Perspective correct interpolation
Perspective correct interpolation

- In pipeline assignment, we found barycentric coordinates in 2D screen space
- but not the correct object space barycentric coords
- these coordinates were okay for z-buffer test
\[ u = \frac{1}{2} u_1 + \frac{1}{2} u_2 \]
\[ u = \frac{1}{2} u_1 + \frac{1}{2} u_2 \]
Interpolation with screen space weights is incorrect

\[ u = \frac{1}{2} u_1 + \frac{1}{2} u_2 \]
Perspective correct interpolation

Using screen space weights looks wrong for textures

[Heckbert and Morton, 1990]
Do we need to transform back to object space?

\[ u = \frac{1}{2} u_1 + \frac{1}{2} u_2 \]

\[ v_{sc} = M_{vp} M_{pers} M_{cam} v \]
Do we need to transform back to object space?

NO!

\[ u = \frac{1}{2} u_1 + \frac{1}{2} u_2 \]
Environment mapping
Environment Mapping

Use a texture for the distant environment to simulate the effect of ray tracing more cheaply.
• Project objects in the environment onto sphere centered at eye
• Unwrap and store as texture
• Use reflection direction to lookup texture value
Cube Mapping

- Compute six projections, one for each wall
- Store as texture
- Use reflection direction to lookup texture value
Different environment maps

- Blinn/Newell latitude mapping
- OpenGL spherical mapping
- Cube mapping

www.reindelsoftware.com
Environment Mapping

Create the effect of a mirror with two-pass rendering

1. First pass: render the scene from the perspective of the mirror
2. Second pass: render from original pov; use the first image as a texture for the mirror
Shadow Mapping

first pass from light’s perspective

1. render scene from pov of light and store z-buffer in a texture

2. when rendering scene from desired pov, also render from light pov and test pixel against stored texture
Bump Mapping

perturb normal vectors
doesn’t affect silhouette
Normal Mapping

- original mesh: 4M triangles
- simplified mesh: 500 triangles
- simplified mesh and normal mapping: 500 triangles