CMPS 160
Intro to Graphics
Fall 2005

Final Exam
1. You are rendering a scene for the highway department (since they are the government, you have scored a $1 million dollar contract to make one picture). They insist that your renderings include the shiny little reflective markers along the side of the road. It turns out these are made from retroreflective material, meaning that light is reflected mostly back in the direction of the light source.

a) Sketch the goniometric diagram for a retroreflective surface.

b) You remember learning that the OpenGL lighting model is $k_a + k_d (N \cdot L) + k_s (N \cdot H)^n$. How would you change this equation to model retroreflection?

The specular lobe is now retroreflective

so we want the peak to occur when the view direction is near the light direction

$\ldots + k_s (V \cdot L)^n$
1. You are rendering a scene for the highway department (since they are the government, you have scored a $1 million dollar contract to make one picture). They insist that your renderings include the shiny little reflective markers along the side of the road. It turns out these are made from retroreflective material, meaning that light is reflected mostly back in the direction of the light source.

a) Sketch the goniometric diagram for a retroreflective surface.

![Goniometric Diagram](image)

**Figure 1 Reflection:** Diffuse reflection bounces light in multiple directions after striking a surface, while specular reflection bounces light off the surface at the same angle it struck the surface at.

**Figure 2 Pavement Marking Retroreflectivity:** When light enters the retroreflective bead in the pavement marking, the light rays are retroreflected back to the drivers' eyes, allowing them to more easily see the critical visual cues on the road.
2. You’re working on compositing for a major movie about killer attack butterflies. You need to composite the butterfly over the flower over the background. You have the butterfly and flower specified as RGB plus an alpha mask.

\[ \text{Butterfly}=[B_R,B_G,B_B,B_a] \quad \text{and} \quad \text{Flower}=[F_R,F_G,F_B,F_a] \]

Unfortunately the background painter isn’t done yet, so you first want to composite the butterfly over the flower and save this in a new image together with a new alpha mask.

---

a) What are the equations to determine \( \text{New}=[N_R,N_G,N_B,N_a] \) if alpha is not premultiplied?

[I’m guessing you don’t have the formula memorized, I want you to derive the equation based on your understanding of compositing. – James]

\[
\begin{align*}
\text{Alpha is a percentage cover of a pixel, so we can derive geometrically, }\ B \text{ is over } \ F & \text{ so it covers part of } \ F \text{ in a pixel.} \\
N_{RGB} &= B_x \cdot B_{RGB} + (1 - B_x) F_x F_{RGB} \\
N_a &= B_x + (1 - B_x) F_x
\end{align*}
\]

b) What are the equations to determine \( \text{New}=[N_R,N_G,N_B,N_a] \) if alpha is premultiplied?

Premultiplied means that \( x \) is already multiplied into the RGB color channels, so that \( B_{pre} = B_x \cdot B_{RGB} \).

Substitution in above gives...

\[
\begin{align*}
\text{New}_{RGB} &= B_{RGB} + (1 - B_x) F_{RGB} \\
N_a &= B_x + (1 - B_x) F_x
\end{align*}
\]

c) Why is the latter preferable?

Note that premultiplied has identical formulas for \( B_{RGB} \) and \( x \). This allows much more efficient implementation.

d) Why is premultiplied undesirable?

This format quantizes the color channel. If we drop \( x \) to 0 and then raise it back up we don’t get the original back. Where \( x=0 \), any color will have \( RGB=(0,0,0) \).
3. An ellipsoid has 2 radii equal to 1 and one radius equal to 2. It can be viewed as a unit sphere which has been stretched to twice its usual width and stretches from \( \mathbf{0} = (0,0,0) \) to \((4,0,0)\). The ellipsoid has circular cross-sections when intersected with planes parallel to the yz-plane.

The GLUT command `glutSolidSphere(r, 10, 10)` can be used to generate a sphere of radius r. Using this command, plus other OpenGL commands, such as `glMatrixMode`, `glLoadIdentity`, `glTranslatef`, `glScalef`, write a code fragment that will generate the ellipsoid.

4. [Question 4 was deleted after practice testers rejected it, thus we have a guest entrance of problem #4 from last spring's final exam (with numbers changed of course). ]

4. Suppose we set up a camera using `gluLookAt(eye=[1,1,3], center=[1,-1,3], up=[1,0,0])`, and then render a scene with three spheres of radius 1, centered at \([4,-1,4]\), \([5,-3,-2]\), and \([7,-4,1]\).

a. Is it reasonable use a near plane with distance zero and a far plane of “really far”? Why or why not?

   No, quantization error and limited bit resolution in the z-buffer will cause errors if we choose really large bounds.

b. What are the best (or “tightest”) distances for the near and far planes?

   The eye is looking in the -y direction.
   
   The eye is at \( y = 1 \), the closest sphere is at \( y = 0 \), and the furthest at \( y = -5 \)
   
   \[
   \text{near} = 1 \quad \text{far} = 6
   \]
5. You are designing a graphics system that only support texture maps that are 256x256. Obviously proper texture filtering will be required in some cases. As a filter, you select box filtering over the area covered by a pixel.

a) As a particular example of texture filtering, lets consider the pixel marked with an X in the diagram above. If \( T(256,256) \) references the texel color at UV coordinates \((1,1)\) in the texture image, write an expression (math involving multiple texel values) for the color which should be copied into the frame buffer at pixel X.

\[
\text{color}(x) = \sum_{u=64}^{128} \sum_{v=128}^{256} T(u,v)
\]

b) Next your boss tells you that you must support summed-area-tables. Write an expression for the color at X assuming your texture is stored in this new format.

Recall that each texel now stores the sum of everything in the box bounded by corners at the texel and the origin. Thus...

\[
\text{color}(x) = T(128,256) + T(64,128) - T(64,256) - T(128,128)
\]

only 4 texture lookups!
6. The diagram below shows a BSP tree. The numbered darker lines are polygons, and the arrows show the orientation of the polygon.

a) Suppose we look at this scene from the right side, instead of the left. Use the BSP tree to determine the order in which triangles should be rendered.

1, 2, 5a, 3, 5b, 4

back to front from the new eye point...
(back and front labels in tree irrelevant)

b) Construct a valid BSP tree starting at triangle 1, instead of triangle 3. (Starting from the left as shown.)
7. Match up the following 4x4 matrices with the corresponding type of 3D transformations. Answer the question by drawing lines connecting the phrase on the left to the matrix on the right.

- Identity (no effect)
  - \[
  \begin{bmatrix}
  \sqrt{2} & -\sqrt{2} & 0 & 0 \\
  \sqrt{2} & \sqrt{2} & 0 & 0 \\
  0 & 0 & 2 & 0 \\
  0 & 0 & 0 & 2
  \end{bmatrix}
  \] (note that \(Z\) doesn't change)

- Uniform scale
  - \[
  \begin{bmatrix}
  0 & -1 & 0 & 0 \\
  0 & 0 & -1 & 0 \\
  1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1
  \end{bmatrix}
  \] Reflect

- Non-uniform scale
  - \[
  \begin{bmatrix}
  0 & -1 & 0 & 0 \\
  0 & 0 & -1 & 0 \\
  -1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1
  \end{bmatrix}
  \] Rot

- Reflection
  - \[
  \begin{bmatrix}
  3 & 0 & 0 & 0 \\
  0 & 3 & 0 & 0 \\
  0 & 0 & 3 & 0 \\
  0 & 0 & 0 & 3
  \end{bmatrix}
  \] Identity

- Rotation about z
  - \[
  \begin{bmatrix}
  3 & 0 & 0 & 0 \\
  0 & 3 & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 3
  \end{bmatrix}
  \] non-uniform scale

- Rotation
  - \[
  \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 3
  \end{bmatrix}
  \] uniform scale

Don't forget the homogenous \((sp^2)\) divide. Thus:

\[
\begin{bmatrix}
1 \\
1
\end{bmatrix} = \begin{bmatrix}
3 \\
3 \\
3 \\
3
\end{bmatrix}
\]
8. Adam showed a non-photo-realistic rendering technique in class that created black and white pen and ink illustrations as shown below using texture mapping and a carefully constructed set of mipmaps (also shown below).

a) Suppose that only the high resolution texture images had been specified (bottom row), and OpenGL had been used to generate the other portions of each mipmap. What would have gone wrong?

OpenGL generated mipmaps are designed for filtering, so "low pass filter" the textures. The high frequency lines would get turned into a blurry grey section without nice black and white pen strokes.

b) Suppose that we just turned off mipmapping altogether, and just use the high-resolution images. What would happen?

The high frequency pen strokes would probably have lots of aliasing, like looking like flickering when the object rotates.
9. Convert the following RGB values to HSV. (Blue in HSV is at rotation 240 degrees, and all non-degree quantities are normalized to the range 0..1)

\[
\text{RGB}(0,0,0) = \text{HSV } (0,0,0) \quad \text{Black}
\]

\[
\text{RGB}(1,1,0) = \text{HSV } (60,1,1) \quad \text{Yellow (opposite of blue)}
\]

\[
\text{RGB}(0,0,0.5) = \text{HSV } (240,1,0.5) \quad \text{Blue (half-brightness)}
\]

\[
\text{RGB}(0.5,0.5,1) = \text{HSV } (240,0.5,1) \quad \text{Blue (half-saturation)}
\]

10. This image of “Lenna” is one of the most widely used standard test images for image processing algorithms. Image processing is often concerned with sampling, Fourier transforms and other signal processing issues. Using your knowledge of signal processing and surrounding issues, speculate as to why this image might make a good standard test image. *(By the way, as trivia, this image was originally scanned from the centerfold of Nov 1972 Playboy. Check Wikipedia for the full story)*

The image contains both high frequency detail in the hair and feathers, and low frequency smooth regions on the skin and blurry background, thus it tests algorithms across a range of frequency characteristics.