

# CS230 : Computer Graphics

Lecture 8: Texture Mapping (cont.)

Tamar Shinar

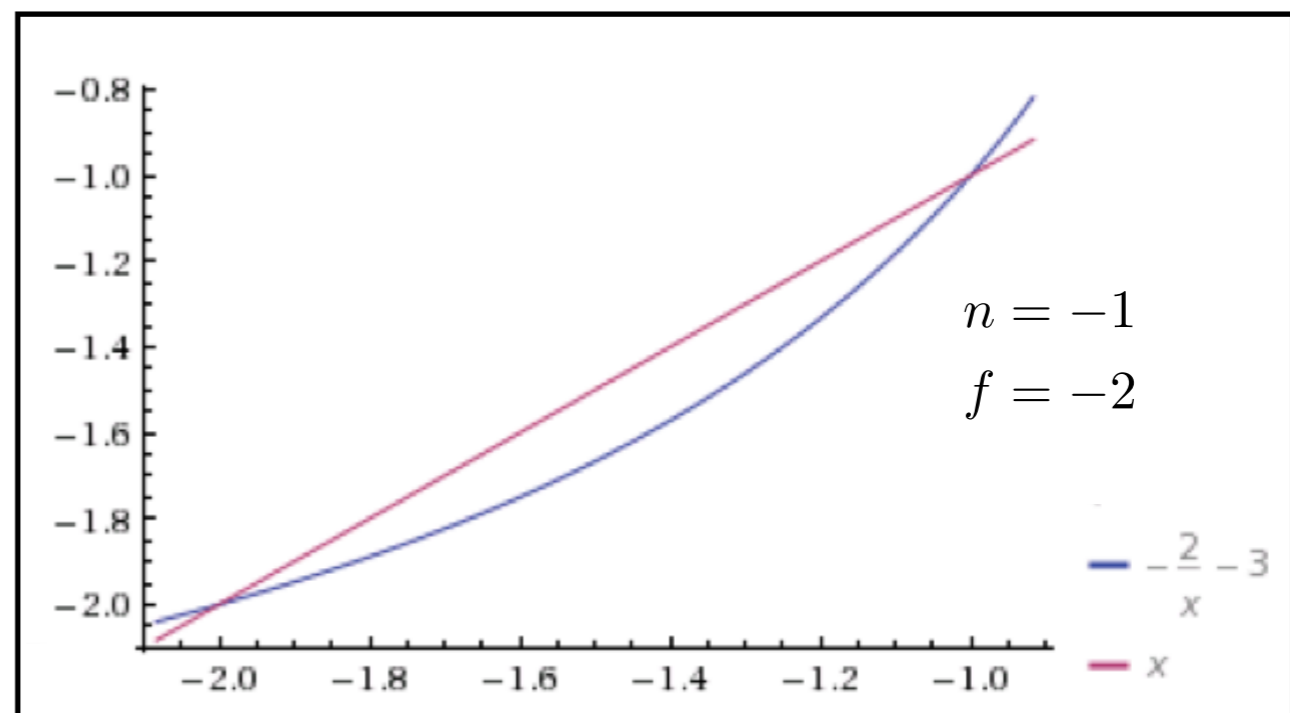
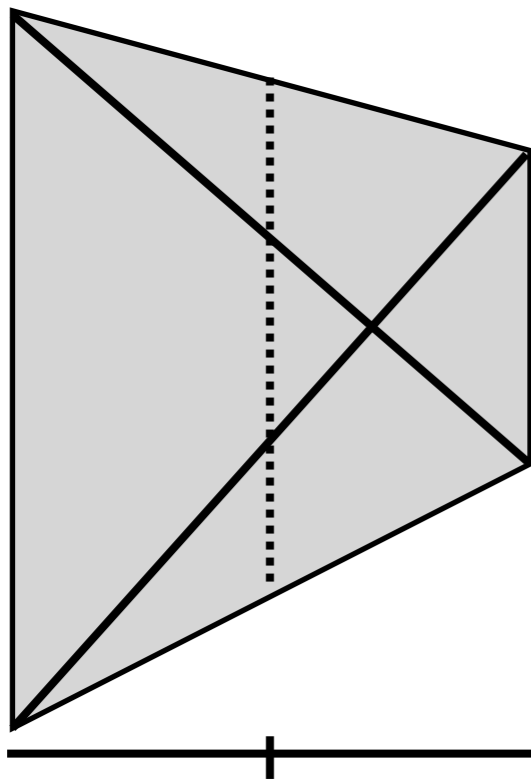
Computer Science & Engineering

UC Riverside

# Perspective correct interpolation

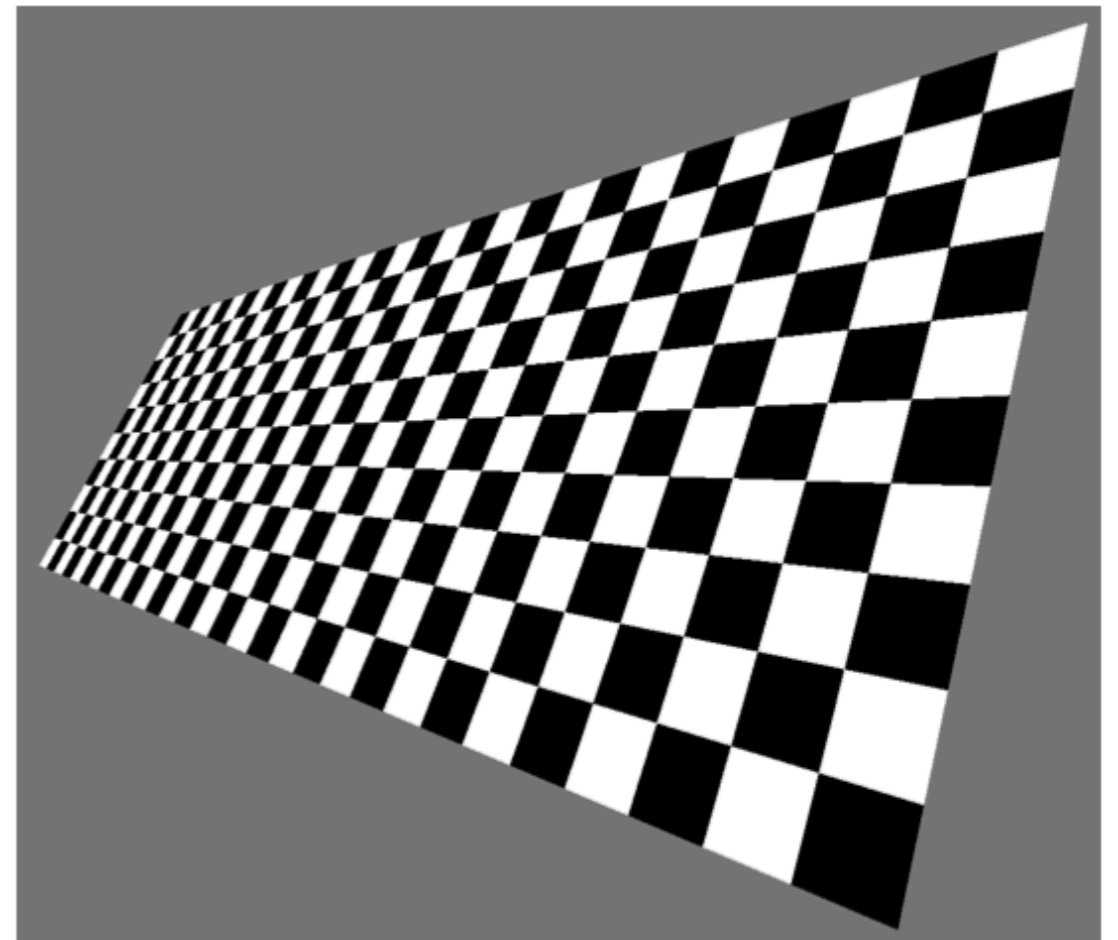
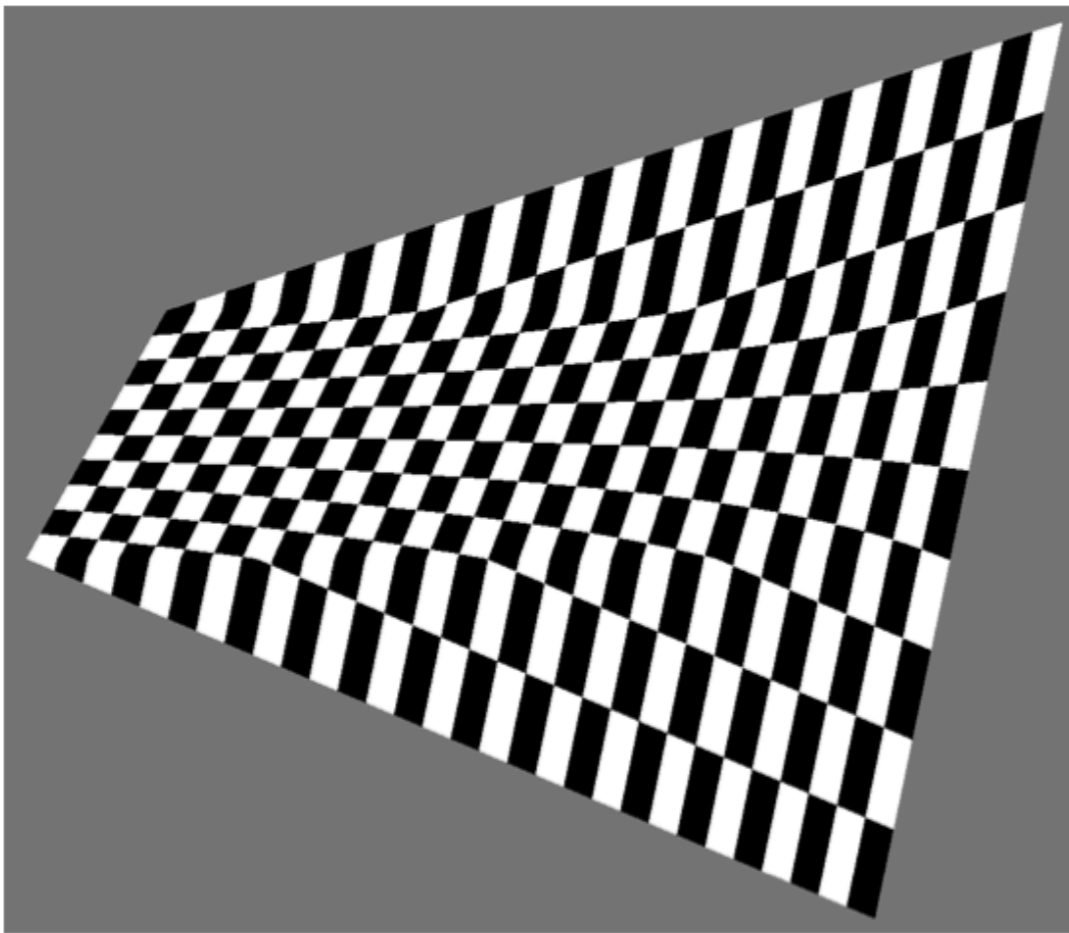
# Perspective correct interpolation

- In assignment 1, we found barycentric coordinates in 2D screen space
- but not the correct object space barycentric coords
- these coordinates were okay for z-buffer test



# Perspective correct interpolation

Using screen space bary. coords. looks wrong for textures



Heckbert and Morton, 1990

# Perspective correct interpolation

Using screen space bary. coords. looks wrong for textures



Wikimedia Commons

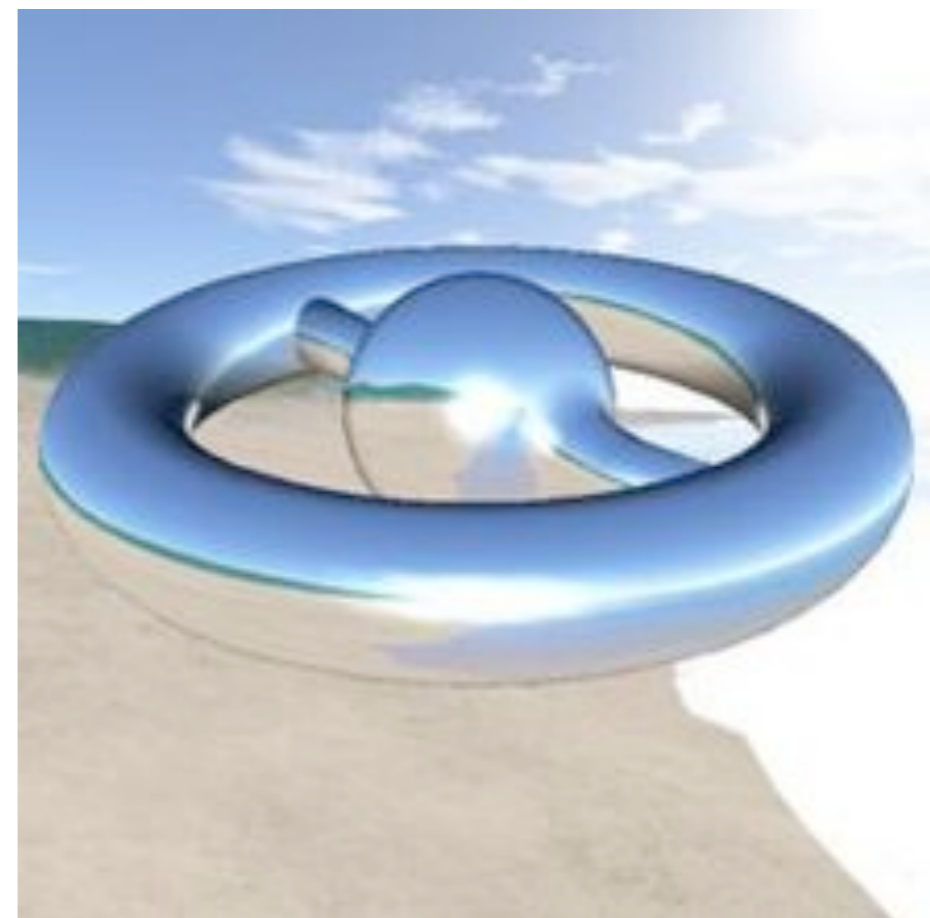
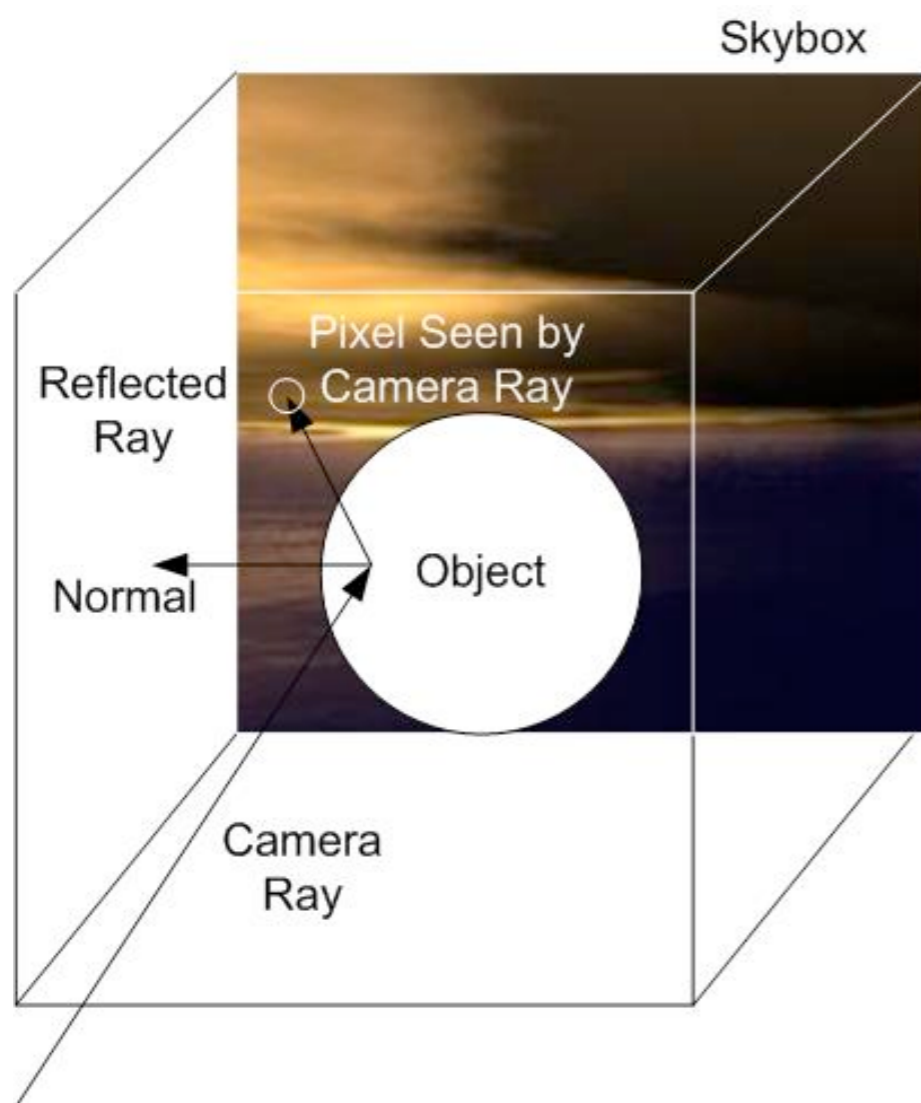
<whiteboard>

# Environment mapping



# Environment Mapping

Use a texture for the distant environment  
simulate the effect of ray tracing more cheaply

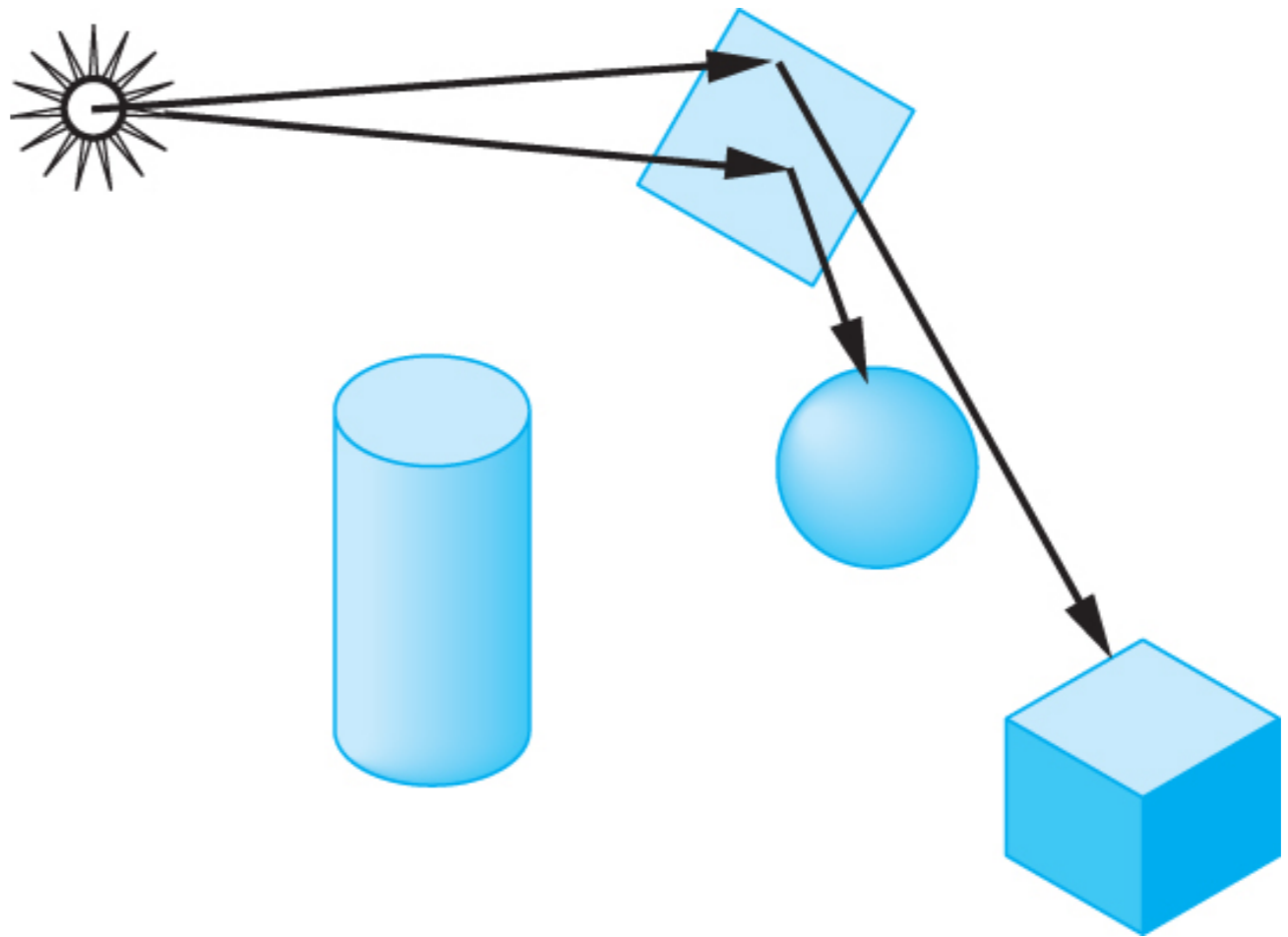


Wikimedia Commons

# Environment Mapping

Create the effect of a mirror with two-pass rendering

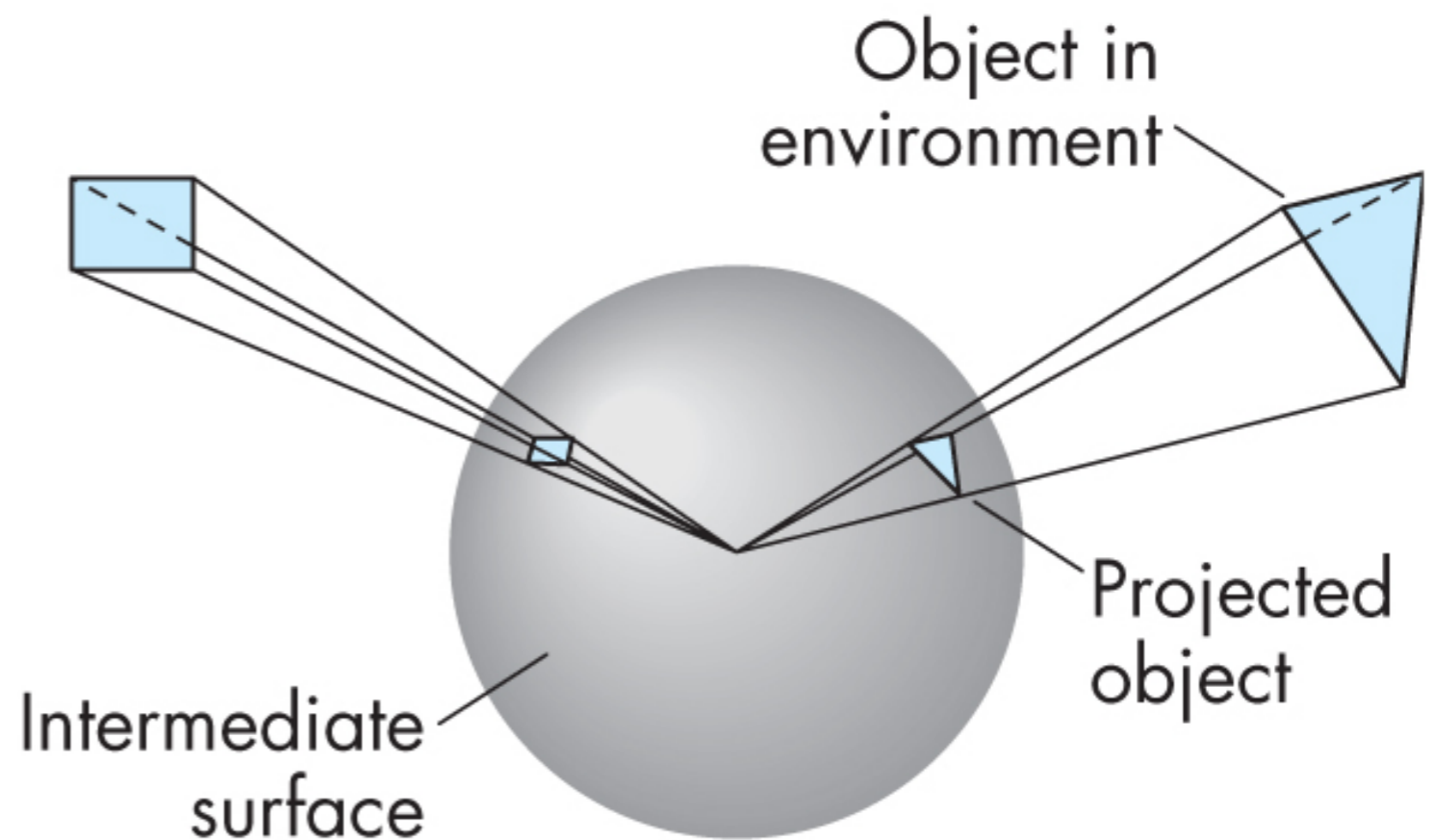
1. First pass: render the scene from the perspective of the mirror
2. Second pass: render from original pov; use the first image as a texture for the mirror





# Sphere Mapping

- Project objects in the environment onto sphere centered at eye
- unwrap and store as texture
- use reflection direction to lookup texture value

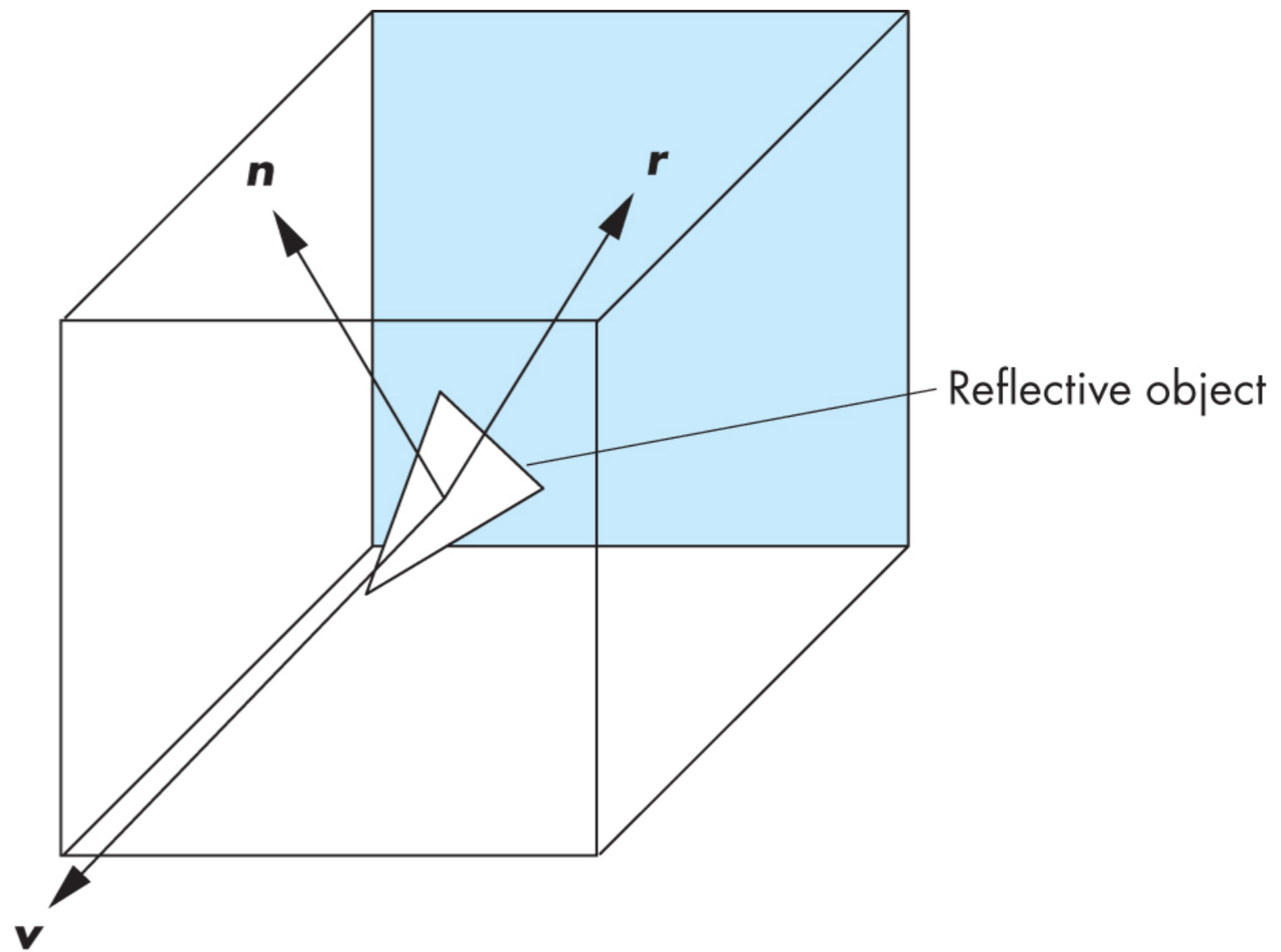


How is environment mapping different from ray tracing?

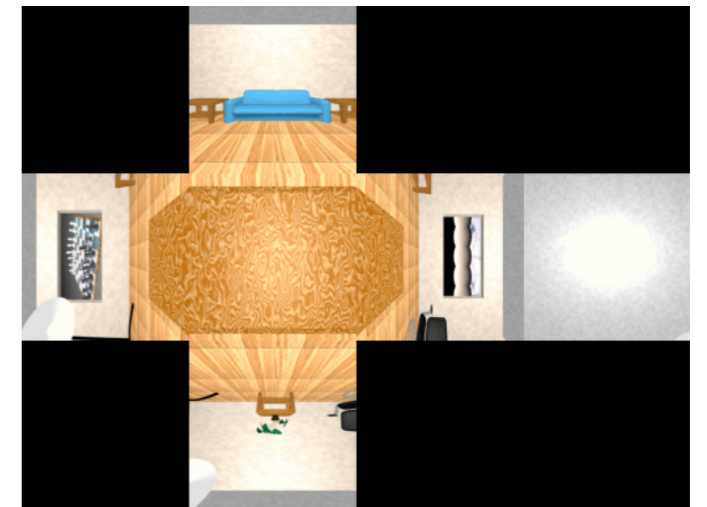
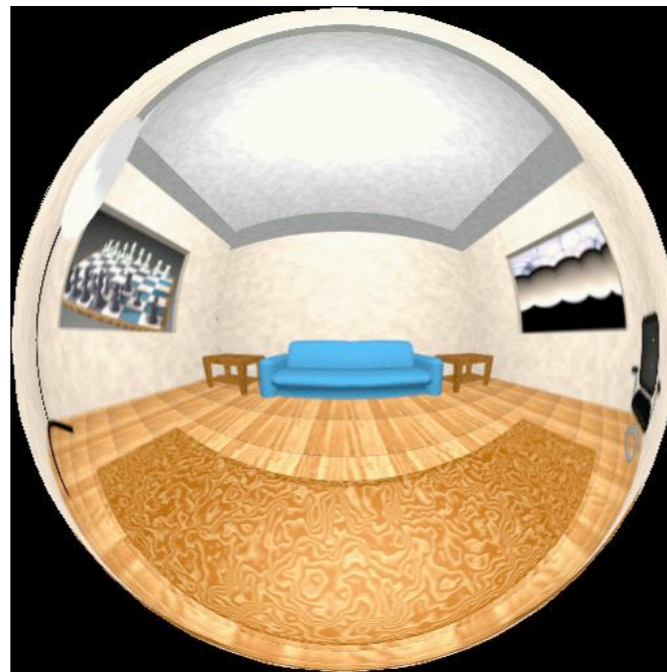
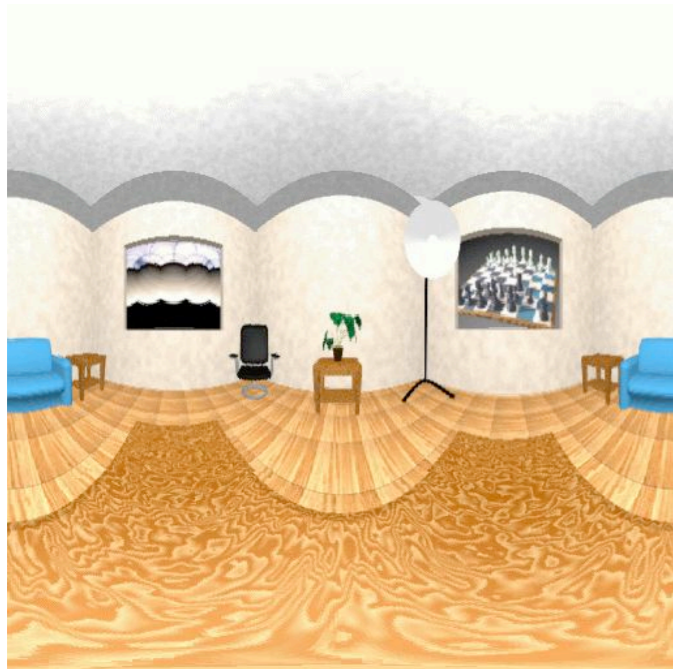
- typically only the direction of the reflection vector is used to look up the texture value- this doesn't reproduce the true intersection of the reflected ray the the object it hits
- Note: realism of environment map degrades as model is displaced from where the textures were generated

# Cube Mapping

- Compute six projections, one for each wall
- store as texture
- use reflection direction to lookup texture value



# Different environment maps



[www.reindelsoftware.com](http://www.reindelsoftware.com)



Blinn/Newell  
latitude mapping



OpenGL spherical  
mapping



Cube mapping

# Shadow Mapping

first pass from light's perspective

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

2. render scene from desired pov, and test pixel against light's z-buffer

