University of California San Diego
Department of Computer Science
CSE167: Introduction to Computer Graphics
Midterm Examination
Thursday, October 21, 2010

Instructor: Dr. Jürgen P. Schulze

Name:________________________________

Please write your name or initials at the top of every page before beginning the exam.

Please include all steps of your derivations in your answers to show your understanding of the problem. Try not to write more than the recommended amount of text. If your answer is a mix of correct and substantially wrong arguments we will consider deducting points for incorrect statements. You may not use calculators, notes, textbooks or other materials during this exam, except for one single sided, hand-written 3x5 inch index card. There are ten questions for a total score of 100 points.

Good luck!

**1. Transformations (10 points)**

Computer graphics relies heavily on geometric transformations. Most common are transformations such as rotations, translations, and scales. In the following, T(dx,dy) refers to a translation by (dx, dy), R(a) refers to a rotation by a degrees, and S(sx,sy) refers to a scaling by (sx, sy). For simplicity, assume all transformations are 2D. The order of transformations may matter. Also, sometimes the order may be rearranged, but the arguments will change. State whether the following statements are True or False.

T(1,0) S(2,2) = T(2,0) S(1,1)?

T(-1,0) T(0,2) = T(0,1) T(-1,1)?

R(180) = R(-180)?

R(45) S(2,1) = S(2,1) R(45)?

T(1,0) R(90) = T(0,1)?

R(-90) T(1,0) R(90) = T(0,-1)?

Transformations have inverses. Applying a transformation followed by its inverse has no effect. State whether the following formulas involving inverses are True or False.

$R(180)^{-1} = R(180)$?

$T(1,1)^{-1} = T(-1,-1)$?

$[R(45)\ T(1,0)]^{-1} = T(1,0)\ R(45)$?

$[R(45)\ S(2,2)]^{-1} = R(-45)\ S(.5,.5)$?

**2. Geometry (12 points)**

Points and vectors are not the same. In particular, points and vectors do not transform in the same way.

a) We are given a point P=(px,py,pz) and a vector V=(vx,vy,vz). We now translate by (tx,ty,tz). What is the new position of the point? What are the new coordinates of the vector? (3 points)

b) Suppose we rotate the coordinate system by 90 degrees about the z axis. What is the new position of the point? And the vector? (In this question, ignore the translation in a) (3 points)

c) We wrote a program that computes a new point from two points using the expression p = a * p1 + b * p2, where p, p1 and p2 are points, and a and b are floats. Suppose the points p1 and p2 are translated by (tx,ty,tz). That is, p1' = T(tx,ty,tz) * p1 and p2' = T(tx,ty,tz) * p2. We would expect the point p' = T(tx,ty,tz) * p. Prove whether this is true or false. (3 points)

d) Suppose we compute p = (1-a) * p1 + a * p2. Now we translate p1 and p2 as we did in c). Again, we would expect p' = T(tx,ty,tz) * p. Is this true or false? (3 points)

**3.Computer Graphics (8 points)**

a) Why are triangles the preferred geometric primitive in computer graphics systems like OpenGL? What are the advantages of using triangles compared to other geometric primitives? (4 points)

b) Computer graphics systems often use homogenous coordinates to represent points. In homogenous coordinates, points are represented by 4-vectors, not 3-vectors. How are the 3D coordinates of a point recovered from the 4D homogenous coordinate? (2 points)

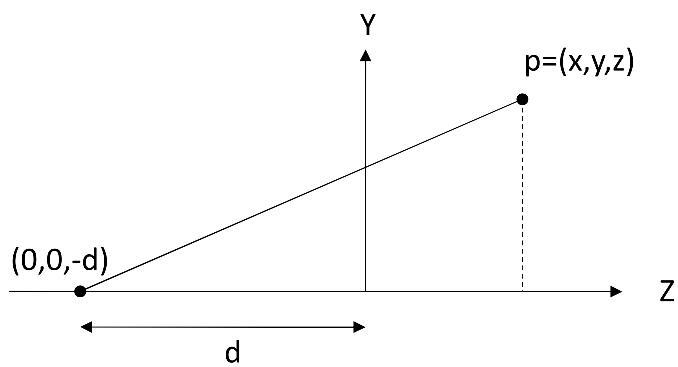
c) Give two reasons that we use homogeneous coordinates in computer graphics? (2 points)

**4. Color Spaces (6 points)**

Patrick decides to stand up to the CIE and choose his own set of primaries in an additive color space (mixing beams of light). Bob's primaries are (R + B) / 2, (B + G) / 2 and (G + R) / 2. In other words, he constructs each of his primaries by taking two CIE primary beams at half the reference intensities, and mixing them together. Is the gamut of his color space smaller than, larger than, or equal to CIE RGB? Briefly justify your answer.

**5. Perspective Projection (10 points)**

In class we constructed a 4x4 projection matrix which performed perspective projection along the Z axis onto a view plane at Z = d for a viewpoint centered at the origin (0,0,0). There are advantages to an alternative formulation in which the viewpoint is at point (0,0,-d) and the view plane is at the origin (i.e., the view plane is just the XY plane). Construct the corresponding 4x4 projection matrix.



**6. Color (10 points)**

Sid wants to render an outdoor scene with a fog effect. He observes that fog has the following characteristics:

- it has a color of its own, usually white but sometimes other colors
- shapes are faded to the fog color with distance

Help Sid write a **fog**(*d*, *src_color*, *fog_color*, *fog_density*) function that takes 4 parameters:

- *d*: the distance to an object
- *src_color*: the original (shaded) color of the object
- *fog_color*: the color of the fog
- *fog_density*: how quickly shapes fade to the fog color – greater density implies a quicker fade

**fog**(...) returns the perceived color of the object seen through the fog. Write either a math formula or pseudocode for a plausible **fog**(...) function. (There is no single correct answer. Your solution should satisfy Sid's two criteria at least.)

**7. Rasterization Artifacts (10 points)**

a) Why does linear interpolation of texture coordinates in screen space lead to artifacts? Explain using a sketch and 2-3 explanatory sentences. (5 points)

b) What technique can be used to correct these artifacts? Explain its basic idea in 2-3 sentences. (5 points)

**8. Shading (12 points)**

a) Both Gouraud and Phong Shading interpolate along polygon edges to compute intensities. But the two shading models interpolate different things.

1) What does Gouraud Shading interpolate along edges? (2 points)

2) What does Phong Shading interpolate along edges? (2 points)

b) What is the difference between Phong Shading and the Phong Illumination Model? (2 points)

c) What two parameters does a Bidirectional Reflectance Distribution Function (BRDF) take in, and what does it calculate? (3 points)

d) Which three components does the simplified illumination model OpenGL uses consist of? (3 points)

**9. Shader Programs (10 points)**

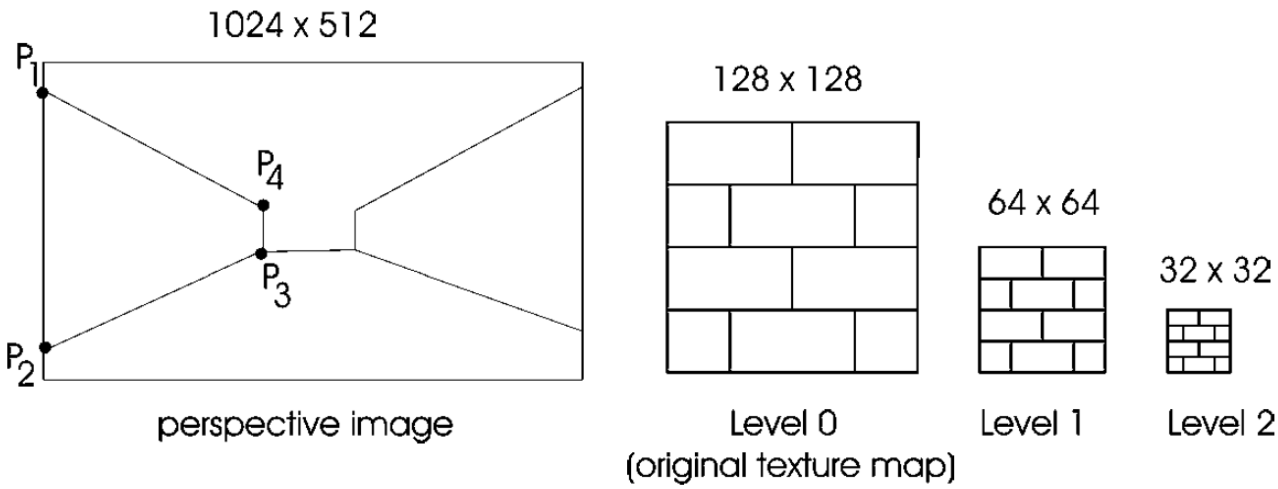
a) Assuming that both vertex and fragment programs have been loaded for rendering a triangle. How many times will the vertex program get executed, how many times the fragment program? (4 points)

b) Name two example each for what vertex and fragment programs can be used for. (4 points)

c) What two types of input data does a vertex shader accept? (2 points)

**10. Mip-Mapping (12 points)**

Suppose we have a brick wall that forms the left-hand wall of a corridor in a maze game, as shown in the image below, and it is defined (in world coordinates) by points P1, P2, P3, P4. Assume that the brick wall is to be 16 bricks high and 200 bricks long.




a) Using the height of the brick wall as seen in the image, estimate (with derivation) how many texels a screen pixel covers, both for near points on the wall, i.e., on the edge P1P2, and at distant points on the wall, i.e., on the edge P3P4. (3 points)

b) In the perspective image above, sketch approximately what regions of the wall will use each of the levels of the mip-map image pyramid on the right. (3 points)

c) Unrelated to the above image: How many texel values have to be read to perform nearest neighbor texture filtering, bilinear texture filtering, and trilinear mipmapping? (3 points)

d) In which order are the texel values in each of the three methods above being averaged? (3 points)