Computer Graphics - Honors Exam 2021

April 15, 2021

Name ________________________________

Geometry ______________

Transformations ______________

Graphics Pipeline ______________

Rasterization ______________

Rendering witeekh Rays ______________

Shading and Texturing ______________

Interactive Graphics ______________

Total ______________/
Instructions

This exam is open book, open computer. All work should be your own. All work should be completed within 3 hours.
Useful formulas

<table>
<thead>
<tr>
<th>Angle (degrees)</th>
<th>Angle (radians)</th>
<th>sin(Angle)</th>
<th>cos(Angle)</th>
<th>tan(Angle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>$\frac{\pi}{6}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{\sqrt{3}}{2}$</td>
<td>$\frac{1}{\sqrt{3}}$</td>
</tr>
<tr>
<td>45</td>
<td>$\frac{\pi}{4}$</td>
<td>$\frac{\sqrt{2}}{2}$</td>
<td>$\frac{\sqrt{2}}{2}$</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>$\frac{\pi}{3}$</td>
<td>$\frac{\sqrt{3}}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\sqrt{3}$</td>
</tr>
<tr>
<td>90</td>
<td>$\frac{\pi}{2}$</td>
<td>1</td>
<td>0</td>
<td>undefined</td>
</tr>
<tr>
<td>180</td>
<td>$\pi$</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>270</td>
<td>$\frac{3\pi}{2}$</td>
<td>-1</td>
<td>0</td>
<td>undefined</td>
</tr>
</tbody>
</table>

Given vectors $\mathbf{u}$ and $\mathbf{v}$ in $\mathbb{R}^3$, the **dot product** is

$$\mathbf{u} \cdot \mathbf{v} = u_x v_x + u_y v_y + u_z v_z$$

The dot product is related to cosine by the following formula

$$\cos(\theta) = \left( \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}|| ||\mathbf{v}||} \right)$$

In $\mathbb{R}^3$, the **cross product** of two vectors $\mathbf{u}$ and $\mathbf{v}$ is

$$\mathbf{u} \times \mathbf{v} = (u_y v_z - u_z v_y)\hat{i} + (u_z v_x - u_x v_z)\hat{j} + (u_x v_y - u_y v_x)\hat{k} = \begin{bmatrix} u_y v_z - u_z v_y \\ u_z v_x - u_x v_z \\ u_x v_y - u_y v_x \end{bmatrix}$$

A rotation $\theta$ around each of the $x$-, $y$-, and $z$- axes corresponds to the following rotation matrices

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, R_y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, R_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
1 Geometry

Consider the following points.

\[ p_1 = (1, 0, 0)^T \]
\[ p_2 = (2, 4, 0)^T \]
\[ p_3 = (3, -1, 0)^T \]
\[ p_4 = (4, 4, 0)^T \]

1. Draw and compute the vectors between \( p_2 - p_1 \), and \( p_3 - p_1 \)

2. Give a formula for the angle between the vectors \( p_2 - p_1 \) and \( p_3 - p_1 \)?

3. What is the distance between \( p_2 \) and \( p_1 \)?

4. Compute the unit vector that points in the direction \( p_2 - p_1 \).

5. Consider the triangle created by the points \( p_1 \), \( p_2 \), and \( p_3 \). Give a formula for the normal to the plane containing the triangle.

6. Consider the direction \( p_4 - p_1 \). Give a formula for the portion of this vector which is parallel to \( p_2 - p_1 \).

7. Consider the direction \( p_4 - p_1 \). Give a formula for the portion of this vector which is perpendicular to \( p_2 - p_1 \).
2 Transformations

1. Suppose a cube is located at position $p = (1, -4, 3.5)$. What is its homogeneous coordinate?

2. Suppose a ray direction is $d = (1, -4, 3.5)$. What is its homogeneous coordinate?

3. Suppose we are using XYZ euler angles. Give an expressions for the 3x3 matrix that corresponds to (30, 0, 45) degrees euler angles. Your expression should be in terms of 3x3 matrices around each of the x, y, and z axes.

4. Give a 3x3 matrix that scales the height of an object by 2.

5. Give a 4x4 matrix that translates an object by (10, -5, 2).
6. Suppose we have a cube located at the origin. We wish to scale its height by 2, rotate it 30 degrees around the X axis, and then move it to location \((10, -5, 2)\). What transformation matrix \(T\) corresponds to the set of transformations?

7. What is the inverse matrix of the matrix \(T\) computed previously?

8. Consider the original UP direction of our cube, e.g. \((0,1,0)\). What direction will this vector be if we apply the inverse transpose of the transformation \(T\) computed previously?

9. Consider a unit cube at the origin. It’s up direction is \((0,1,0)\) and its forward direction is \((0,0,1)\). Suppose we want the cube’s forward direction to align with the direction \((1, 3, 1)\). Derive the rotation matrix which points the cube’s forward direction towards \((1, 3, 1)\).
3 Graphics Pipeline

The graphics pipeline describes a standard series of steps for rendering a 3D scene onto a 2D raster image.

Describe how a typical real-time application which renders a 3D cube would be drawn using the graphics pipeline. What sub-tasks would need to performed in each step?

1. Application

2. Geometry Processing

3. Rasterization

4. Pixel Processing

The graphics pipeline is an example of an object-order rendering algorithm as opposed to an image-order rendering algorithm, such as ray marching. How does an object-order rendering approach differ from an image-order approach?
4 Rasterization

Suppose we are rastering 2D lines to an image using the following algorithm

DrawLine(ax, ay, bx, by):
\[
\begin{align*}
y &= ay \\
W &= bx - ax \\
H &= by - ay \\
F &= 2*H - W \\
\end{align*}
\]

for (x = ax; x <= bx; x++):
\[
\begin{align*}
&\text{fill}(x,y) \quad // \text{color the pixel black} \\
&\quad \text{if } F > 0: \\
&\quad \quad y += 1 \\
&\quad \quad F += 2*(H-W) \\
&\quad \text{else:} \\
&\quad \quad F += 2*H \\
\end{align*}
\]

1. Suppose we call the function \textbf{DrawLine}(0, 1, 3, 2) \textbf{Note}: the origin is in the top, left corner. What pixels would the algorithm fill in? Show your answer on the grid above.

(a) What is the starting value for W?
(b) What is the starting value for H?
(c) What is the starting value for F?

For each iteration of the algorithm’s loop, fill in the values in the following table

<table>
<thead>
<tr>
<th>iteration</th>
<th>x</th>
<th>y</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. Suppose we wish to draw a line from (3, 2) to (0, 1). Does the above algorithm work? If not, say what we need to change in DrawLine to support this case.

3. Suppose we wish to draw a line from (0, 3) to (4, 2). Does the above algorithm work? If not, say what needs to change in DrawLine to support this case.

4. What assumptions does the given function DrawLine make?
5 Raytracing

Below is code for an optimized intersection test between a ray and a sphere. The corresponding diagram shows the quantities used in the algorithm. Assume the input ray direction is normalized. This function returns the time $t$ at which point the ray hits the sphere, or -1 if the ray misses the sphere.

```cpp
RaySphereIntersection(center, radius, ray)
    el = center - ray.origin
    s = dot(el, ray.direction)
    elSqr = dot(el, el);
    rSqr = radius*radius
    if (s < 0 and elSqr > rSqr) return -1 // case 1
    mSqr = elSqr - s*s
    if (mSqr > rSqr) return -1 // case 2
    q = sqrt(rSqr - mSqr)
    if (elSqr > rSqr) t = s - q // case 3a
    else t = s + q // case 3b
    return t
```

5. $q$ must always be less than or equal to $r$. Why?

6. Why do we return -1 when $s < 0$ and $elSqr > rSqr$? Draw a ray and sphere which would trigger this case.

7. Why do we return -1 when $mSqr > rSqr$? Draw a ray and sphere which would trigger this case.

8. Consider the case 3a. Draw a ray and sphere which would trigger this case.

9. Consider the case 3b. Draw a ray and sphere which would trigger this case.
6 Shading and Texturing

The phong shading model computes a color as a combination of ambient, diffuse, and specular reflections of light from a surface. Suppose we wish to shade a vertex \( p \). We have the following quantities defined.

- \( p \in \mathbb{R}^3 \): the hit point
- \( n \in \mathbb{R}^3 \): the hit normal (unit length)
- \( L \in \mathbb{R}^3 \): the direction from the light position and the hit point (unit length)
- \( V \in \mathbb{R}^3 \): the direction from the camera to the hit point (unit length)
- \( R \in \mathbb{R}^3 \): the reflection of the light around the hit normal (unit length)
- \( c_a \in \mathbb{R}^3 \): the ambient color, components in range \([0,1]\)
- \( c_d \in \mathbb{R}^3 \): the diffuse color, components in range \([0,1]\)

The final color is

\[
I = I_a + I_d + I_s
\]

where

\[
I_a = c_a \\
I_d = c_d (n \cdot L) \\
I_s = (R \cdot V)^{\text{shininess}}
\]

1. We are visualizing the normals of a sphere using the following formula:

\[
\text{color} = 0.5 \times (\text{normal} + \text{vec3}(1,1,1));
\]

In our scene, the Y direction is UP and positive Z points out of the page. Are the normals of our sphere correct? Why or why not?
2. We are using the given Phong model to shade a sphere. Our scene has a single light source in the top, right of the rendered view. We expect there to be a single highlight at the top, right of the sphere but instead there are two strong highlights. Broadly, what are some errors that may be in our shader implementation?
3. Suppose we wish to texture a flat quad with vertices (-1,-1,0), (-1,1,0), (1,1,0), and (-1,1,0). Assume our view has Y up and Z pointing out of the page. What texture coordinates should we set so that the left, bottom corner of the image is at vertex (-1,-1,0) and the top, right corner of the image is at position (1,1,0)? Draw your answer on the quad below.

4. Suppose we wish to texture a flat quad with vertices (-1,-1,0), (-1,1,0), (1,1,0), and (-1,1,0). Assume our view has Y up and Z pointing out of the page. What texture coordinates should we set so that the image is repeatedly tiled 5 times horizontally and 3 times vertically? Draw your answer below.
5. Sketch the code for a fragment and vertex shader which draws our textured quad using the Phong model above. Your code should include variables for the inputs and outputs of each shader as well as pseudocode to implement the lighting equations.
7 Interactive graphics

Give a design (e.g. software architecture) for a first person shooter game, such as the original Doom (1993), written in a low-level graphics API such as OpenGL, webGL, or DirectX. Your game should have a player character and targets. You may assume that you can load pre-made assets for the terrain, character, and targets. Describe the different modules that you would need to implemented and how you would fit the various components together.
FPS Design (cont).
FPS Design (cont).