CS 465 Prelim 1

Tuesday 5 October 2004

Problem 1: Image formats (18 pts)

A ray tracer computes and stores its image using this data structure to hold each pixel in the final image:

```
public class RGBColor {
  double r;
  double g;
  double b;
}
```

The image needs to be stored to an output file format that uses 8-bit RGB, gamma corrected for $\gamma = 2.2$.

1. What is the ratio between the image size in memory and the size on disk?

2. If we omit gamma correction, will the image look darker than, lighter than, or the same as the correct image?

3. Give the expression we would use to convert one of the color channels of an individual pixel before writing to disk. Assume the functions `round`, `pow`, `min`, and `max` are available. Don't worry about Java type conversion rules—just compute the integer value that needs to be stored in the file.

Problem 2: Ray Tracing Bugs (24 pts)

For each of the following bugs, give a simple test case that would catch the bug. Also describe how the image will be different, both visually and numerically, from the correct image.

1. Your implementation of the Phong shading model computes the reflection vector as

$$\mathbf{v}_R = \mathbf{v}_L + 2((\mathbf{n} \cdot \mathbf{v}_L)\mathbf{n} - \mathbf{v}_L)$$

   where $\mathbf{v}_L$ is the unit vector pointing toward the light and $\mathbf{n}$ is the surface normal. But you accidentally left out the 2.

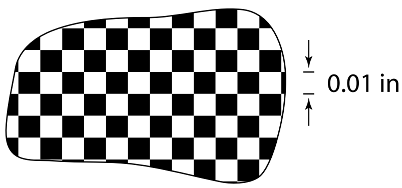
2. You and your partner somehow never discussed whether ray directions should be normalized. You wrote the Phong shader (which is now fixed to properly use the formula above) assuming the eye direction is always normalized. But your partner wrote the camera class, which computes the ray direction as the difference of the image plane point and the viewpoint and does not normalize it. Assume you're testing with a camera that has an image plane distance of 1.

3. You fix the previous bug by having the Phong shader normalize the eye vector, but you soon run into a related bug. You wrote the sphere intersection routine so that it normalizes the ray direction, then computes the correct t for the normalized ray. Your partner wrote the triangle intersection routine, and it computes t correctly but does not normalize the ray direction.

4. You fix the previous bug by having the camera normalize ray directions. But now you have a problem with your shadows. You can't remember who wrote the shadow ray generation code, but it does not normalize the direction of the shadow ray, leaving it as the difference between the light source position and the shading point. It tests for a shadow by looking for intersections in the interval $[\epsilon, 1]$.

Problem 3: Moiré patterns (25 pts)

A photographer working on a job for a housewares catalog is having trouble with Moiré patterns in a particular tablecloth. The fabric is woven of black and white threads to form a fine checkerboard pattern with 100 squares per inch. A small area of the cloth would look like this:



The photographer is using a grayscale digital camera. For the most problematic shot, the tablecloth is perpendicular to the view direction and the camera's pixel grid projects onto the cloth with a spacing of 49 pixels per inch. This tends to produce a large scale checkerboard artifact in the image.

1. If the camera performed ideal point sampling, what color would the checkers be and how big would they be (in image pixels)?

2. Fortunately the camera does not do point sampling; in fact, each pixel collects light from a square area, and the pixel areas exactly tile the image plane. How does this change the checkerboard artifact from the previous question? What gray levels appear in the image (qualitatively, not quantitatively)?

Feel free to assume the cloth is perfectly rectilinear and is aligned exactly parallel to the pixel grid. You can also assume the black and white areas of the cloth map exactly to full black and full white in the image.

Problem 4: Compositing (12 pts)

Prove that compositing with premultiplied alpha is an associative operation. That is, for any three images A, B, and C, prove that

$$(A\,\mathbf{over}\,B)\,\mathbf{over}\,C = A\,\mathbf{over}\,(B\,\mathbf{over}\,C).$$

Problem 5: Transformations (21 pts)

Classify each of the following 2D homogeneous matrices as follows: (a) rotation, (b) mirror reflection, (c) uniform scale, (d) nonuniform scale, (e) translation, (f) shear, or (g) combination.

Earlier categories take precedence over later categories if more than one applies. There is not a one-to-one mapping between the matrices and the categories.

1. $\begin{bmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

2. $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

3. $\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

4. $\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

5. $\begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

6. $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

7. $\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$