TRANSFORMS AND TEXTURE MAPPING 2

CS184: COMPUTER GRAPHICS AND IMAGING

Feb 1-2, 2021

1 Sampling

1. What is the connection between applying a box blur to an image and supersampling each pixel within the image?

2. We have two wheels that rotate at different speeds with different numbers of spokes that we want to record with our camera.

   - **Wheel A** has 4 spokes and rotates at a rate of 6 rotations per second.
   - **Wheel B** has 6 spokes and rotates at a rate of 5 rotations per second.

   What is the lowest integer frame rate (in frames per second) we need on our camera to avoid any aliasing effects?

2 Basic Transforms

1. Which of the following transforms preserves the distances between every pair of points on an object? (Rotation, Translation, Scaling, Reflection, Shearing)

2. Write the transformation matrix for a $30°$ counterclockwise rotation (assuming homogenous coordinates in 2D).

3. Write the transformation matrix for a 2D object that is reflected across the y-axis, then translated up by 1 unit, to the right by 3 units, then rotated around the origin by $90°$ counterclockwise. (Leaving it as a product of matrices is fine).

3 Texture Mapping

3.1 Barycentric Coordinates

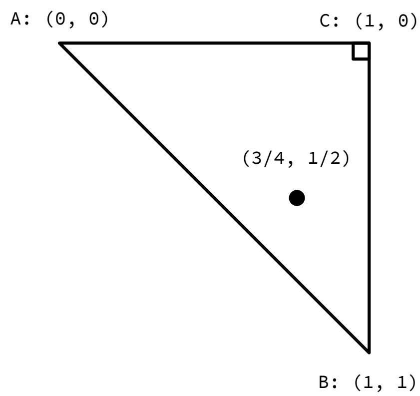
Since triangles form the basic building blocks of most 3D models, we use a special coordinate system to locate points within a triangle relative to its vertices.

The first step in mapping a texture onto a triangle is to convert the screen pixel to barycentric coordinates. These coordinates (α, β, γ) can be thought of as the weights being assigned to each vertex such that the weighted average of the vertices forms a screen-space coordinate. That is,

$$(x, y) = \alpha A + \beta B + \gamma C$$

1. What happens if α, β, or γ is less than zero?

2. What is the barycentric coordinate of the point corresponding to the screen-space coordinates $(x, y) = (\frac{3}{4}, \frac{1}{2})$ in the following diagram? (Hint: You don't have to use the closed-form solution, you can figure it out from just the definition of barycentric coordinates).



3. Suppose we specify colors at each of the vertices in order to color the triangle. If the RGB values of (A, B, C) are (`(1, 0, 0)`, `(0, 1, 0)`, `(0, 0, 1)`), then what is the interpolated color at the selected point above?

3.2 Texture Coordinates and Mipmaps

Now that we have the barycentric coordinates (α, β, γ), we can convert them to texture-space coordinates (u, v). To do this, we use the same interpolation trick as we did with the colors above - we specify coordinates A_{uv}, B_{uv}, C_{uv} on the texture itself, then use our barycentric weights to compute $(u, v) = \alpha A_{uv} + \beta B_{uv} + \gamma C_{uv}$.

Sometimes pixels lie in-between texels, and a pixel may cover an area much larger or smaller than a single texel. When this occurs, aliasing artifacts are often visible when textures are actually rendered. One way of dealing with this issue is to filter while rasterizing, but that can often be computationally intensive. Instead, we often store prefiltered textures in the form of a mipmap to approximate the correct result and reduce aliasing.

1. The resolution of a mipmap at level 3 is what fraction of the size of the original texture?

2. A mipmap takes up what fraction of the original's size in memory?

3. How do we determine which level of a mipmap to use for a particular screen-space pixel (x, y)?

4. How can we combine values from two neighboring mipmap levels and why would we do that?