Computer Graphics Comprehensive Exam

2007-2008

NAME:

Note: This is exam is *closed-book.*

The exam consists of 5 questions. Each question is worth 20 points. Please answer all the questions in the space provided, overflowing on to the back of the page if necessary.

This exam has been designed to take 1 hr.

1. [20 points] General.

1A [5 points]. What is the relationship between the RGB and CMY color spaces? How are colors in one space computed from colors in the other space?

RGB is an additive color space, CMY is a subtractive coor space.

R = 1-C
G = 1-M
B = 1-Y

(Need these formulas to get full credit)

1B [5 points]. What is a z-buffer and how is it used in computer graphics?

The z-buffer is an image buffer that holds the depth of the closest surface visible at each pixel. The z-buffer is used in hidden surface elimination. As each surface is rasterized, the depth of that fragment is compared with the depth in the z-buffer. If the fragment is closer, it is drawn and the z-buffer is updated. If the fragment is further, it is discarded.

1C [5 points]. What is quantization? And why can it be used for compression?

Quantization is the process of mapping a continuous (higher-precision) value to a discrete (lower-precision) value. A set of values in the input range are associated with a single value in the output range. Since there are fewer values to store, fewer bits are needed to represent the output set of values.

1D [5 points]. Aliasing is an undesirable artifact of generating images using computers. What causes aliasing?

Aliasing is caused by undersampling a signal. Frequencies that are greater than ½ the sampling rate will appear as aliases. Aliases are frequency components that appear at different frequencies.

(Sampling need not generate aliasing; merely creating an image may not cause aliasing.)

[20 points] Shading Polygon Meshes

Polygon meshes are the most common representation of geometric shapes in computer graphics. Many computer graphics algorithms operate on polygon meshes. For this question, assume that the mesh is made of polygons with any number of sides.

2A [5 points]. What is the difference between Gouraud shading and Phong shading?
In Gouraud shading, shading is performed at each vertex to generate a color, and these colors are interpolated over the polygon. In Phong shading, a normal is computed at each vertex, these normals are interpolated over the polygon, and shading is performed at each pixel.

All shading algorithms require a surface normal. In the case of Phong and Gouraud shading, a surface normal is required for each vertex in the mesh. Assume that you are given a polygon mesh without normals. That is, with just a position for each vertex.

2B [15 points]. Describe a method for computing the normal at each vertex. Your explanation should explain the math of how to compute the normal from the vertex positions (7.5 points) and how to define a polygon mesh data structure that makes this computation efficient (O(n) where n is the number of vertices in the mesh).

Three vertices define a triangle and the normal to the triangle can be computed using the vector cross-product. If the vertices are p0, p1, p2, then n = (p1-p0) x (p2-p0).

The normal to a polygon is often computed by averaging the normals from each set of three adjacent vertices. This requires a data structure to store the list of vertices associated with each polygon.

The normal at a vertex is then computed by averaging the normals of the adjacent polygons. This requires a data structure to store the list of polygons adjacent to a vertex.

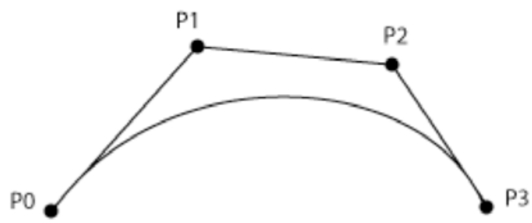
Here is one possible set of data structures:

```
struct { float p[3], n[3]; List *polygons; } Vertex;
struct { List *vertices; } Polygon;
```

There are lots of variations that are acceptable and any reasonable data structure was given full credit.

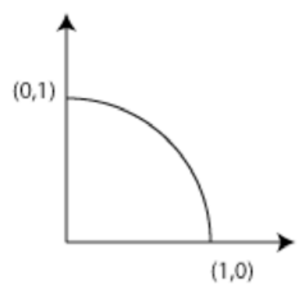
3. [20 points] Bezier Curves.

One the most important primitives in computer graphics is the cubic Bezier Curve. The cubic Bezier curve is constructed from a control polygon consisting of 4 points as shown below.



The curve is given by the parametric equation P(t) where t lies between 0 and 1. This diagram illustrates two key properties of Bezier curves. First, the curve passes through the end points P0 and P3. That is, P(0)=P0 and P(1) = P3. Second, the curve is tangent to the control polygon at the end points. Precisely, T(0)=3(P1-P0) and T(1)=3(P3-P2). P(t) can be evaluated using a recursive algorithm. Recall the classic result that P(1/2) = 1/8 P0 + 3/8 P1 + 3/8 P2 + 1/8 P3.

One reason Bezier curves are so popular is that they can be used to accurately approximate other curves. Suppose you want to approximate the circular arc shown below. Where would you position the control points P0, P1, P2, P3 to approximate this arc with a bezier curve whose P(1/2) is the arc's midpoint (cos 45, sin 45)? Provide the coordinates of the positions of these points.



v = 4/3 (sqrt(2) – 1)

P0 = (1,0)
P1 = (1,v)
P2 = (v,1)
P3 = (0,1)

4. [20 points] Shading

There are two important shading models used to model the appearance of real materials, the diffuse and specular (sometimes called glossy) models. In this problem, just consider point light sources.

4A [10 points]. What is the equation for diffuse reflection? That is, what is the reflected color given the direction to the eye **E**, the surface normal **N**, and the direction of the point light source **L**. Include other parameters that might be useful. Explain briefly the physics underlying diffuse reflection. Finally, give an example of a material whose appearance can be modeled using diffuse reflection.

kd Cl max(L . N, 0), here kd is the diffuse reflection coefficient and Cl is the color of the light.

In diffuse reflection, light is scattered equally in all direction. The factor N . L gives the amount of light falling on the surface.

Chalk is a good example of a diffuse material.

4B [10 points]. Give an equation for specular or glossy reflection. Explain briefly the physics underlying this type of reflection. Give an example of a material that can be modeled using specular reflection.

ks Cl (N . H)^s, here ks is the specular reflection coefficient and H is the vector halfway between L and E (H = (L+E)/len(L + E).

In specular reflection, the surface behaves as a mirror. In the above formula, (N.H)^s gives the number of microfacets (small mirrors) oriented to reflect light.

Plastic and metals are good examples of glossy materials.

Full credit was also given for using ideal specular reflection from a true mirror, and the Phong Reflection Model.

5 [20 points] Matrices and the Matrix Stack

5A [5 points]. Transformations in computer graphics are represented with 4x4 matrices even though the world is three-dimensional. Why are 4x4 matrices used instead of 3x3 matrices?

Adding an extra 4th coordinate allows the use of homogenous coordinates, which permit the representation of translations and perspective transformations as matrices.

In a typical graphics system, a current transformation matrix T is part of the graphics state. Each time a triangle or line is drawn, this matrix transforms the vertices. Assume that the vertex position p is given by a column vector. With this assumption, the transformation involves a matrix-vector product, T * p.

5B [5 points]. In a graphics program, commands exist to modify the current transformation. For example, translate(x,y,z) and rotatex(30). What happens in the graphics system when a transformation command is given? Be precise.

Suppose the current transformation matrix is T, and that Tr represents the translation and R a rotation. Then specifying T causes T = T * Tr, and specifying R causes T = T * R. The matrix is post-multiplied onto the CTM because the CTM premultiplies the position vector.

5C [5 points]. The order of the transformations sometimes matters. That is, a program that first executes translate(x,y,z) and then executes rotatex(30) may lead to a result different than one that first executes rotatex(30) and then translate(x,y,z). Why does the order matter? If order does not always matter, can you give an example when it does not matter?"

Order matters because in general matrices do not commute; alternatively many transformations do not commute. However, some transformations do commute. For example, two translates will commute, so the order does not matter.

5D [5 points]. Commands also exist to push and pop the current transformation onto a stack. Why is it useful to be able push and pop the current transformation?

Many applications use hierarchical models such as a skeleton. The hierarchies are drawn recursively. When a node in the hierarchy is drawn, the current transformation is pushed, the node drawn, and then the current transformation is popped.