Solutions

CMPS 160 F06
Introduction to Computer Graphics
November 14, 2006
Midterm Exam #2

You have the entire class period to complete this exam.

All pages are worth an equal amount.

Partial credit will be given for clear evidence of correct reasoning even if the final solution is incomplete.

No books
No notes
No calculators
No cooperation
No smiling

1. The velociraptor spots you 40 meters away and attacks, accelerating at 4 m/s² up to its top speed of 25 m/s. When it spots you, you begin to flee, quickly reaching your top speed of 5 m/s. How far can you get before you're caught and devoured?

2. You are at the center of a 20cm equilateral triangle with a raptor at each corner. The top raptor has a wounded leg and is limited to a top speed of 10 m/s.

(Not to scale)

The raptors will run toward you. At what angle should you run to maximize the time you stay alive?

3. Raptors can open doors, but they are slowed by them. Using the floor plan on the next page, plot a route through the building, assuming raptors take 5 minutes to open the first door and halve the time for each subsequent door. Remember, raptors run at 10 m/s and they do not know fear.
1. Color Spaces

a) Sketch the HSV and RGB color spaces with labels for each of the axes and indicate which location each of the colors black, white, gray, red, cyan, and yellow occupy in each color space.

b) In the HSV color model, starting with a pure red color, describe the effect that adding more white, black, etc... has on H, S, and V. Define 'adding more' as moving in the HSV cone in the direction of the new color. Use the terms constant, increase, or decrease to describe the change.

<table>
<thead>
<tr>
<th></th>
<th>Hue</th>
<th>Saturation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add white</td>
<td>constant</td>
<td>decrease</td>
<td>constant/increase</td>
</tr>
<tr>
<td>Add black</td>
<td>constant</td>
<td>constant/</td>
<td>decrease</td>
</tr>
<tr>
<td></td>
<td></td>
<td>increase</td>
<td>constant</td>
</tr>
<tr>
<td>Add gray</td>
<td>constant</td>
<td>decrease</td>
<td>decrease/constant</td>
</tr>
<tr>
<td>Add yellow</td>
<td>decrease/increase</td>
<td>constant</td>
<td>constant</td>
</tr>
<tr>
<td>Add cyan</td>
<td>constant</td>
<td>decrease</td>
<td>constant</td>
</tr>
</tbody>
</table>
2. Lighting models

a) You have a material with a strange BRDF. It totally absorbs red light, perfectly reflects green, and evenly diffuses blue light. Sketch a separate goniometric diagrams describing this BRDF for each of the red, blue, and green channels.

```
  "no reflection"   "mirror reflection"   "diffuse reflection"
   
   Red               Green              Blue
```

b) Describe in words what you would see if you shined a white light on a smooth sphere made from this material?

```
The sphere would look blue with the world (any lights) reflected in green, producing blue + green = cyan regions.
```

c) Using the OpenGL lighting model, specify the RGB color for each of $k_{ambient}$, $k_{diffuse}$, and $k_{specular}$, as well as the single value of the specular_exponent that best approximates the BRDF above?

```
k_{a} = (0, 0, 1)
k_{d} = (0, 0, 1)
k_{s} = (0, 0, 1)
```

$n = \infty$ or as high as you can get

d) In general, describe the difference between Gouraud and Phong shading.

In Gouraud shading the lighting equation is only evaluated at each vertex, however in Phong shading the equation is evaluated at each pixel using interpolated normals.
3. BSP Trees

a) Is BSP tree creation view dependent? Separately, is tree traversal view dependent?
   
   Creation is not view dependent.
   Traversal is view dependent.

Below is a scene with three labeled polygons and two possible camera positions.

b) Draw two possible BSP trees, making one balanced, and one unbalanced.

c) For each tree, what is the traversal order for painters algorithm from camera C1?
   
   C, B, A

   1. A, C, B
   2. A, C, B
   3. C, B, A
   4. C, A, B

   1. A, C, B
   2. A, B, C
   3. C, A, B
   4. C, A, B

d) For each tree, what is the traversal order for painters algorithm from camera C2?
4. Z-buffer

Consider an orthographic scene with objects listed below being rendered at 5 by 5 pixels, with the view plane at z=0 (looking along the positive z axis). Show the state of the z-buffer (with numbers) once these squares have been rendered (without using anti-aliasing or sub-sampling).

There is a red square with corners at
(0, 0, 1),
(1, 0, 1),
(1, 1, 1),
(0, 1, 1).

There is a green square with corners at
(2, 2, 2),
(4, 2, 2),
(4, 4, 2),
(2, 4, 2),

and a blue square with corners at
(0, 0, 3),
(4, 0, 3),
(4, 4, 3),
(0, 4, 3).

\[
\begin{array}{cccccc}
4 & 5 & 5 & 5 & 5 & 5 \\
3 & 3 & 2 & 2 & 2 & 2 \\
3 & 3 & 2 & 2 & 2 & 2 \\
1 & 3 & 3 & 3 & 3 & 3 \\
0 & 3 & 3 & 3 & 3 & 3 \\
0 & 1 & 2 & 3 & 4 &
\end{array}
\]
5. Raytracing

You are designing a very simple ray tracer with shadows and reflections (no refractions). It should render scenes with a single color channel (grayscale images). The critical part of this program is a function called trace that takes a ray and recursion depth as arguments and returns a pixel intensity. The trace function should call itself recursively where appropriate to accomplish the final task. In a top-level loop in the bigger program, trace would be called for a ray emanating from each pixel in the image with a recursion depth value of 0. Write a pseudo-code definition for trace(ray,depth).

Follow a consistent syntax and use any of the definitions provided below. If your code needs to make use of any other functions or global values, give a one line description of what you are accessing.

Helpful assumptions:
- All lights have unit intensity (1) and do not fall off with distance.
- Calling trace with a ray that does not intersect any object should result in a background color.
- Calling trace with a depth greater than the maximum should result in a background color.
- Every point on every object has half diffuse reflectance and half perfect-mirror reflectance.
- No object emits light on its own.
- 0 represents Black

Global values:
- MAX_DEPTH – maximum recursion depth
- OBJECTS – collection of object in the scene
- LIGHTS – list of lights in scene
- BACKGROUND_INTENSITY – some background color (single channel)

Utility functions:
- SomeObject | None = closest_object( Ray, OBJECTS)
- True | False = can_see( Ray, Light )
- OutgoingRay = reflect( IncomingRay, Object)
- Intensity = diffuse(Object, Ray, Light)

Write your response on the next page.
trace (Ray, Depth) {
    if (Depth > MAX_DEPTH) {
        return BACKGROUND_INTENSITY
    }
    find_intersection (1 pt)
    Object = closest_object (Ray, OBJECTS)
    if (Object == None) { handle no intersection (1 pt)
        return BACKGROUND_INTENSITY
    }
    Intensity = 0
    loop for all lights (1 pt)
    for (Light in LIGHTS) {
        if (can-see (Ray from object intersection, Light)) {
            Intensity += 1
            per-light logic (1 pt)
        }
    }
    Reflected = reflect (Ray, Object) trace reflected ray (2 pts)
    return \( \frac{1}{2} \cdot \text{Intensity} + \left( \frac{1}{2} \cdot \text{trace(Reflected, Depth + 1)} \right) \)
    return intensity (2 pts) combine intensities (1 pt)
}
6. Back Face Detection

The OpenGL specification defines that (by default) the winding for front-facing polygons should be counter-clockwise – following intuition matching use of the right hand rule. For the solid cube below, specify an ordering of vertices for each of the eight faces that would allow it to be rendered properly with back face culling enabled.

![Cube Diagram]

<table>
<thead>
<tr>
<th>A</th>
<th>C</th>
<th>F</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>B</td>
<td>F</td>
<td>C</td>
</tr>
<tr>
<td>E</td>
<td>G</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>H</td>
<td>E</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>D</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>E</td>
<td>H</td>
<td>B</td>
</tr>
</tbody>
</table>

Rotations of these are also valid:
- GCAE
- BFCG
- DFSA, HDBF
- HBGD

Oops! Only six faces in a cube!
7. Accumulation Buffer Algorithms

In class you saw how the accumulation buffer (or equivalent screen-sized color buffer storage like a texture) could be used to affect anti-aliasing. This involved rendering the same scene several times with different camera position offsets for each pass and using the accumulated result as the final image.

In general, accumulation buffer algorithms involve rendering several versions of the same scene with some parameter changed and averaging the resulting pixel colors to produce final image. Following this general template, describe how you might affect one of depth-of-field, motion blur, or soft shadowing (assuming rendering with sharp shadows was part of normal operations).

- **Antialiasing**
- **Depth of Field**
- **Motion Blur**
- **Soft Shadows**

Similar idea but with camera positions jittered for each sample

Similar idea but with each moving object jittered along its direction of motion a bit
8. Spectral and Perceptual Colors

a) If a filter (some colored, semitransparent plastic) has a broad spectral transmittance curve with a hump centered around the blue region. Which type of electromagnetic radiation will the filter let passed more, infrared or ultraviolet?

b) If two LEDs (which emit pure spectral colors), one red and one green shine on a white wall. The wall naturally appears yellow in your perception. In your hands are those yellow sunglasses you designed in homework question earlier this quarter. When you put the sunglasses on the scene becomes very dark, if not black. What might be going on here to keep you from seeing the scene like you without the glasses? Draw any curves necessary to support your explanation.

c) Black and white film in cameras is said to be panchromatic because it is sensitive to a wide range of visible light. However many types of black and white film are particularly sensitive to red-like colors. In old horror films chocolate syrup was often substituted for red prop-blood because the red-appearing liquid did not right in the black and white output (it looked too bright). What kind of colored filter would you place in front of the camera lens to make prop-blood (or even real blood for that matter) look correct in the black and white film (and avoid attracting ants with chocolate everywhere)? Chocolate, in these situations, was usually preferred because adding any kind of filter in front of the camera would reduce the amount of light collected in an already dark scene.