Texture Mapping & OpenGL

Textures

• Textures are cool
• Except sometimes when they’re not
• Why that happens & what to do
• Other cool things to do with textures

We want to “glue” the image onto the triangle.
How should the texture be situated?

Texture Mapping Algorithm
takes texture coordinates,
geometric coordinates, and a point p.

Algorithm
1. Find the texture coordinates for p.
2. Find the color of the texture at the
   texture coordinates for p.

If p is a vertex the answer is easy.
Suppose \( p \) lies on an edge...

Interpolate based on endpoints of edge.

Suppose \( p \) is an interior point.

Draw a line through \( p \) and find its intersection points with the triangle edges.

Compute the texture coordinates of the intersection points.

Interpolate along the line to find the texture coordinates of \( p \).
What if we had chosen a different line?

Doesn't matter — we'd end up with the same texture coordinates.

pipeline: what line should we choose?

triangle in 3D texture

Where are we? 3D or 2D?

scan line algorithm

For each scan line:
1. Find edge/scan line intersection points.
2. Order by x-coordinate.
3. Use odd-even test to turn on pixels.
4. Interpolate to compute color/texture.

scan line algorithm

perspective demo
correct vs. incorrect

distortion is angle-dependent

interpolate in 3D or 2D?

SO OPENGL TAKES CARE OF THIS FOR US ...RIGHT?

so what can you do?
increase polygon count

Algorithm
1. Find the texture coordinates for p.
2. Find the color of the texture at the texture coordinates for p.

beyond triangles
• OpenGL typically converts general polygons into triangle fans or triangle strips.

Algorithm
• Find the texture coordinates for p.
• Find the color of the texture at the texture coordinates for p.

To Do!

blending
Texel color = pixel color
  = ambient response
  = diffuse response
  etc.
opengl howto: woo ch 9

• create a texture object
• specify resampling method, etc.
• specify the texture
• enable texture mapping
• draw scene supplying texture coordinates for each vertex

| OpenGL supports nearest and bilinear interpolation |
| • can choose different interpolation method for minification and magnification of textures |

• create a texture object

```c
unsigned int hTexture;
glGenTextures(1, &hTexture);
glBindTexture(GL_TEXTURE_2D, hTexture);
```

• specify resampling method, etc.

```c
glTexParameteri(GL_TEXTURE_2D, GL_MIN_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D, GL_MAX_FILTER, GL_NEAREST);
```

• specify the texture

- read texture
- height and width must be a power of 2

```c
glTexImage2D(..., image)
```

• enable texture mapping
• draw scene

```c
glEnable(GL_TEXTURE_2D);
glTexParameteri(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);
gBegin(GL_TRIANGLES);
gTexCoord2f(0,0);
gVertex3f(0,0,0);
gTexCoord2f(1,0);
gVertex3f(10,0,0);
gTexCoord2f(.5,1);
gVertex3f(5,5,0);
gEnd();
```
ideal interpolation

neighborhood size increases with distance from the viewer

mipmapping

billboarding

use picture of complicated object instead of complicated geometry

screen aligned billboard

same up

toward

view plane

world oriented billboard

billboard

world up

toward

view plane

billboard trees
**viewplane vs. viewpoint aligned**

**exercise**

determine the rotation transforms for each of these billboarding techniques

LAB TONIGHT: Implement