**NAME:**

Login name:

# Computer Science 426 Midterm
# 3/11/04, 1:30PM-2:50PM

This test is 5 questions, of equal weight. Do all of your work on these pages (use the back for scratch space), giving the answer in the space provided. This is a closed-book exam -- you may use one-page of notes with writing on both sides during the exam. **Please write out and sign the Honor Code pledge before turning in the test.**

***``I pledge my honor that I have not violated the Honor Code during this examination.''***

| Question | Score |
|---|---|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| Total | |

## Q1: Image Processing

Joe Shmoe has an 8-bit gray-level image with 512x512 pixels (shown on the left below, and labeled “Image A”) -- it stores a sine function with frequency increasing from left to right across the image. Please answer the following questions with one or two sentences each.

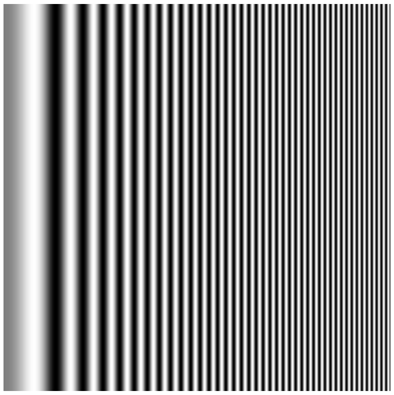


Image A

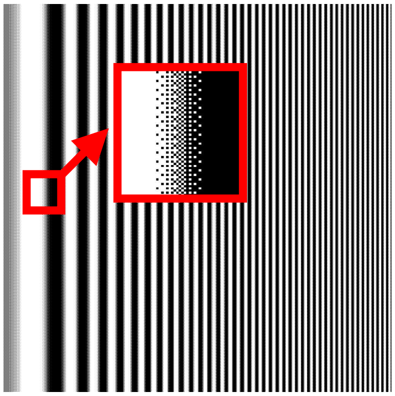


Image B

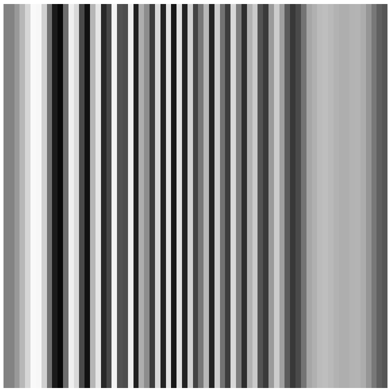


Image C

(a) When Joe displays Image A on a black and white computer screen he gets Image B (in the middle above). Note that a small portion of the image has been zoomed for closer inspection. Why is there a dot pattern in the zoomed region?

(b) Briefly, describe the idea behind the algorithm used to generate the dot pattern in Image B.

(c) When Joe zooms Image A down to 40x40 pixels with the latest program from PhotoIdiot Corporation, he gets Image C (on the right above). Note that the sine wave pattern appears fine on the left side of image C, but a very different pattern appears on the right side. Why?

(d) What should PhotoIdiot Corporation have done when zooming the image down in order to avoid the visual artifacts on the right side of image C?

(e) Describe what the right side of image C would look like if PhotoIdiot Corporation had implemented your solution in part (d).

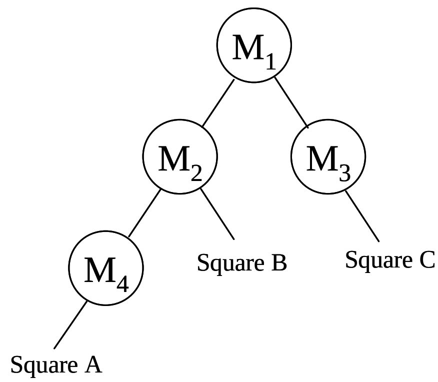
(f) PhotoIdiot Corporation also provides an image warp program implemented with the pseudo-code below. When Joe uses it to perform a fun warp of Image A (compress by the square root in Y dimension), he finds that the image becomes lighter than it should be (especially near the top). Why?

```
for (i = 0; i < src_width; i++)
   for (j = 0; j < src_height; j++)
      dst(i, sqrt(j)) += src(i,j);
```

(g) Write pseudo-code that does not have the problem in part (f). You may use calls to a function *float Gauss(float a, float b, float width)* that returns an estimate of the sample at location *(a,b)* using a Gaussian filter with sigma derived from *width*. *Note: your answer can have as little as three lines of pseudocode.*

## Q2: Modeling Transformations

The figure below shows the transformation hierarchy for a 2D scene in which each node $N_i$ stores a 3x3 matrix $M_i$ used to transform all primitives referenced by its descendant nodes. Each node labeled “Square” represent an axis-aligned 2D square with vertices at (-1,-1) and (1,1) in its local coordinate system. Please answer the following questions. When writing matrices, you may leave your answers as compositions of multiple matrices, where translations are written as T(dx, dy), scales as S(sx, sy), counterclockwise rotations as R(Θ), and other transformations as 3x3 matrices. Assume that points are represented by column vectors with homogeneous coordinates.




$M_1$ = T(0,1)

$M_2$ = S(2, 1)

$M_3$ = T(-3,0)

$M_4$ = T(2,0)

(a) Draw the 2D scene represented by the transformation hierarchy.
Label the coordinate axes and mark the squares “A”, “B”, and “C” in your drawing.

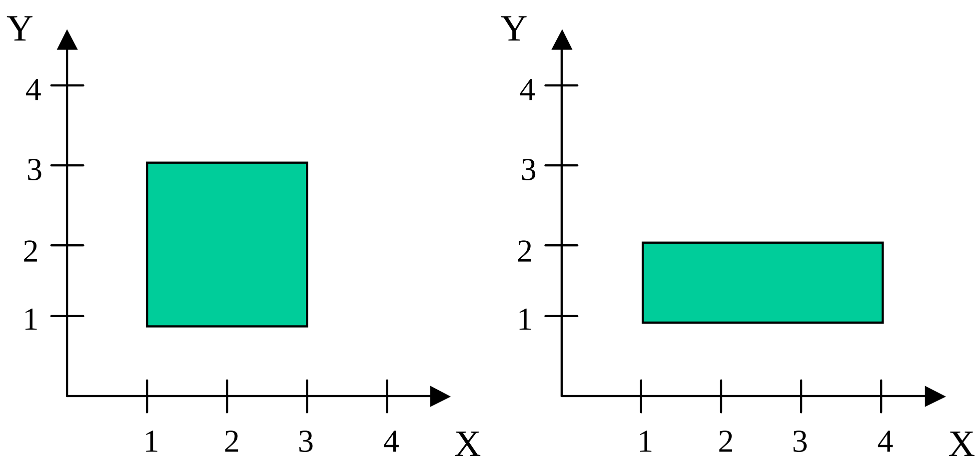
(b) Which matrix(es) would you change to translate the whole scene of part (b) by two units in the positive X direction? Write the new matrix(es).

(c) Which matrix(es) would you change to scale **only** Square "A" to be twice as big while keeping the same position of its center. Write the new matrix(es).

(d) Which matrix(es) would you change to scale **only** Squares "A" and "B" to be twice as big while keeping the same position for the center of B. Write the new matrix(es).

(e) Give at least two reasons why transformation hierarchies are used to represent scenes in computer animations?

(f) Write an expression for the matrix that transforms the box drawn on the left to the box on the right.



## Q3: Viewing Transformations

(a) Write the matrix that transforms a 3D coordinate system with origin O and orthogonal basis vectors $(x_1, y_1, z_1)$, $(x_2, y_2, z_2)$, and $(x_3, y_3, z_3)$ to the standard cartesian coordinate system with the origin at (0,0,0) and basis vectors (1,0,0), (0,1,0), and (0,0,1).

(b) A parameterized matrix that can be used for all possible parallel projections from camera coordinates $(x_c, y_c, z_c)$ to screen coordinates $(x_s, y_s, z_s)$ is provided below. What is the geometric interpretation for the parameters L and $\phi$ (draw a picture)?

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ w_s \end{bmatrix} = \begin{bmatrix} 1 & 0 & L\cos\phi & 0 \\ 0 & 1 & L\sin\phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

(c) Draw a picture of an axis-aligned cube being viewed in parallel projection with the view direction aligned with the negative Z axis and L = 1, $\phi$ = 45 degrees.

(d) The matrix that converts a perspective view frustum to a canonical viewing volume is provided below. What is the geometric interpretation for each parameter (draw a picture)?

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ w_s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/D & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

(e) Draw a picture of an axis-aligned unit cube centered at the origin being viewed in perspective projection with the eye point at (10, 10, 0) and the view direction looking back towards the origin.

(g) Circle all of the classes of transformations to which parallel projections belong?

Linear Affine Projective

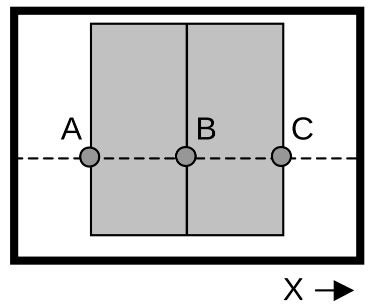
(h) Do parallel projections produce more realistic-looking images than perspective projections? If so, explain why. If not, why do people use them?

## Q4: Polygon Shading

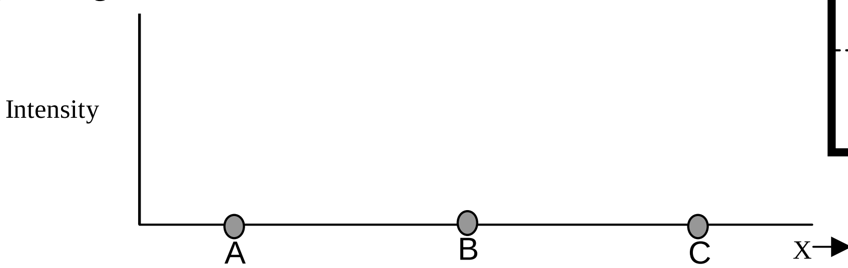
Consider a gray-level image of the 3D cylinder shown below. The cylinder is being viewed with an orthographic projection perpendicular to its vertical axis (shown as a solid line), and it is being lit "head on" by a single directional light source whose direction vector matches the viewer's. Assuming the surface of the cylinder is completely diffuse, draw the gray-level values rendered along a horizontal scan line (e.g., the dashed line) for each of the following rendering techniques.

(a) Write an expression for the intensity across the scan line as a function of x:

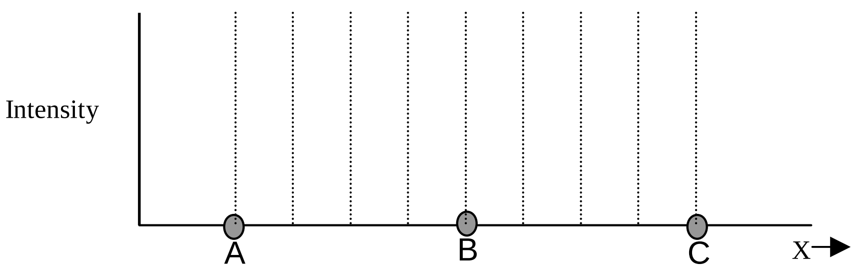
Intensity(x) =



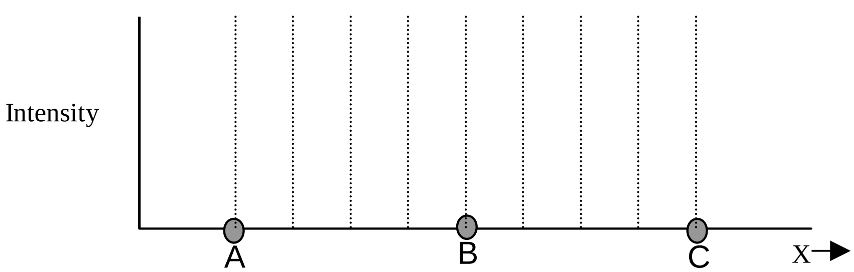

(b) Ray casting:



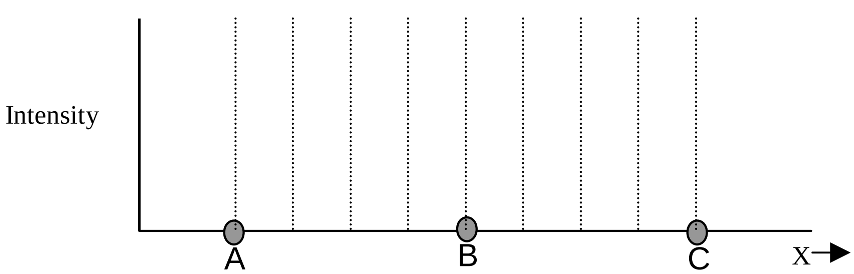

(c) Flat shading of a prism approximating the cylinder with 8 front-facing rectangles:



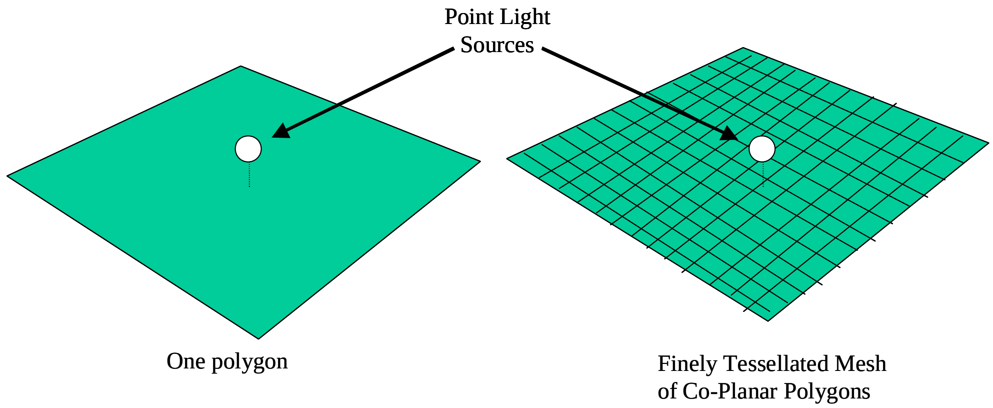

(d) Gouraud shading of a prism approximating the cylinder with 8 front-facing rectangles:




(e) Phong shading of a prism approximating the cylinder with 8 front-facing rectangles:



(f) Consider the scene shown below where a point light source (white dot) hovers a little bit above a nearby planar surface. Discuss the differences between representing the planar surface with a single polygon (left) versus a finely tessellated mesh of co-planar polygons (right) with respect to how an image of it will look when rendered with Gouraud shading.




(g) Same question as (f), except for rendering with Phong Shading.

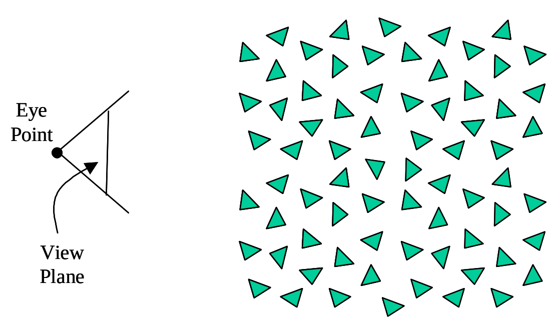
## Q5: Hidden Surface Removal

(a) What is back-face culling?

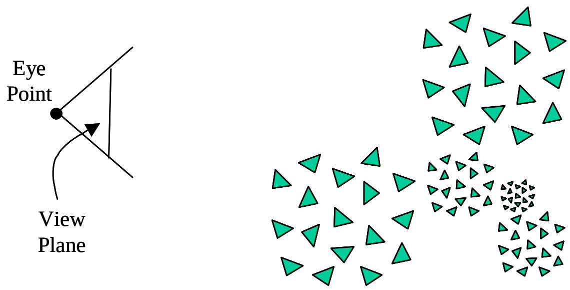
(c) Rank the following hidden surface removal algorithms from least to most with respect to how much more slowly they execute for images with higher spatial resolution: z-buffer, painter's algorithm, ray casting with a spatial indexing structure (e.g., a uniform grid). Explain your answer.

(b) Rank the following hidden surface removal algorithms from least to most with respect to how much more slowly they execute for scenes with higher depth-complexity (more polygons overlapping each pixels at different depths): z-buffer, painter's algorithm, ray casting. Explain your answer.

(d) What spatial data structure would be best for accelerating ray intersection calculations for the scene below. Explain why you chose your answer rather over the other alternatives.



(e) What spatial data structure would be best for accelerating ray intersection calculations for the scene below. Explain why you chose your answer rather over the other alternatives.




(f) Write the equation required to solve for the intersection between a ray and an ellipsoid. The interestection point P is (x,y,z). The ray is defined parametrically by $P = P_0 + tV$, where $P_0 = (x_0, y_0, z_0)$ and $V = (dx, dy, dz)$. The ellipsoid can be defined in either of two ways:

$$\left(\frac{x}{r_x}\right)^2 + \left(\frac{y}{r_y}\right)^2 + \left(\frac{z}{r_z}\right)^2 - 1 = 0$$

$$x = r_x \cos\phi \cos\theta$$
$$y = r_y \cos\phi \sin\theta$$
$$z = r_z \sin\phi$$