The points {C0, C1, C2, C3, C4 } shown below are control points for a cubic Catmull-Rom spline. Construct the Bezier control points {V0, V1, V2, V3 } that are derived from the Catmull-Rom spline on the diagram below as carefully as you can. After you have found the Bezier control points, carefully construct the resulting spline (be sure to construct a few points to help your approximation).