CMSC 435/634

Texture

Texture Mapping

- Def: mapping a function onto a surface; function can be:
  - 1, 2, or 3D
  - sampled (image) or mathematical function

Mapped Parameters

- Surface color (Catmull 74)
- Specular reflection (Blinn and Newell 76)
- Normal vector perturbation (Blinn 78)
- Specularity (Blinn 78)
- Transparency (Gardner 85)
- Diffuse Reflection (Miller and Hoffman 84)
- Shadows, displacements, etc (Cook 84)
- Local coord system (Kajiya 85)

Map Indices

- Surface parameters
- Ray direction
  - reflection/environment mapping
- Surface normal direction
  - diffuse reflection mapping
  - transparency/refraction mapping

Key Challenges

- Mapping function determination
- Resolution issues
- Texture design/capture

Mapping Functions

- Standard projecting functions
  - planar
  - cylindrical
  - spherical
- Mechanism
  - Two-stage mapping
  - Reverse projection
- Arbitrary

Two-stage Mapping

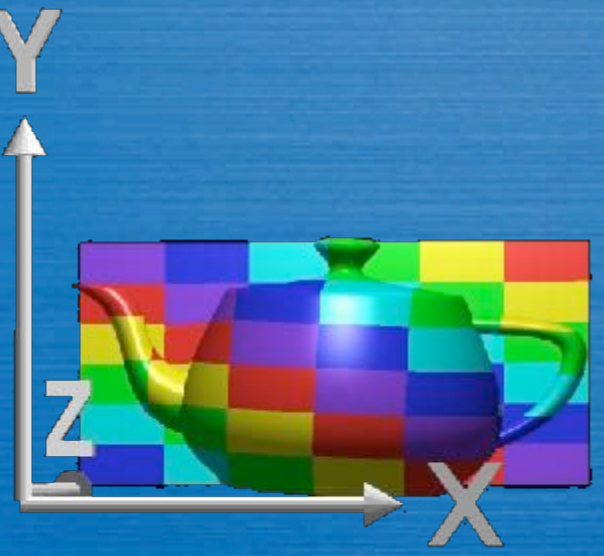
- S-mapping
  - map to simple 3D shape
  - intermediate surfs: plane, cylinder, cube, sphere
- O-mapping
  - map 3D texture onto surface
  - map entities: reflected view ray, surface normal, line through centroid, intermediate surface normal

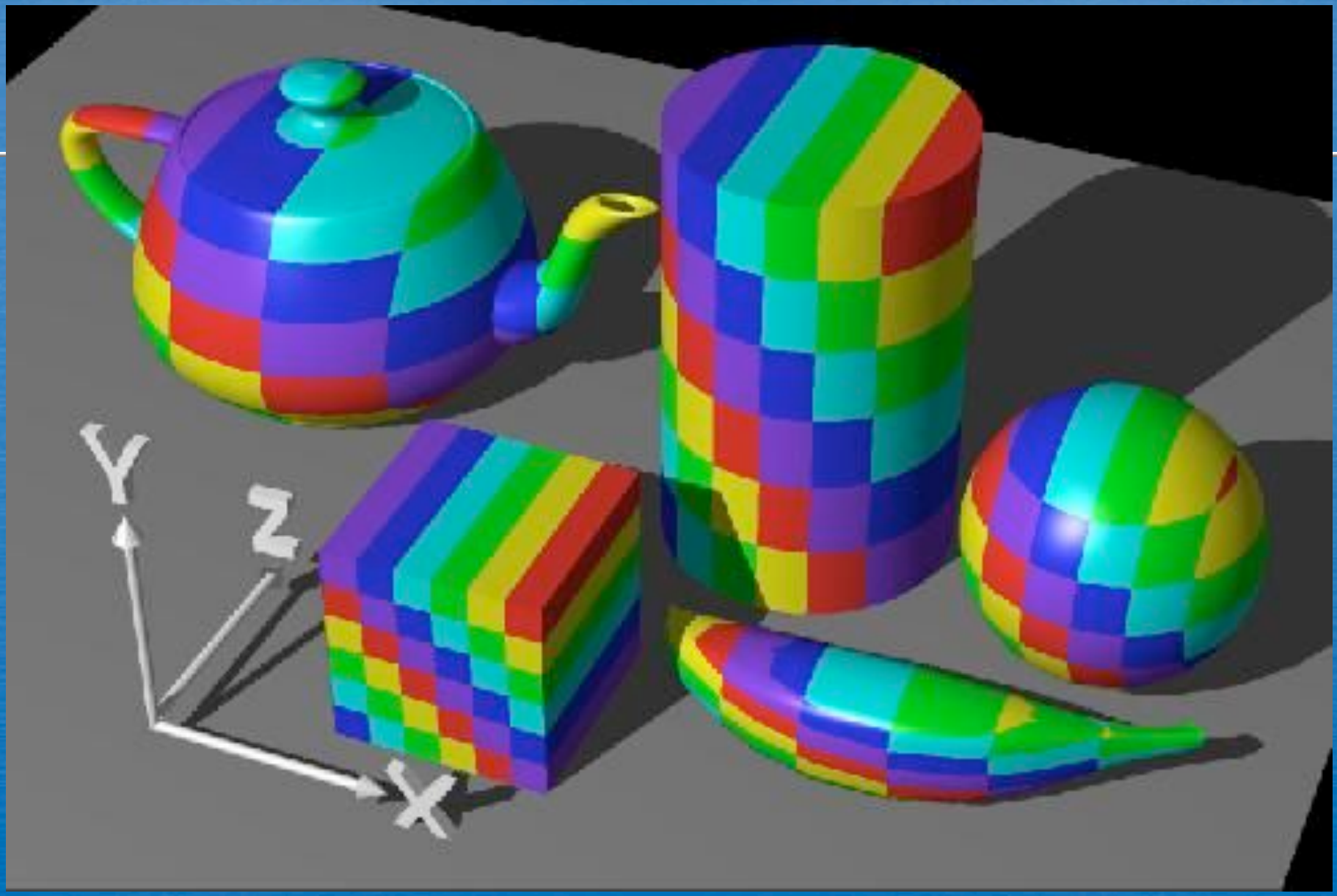
Planar Mapping

- For xy aligned plane

$$(u, v) = \left(\frac{x - x_1}{x_r - x_1}, \frac{y - y_1}{y_r - y_1} \right)$$

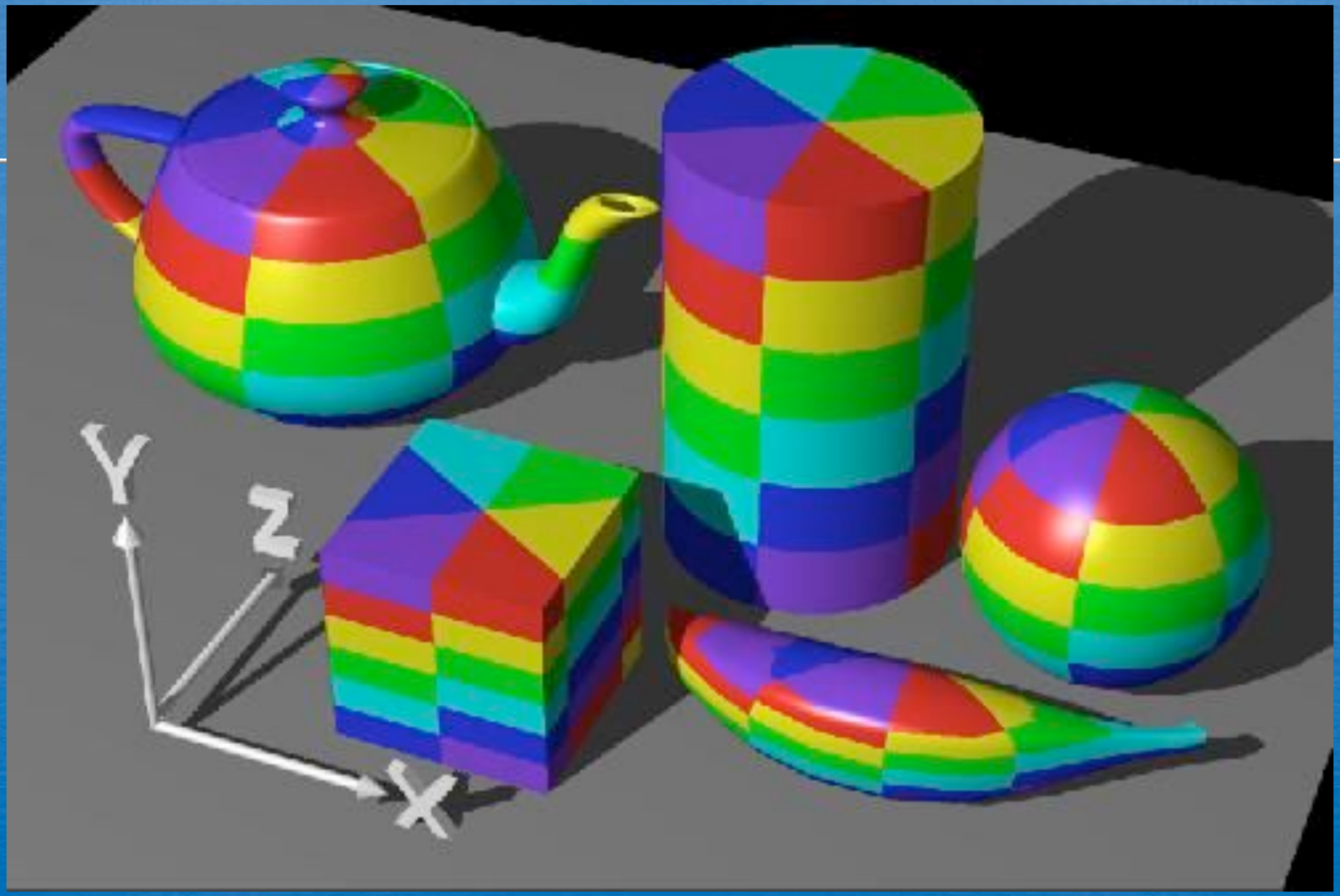
- Reverse projection




Y
Z
X

Cylindrical Mapping

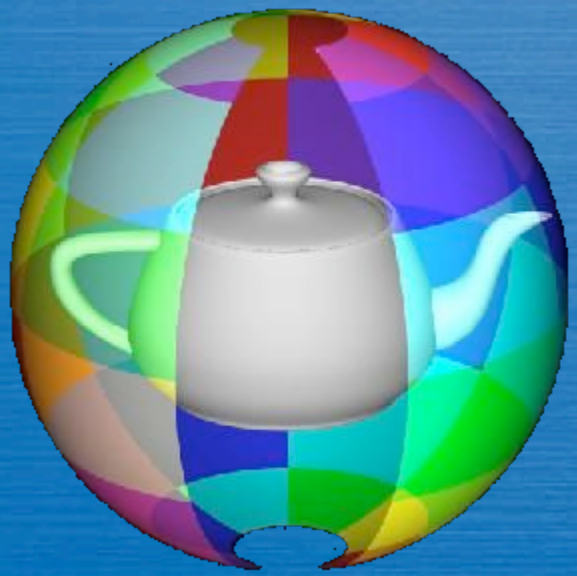
- For cylinder with point
  $(r \cos\theta, r \sin\theta, h\ z)$
- Texture coords
  $(u,v) = (\theta/2\pi, z)$

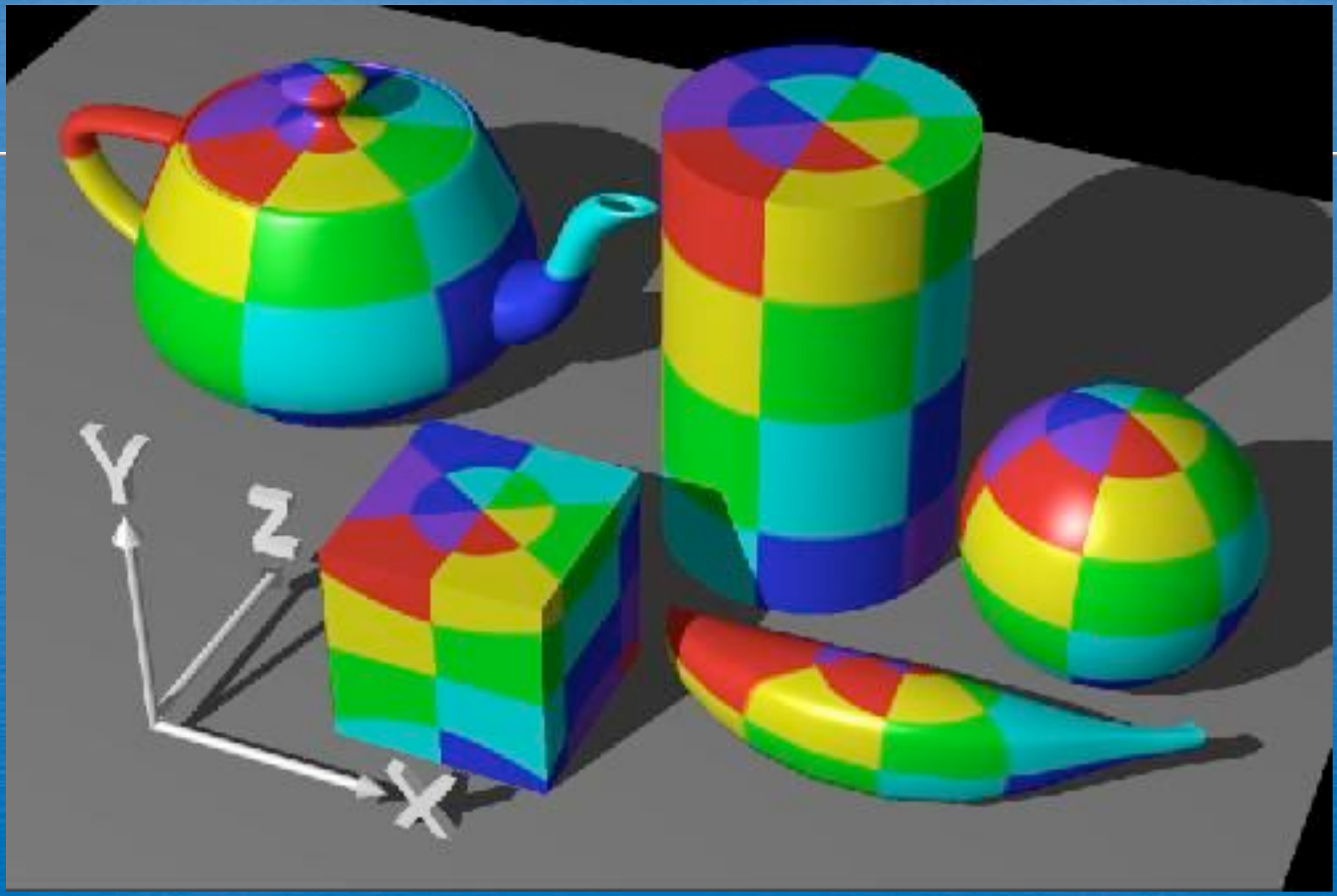

Y
Z
X

Spherical Mapping

- For sphere with point
  (r cosθ sinϕ, r sinθ sinϕ, r cosϕ)
- Texture coords

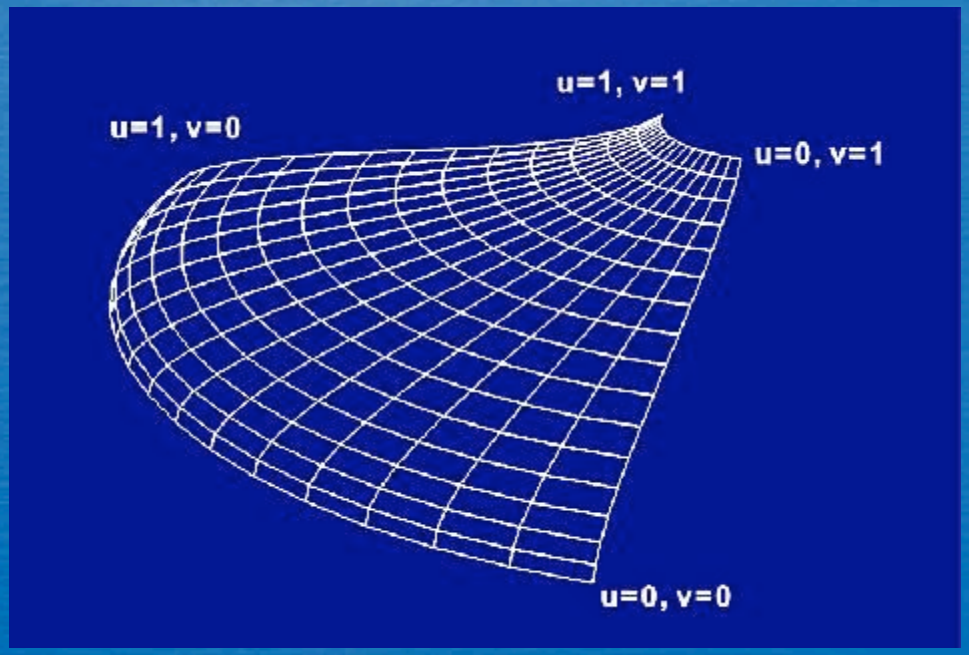
$$(u, v) = \left(\frac{\theta}{\pi/2}, \frac{\pi/2 - \phi}{\pi/4} \right)$$




Y
Z
X

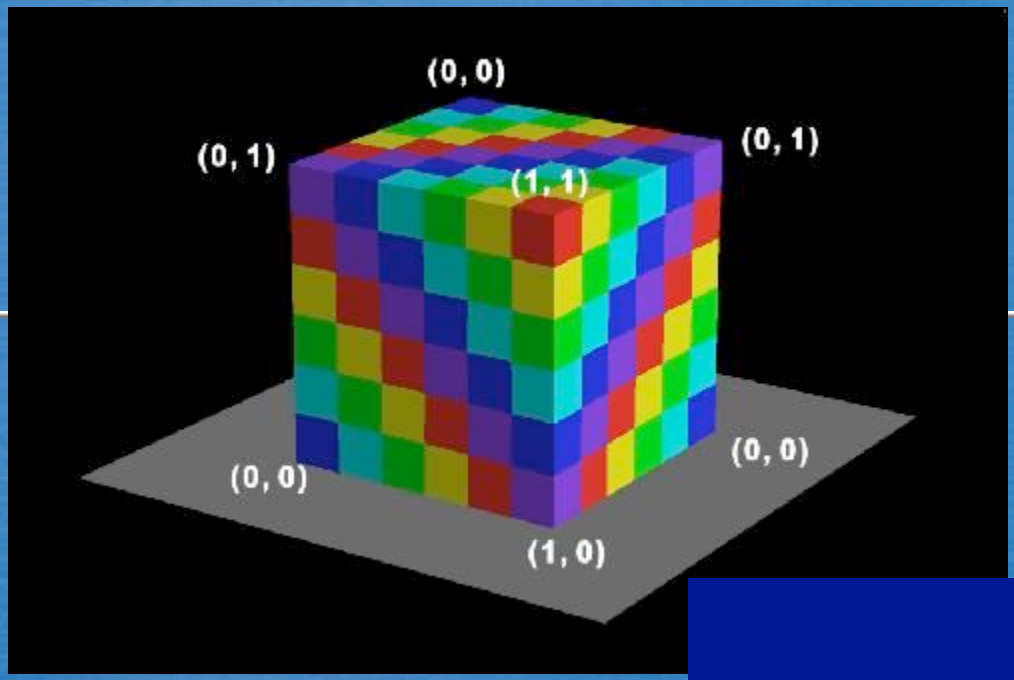
Mapping onto Parametric Patches

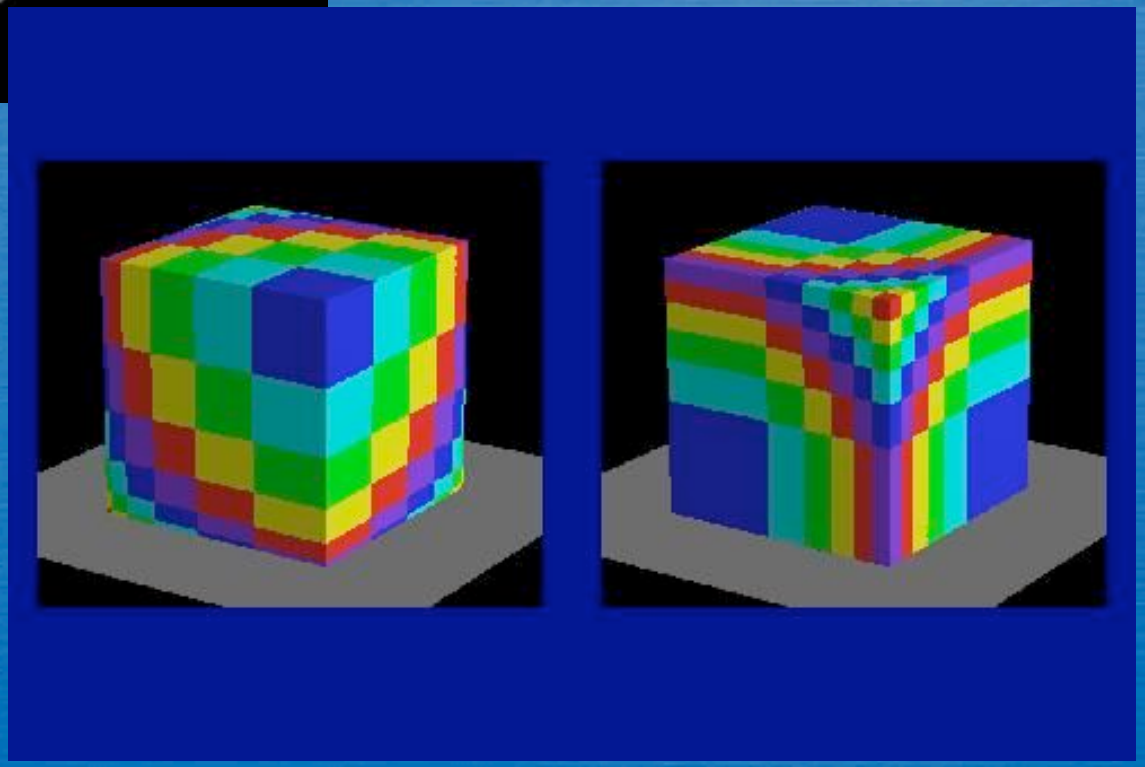
- Use scaled surface u,v parameters for texture u,v



Mapping onto Polygons

- Like parametric surfaces, but use explicit vertex texture coordinates
- Screen-space Interpolation
  - Interpolate u,v
  - Nonlinearity and errors from lack of rotational invariance
  - use small polygons to minimize artifacts
- Correct solution: per-pixel projection
  - Interpolate (u/w, v/w, 1/w); divide to get pixel (u,v)


(0, 0)
(0, 1)
(0, 1)
(1, 1)
(0, 0)
(0, 0)
(1, 0)



Bump Mapping

- Perturb surface normals to simulate shape variations

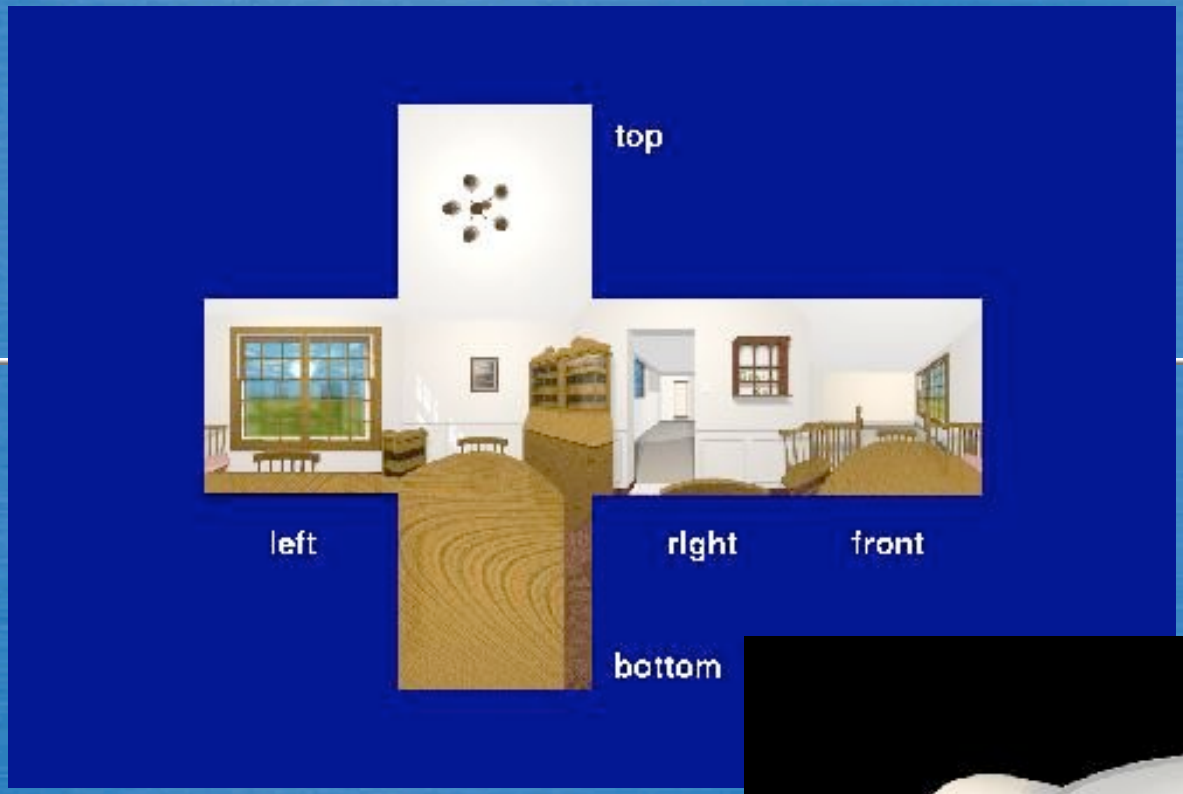



Bump mapping

Displacement Mapping

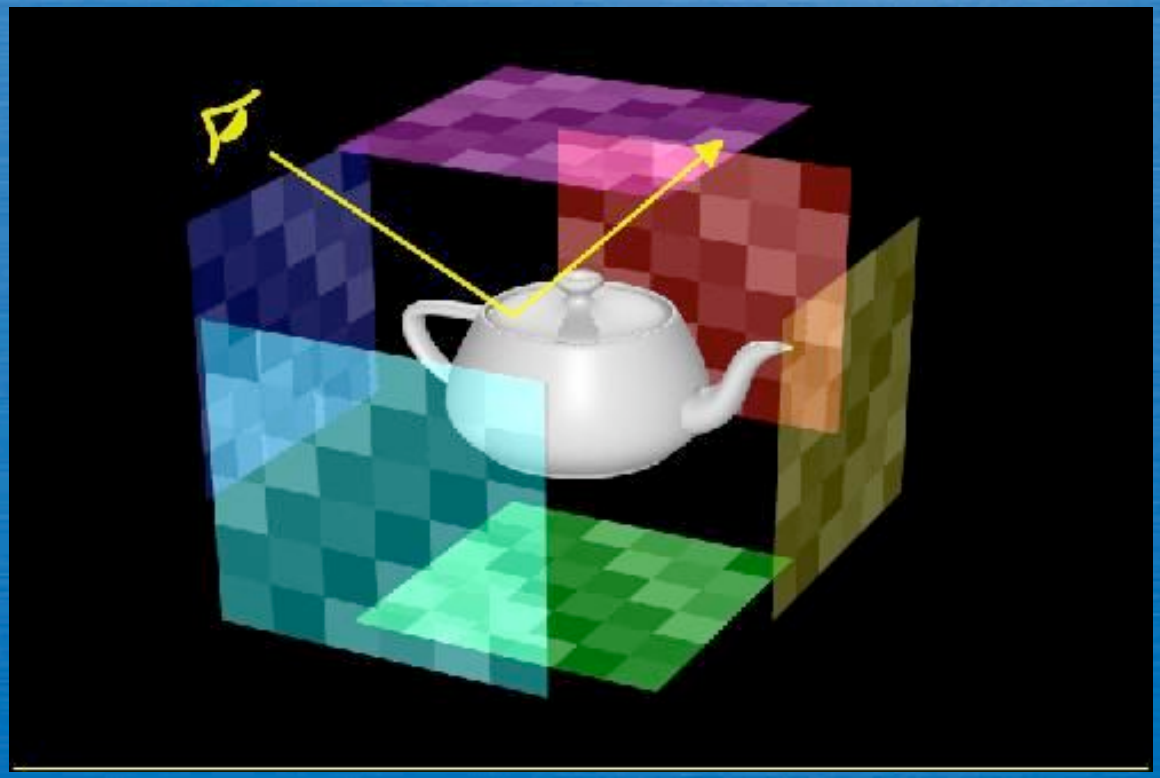
Reflection Mapping

- Look up reflections on an object from a map simulating surrounding environment


top
left
right
front
bottom

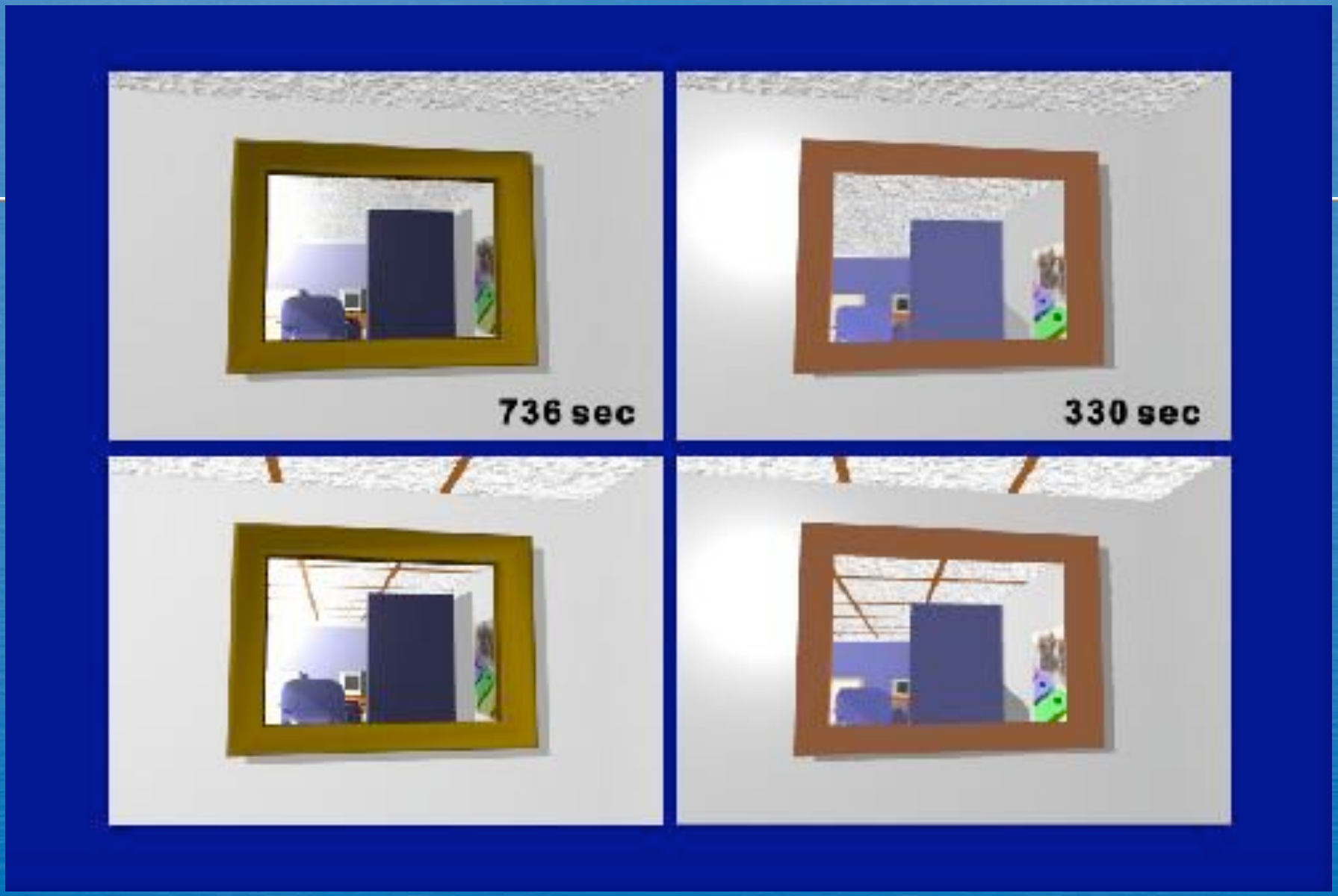
Environment Mapping

- Surround scene with maps simulating surrounding detail



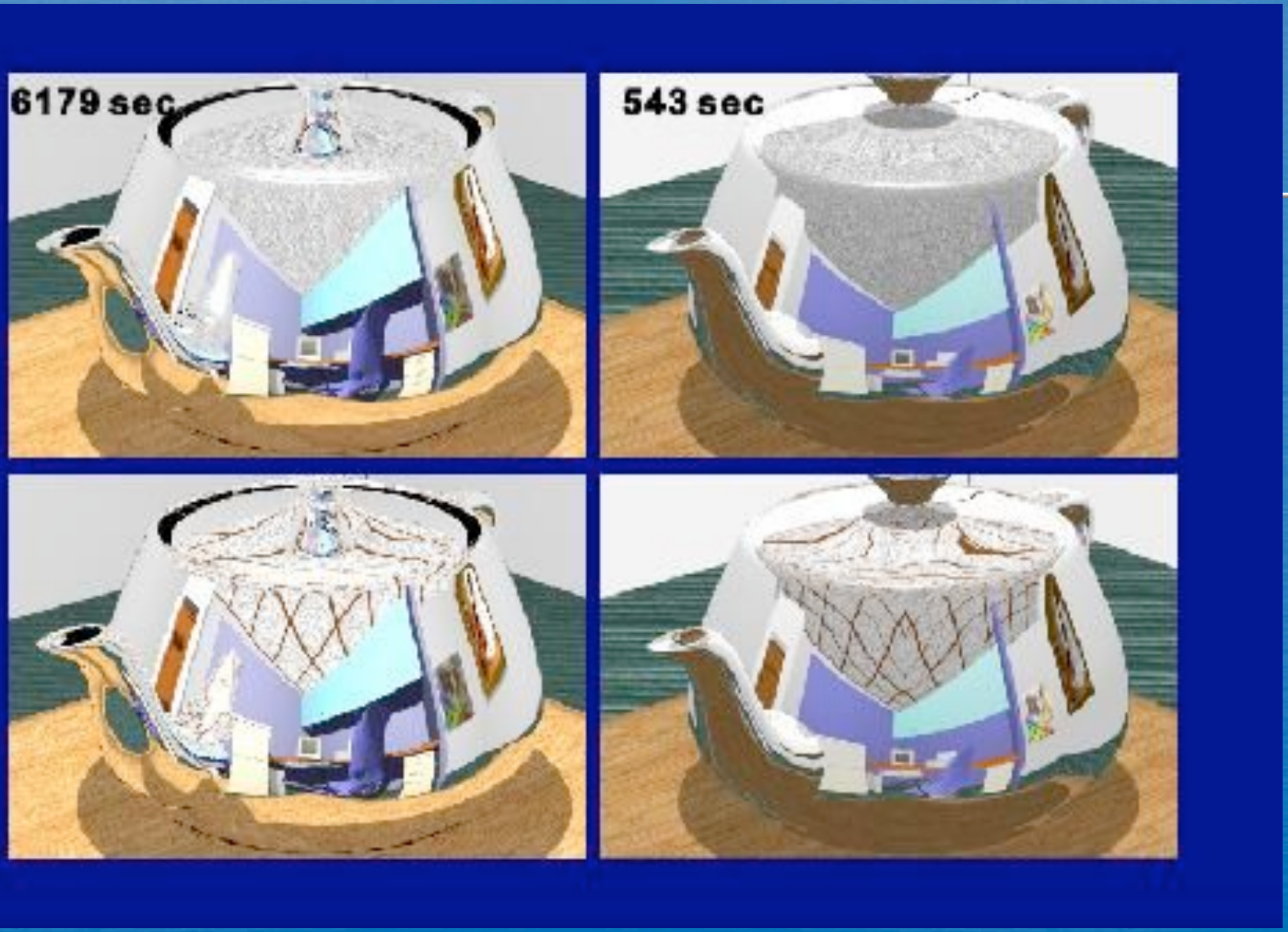
Ray tracing

Environment Mapping



Ray tracing

Environment mapping

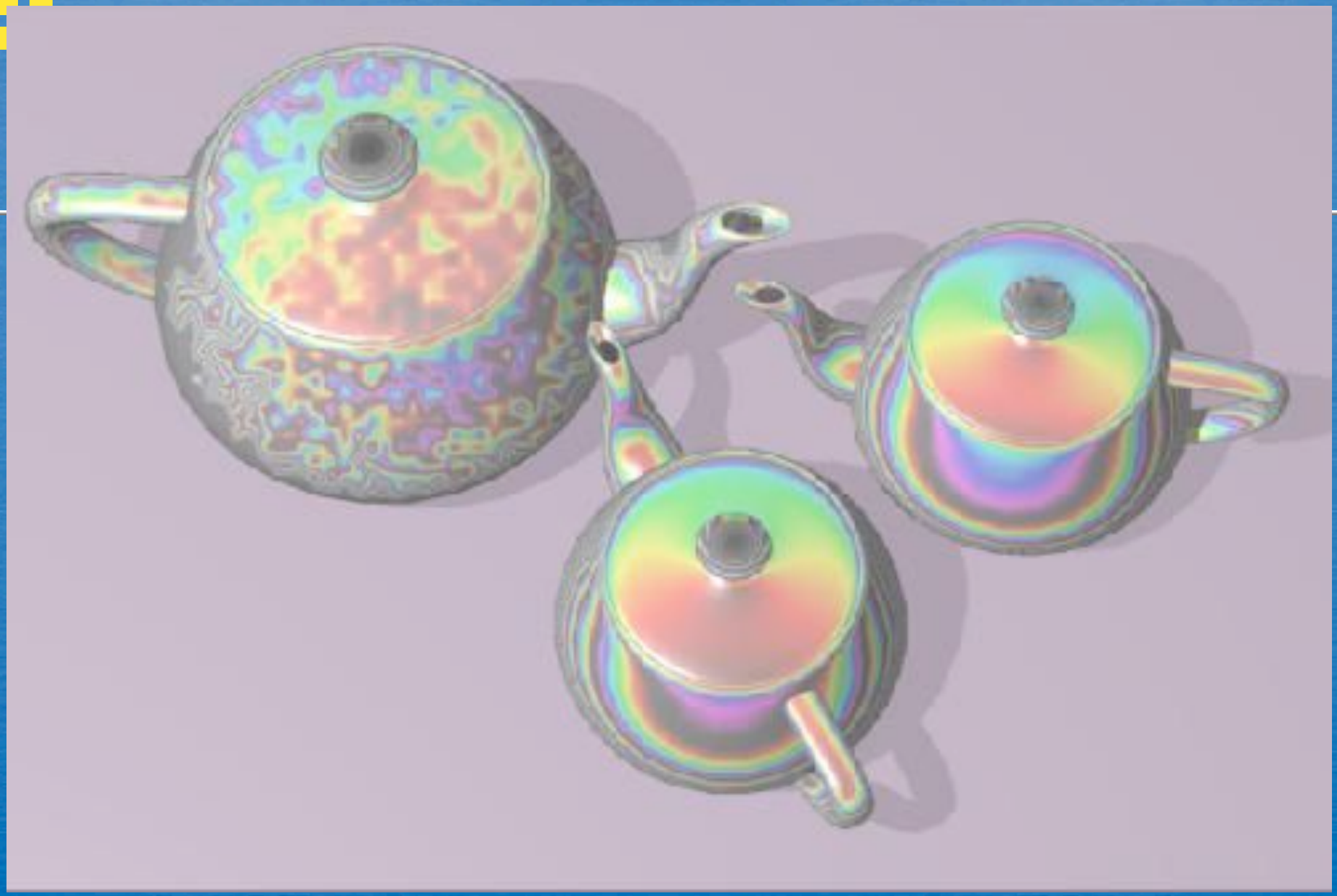


Ray tracing

Environment Mapping

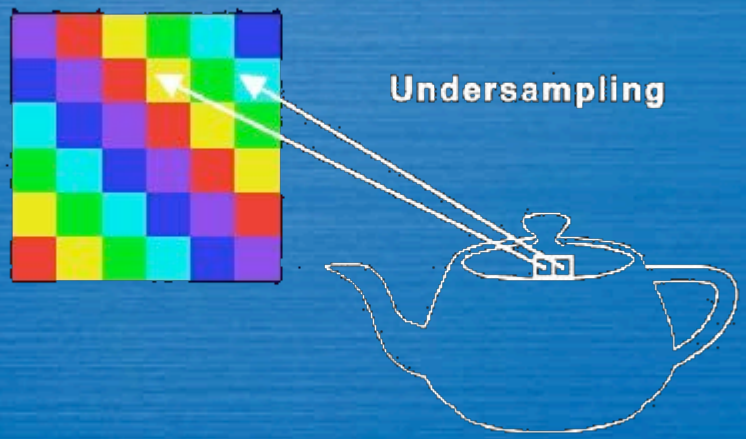
Refraction Mapping

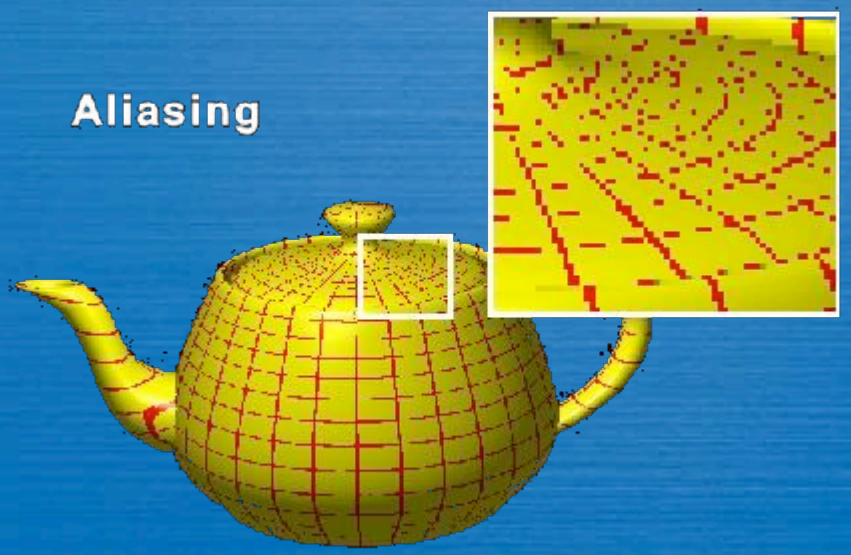
- Perturb refraction rays through transparent surface by disruption of surface normal



Texture Aliasing

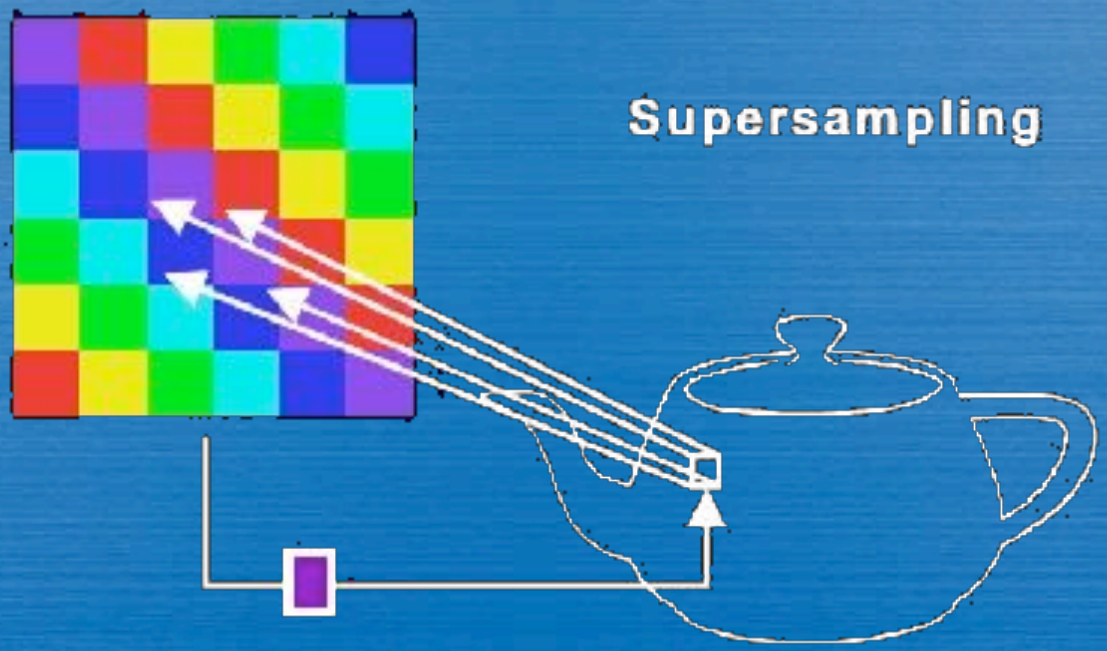
- Undersampling of texture map leads to texture aliasing
- Oversampling can show limited texture resolution





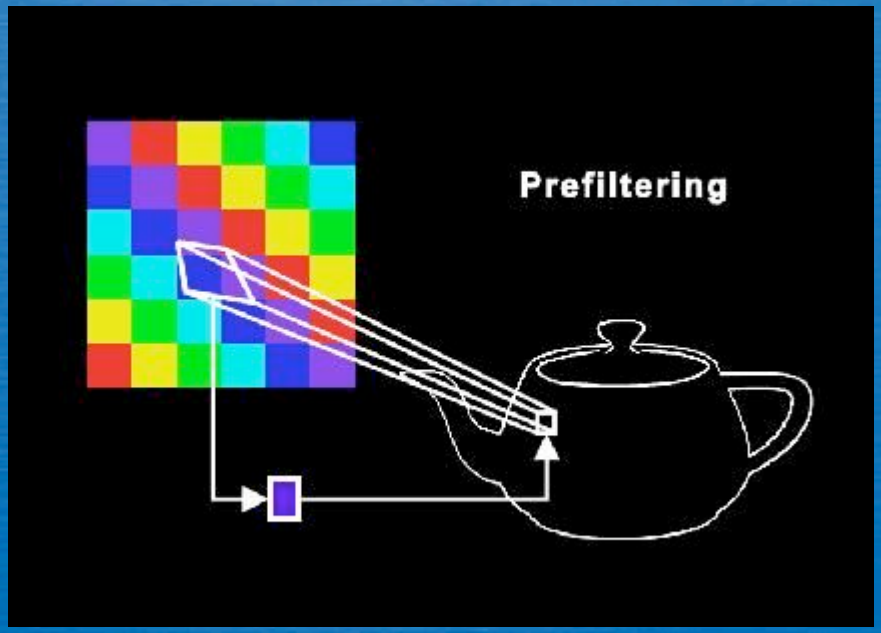
Supersampling

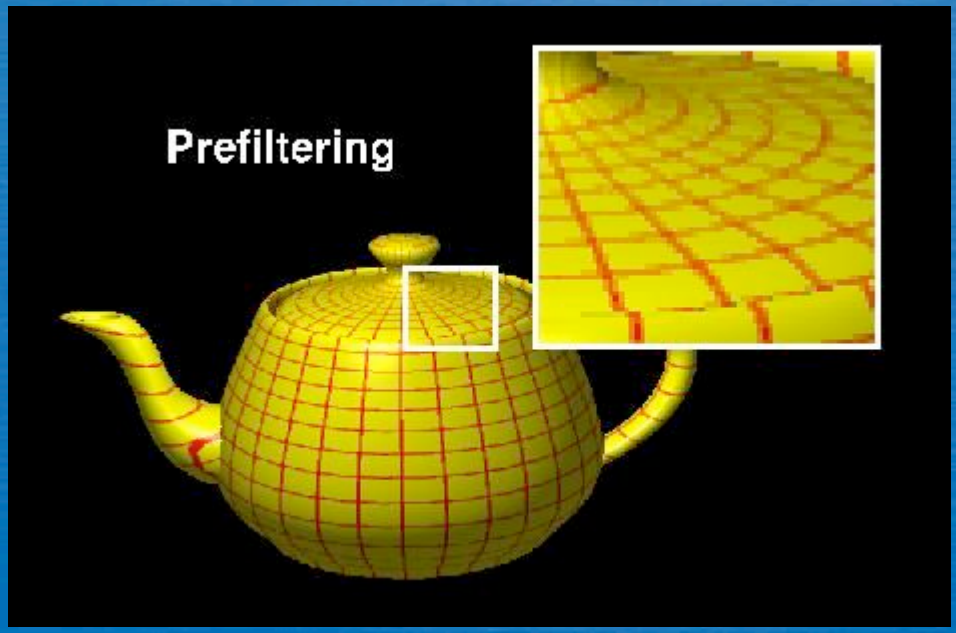
- Sample texture multiple times per pixel and reconstruct



Filtering

- Basic method (Catmull 78)
  - Project pixel pgon onto texture map
  - Average color over projected area



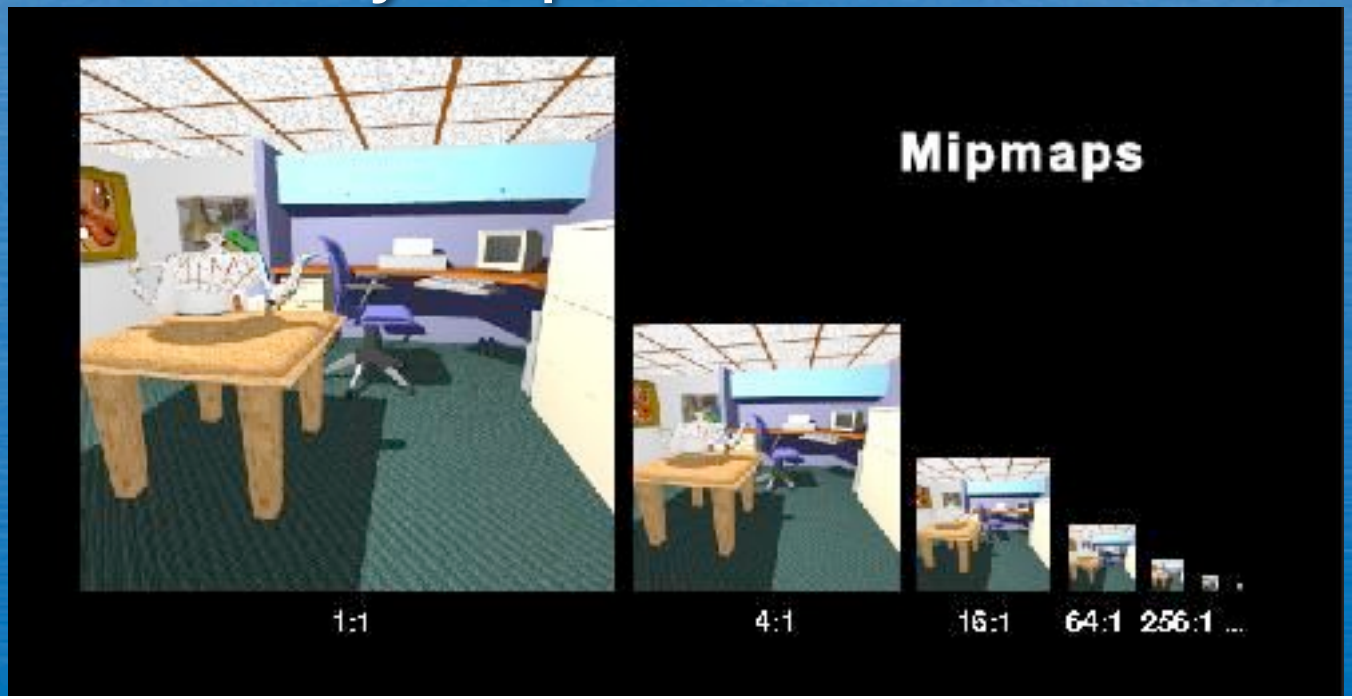


Filtering Types

- Direct Convolution
  - average multiple samples from texture (usually selected in texture space)
- Prefiltering
  - construct multi-resolution copies of texture
- Fourier filtering
  - low pass filter texture in frequency space

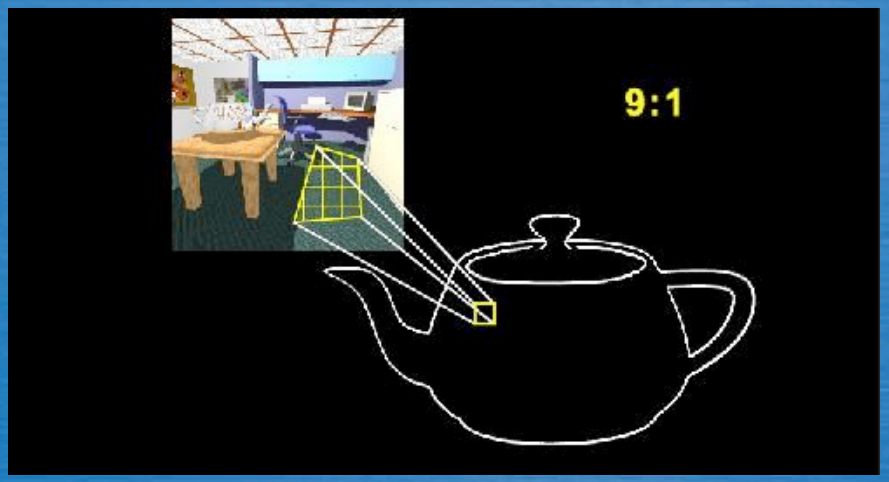
Mipmappng

- Precalculate filtered maps at a range of resolutions (Williams 83)
- Higher memory requirements

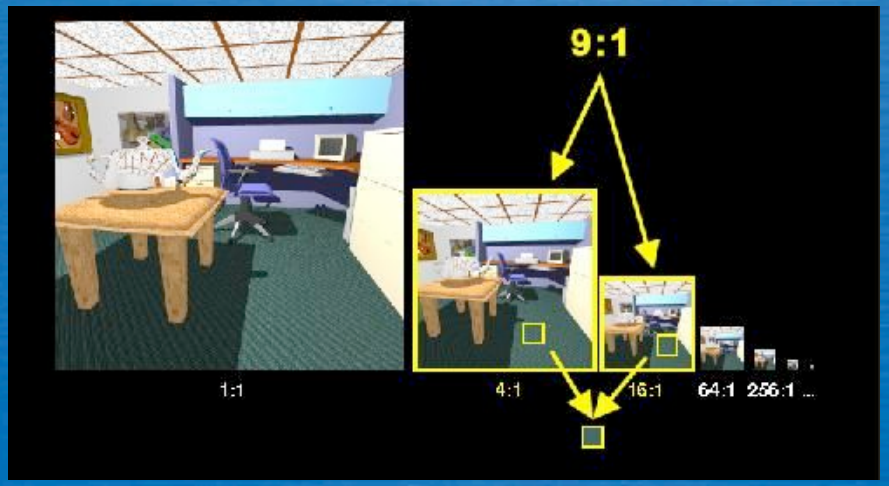


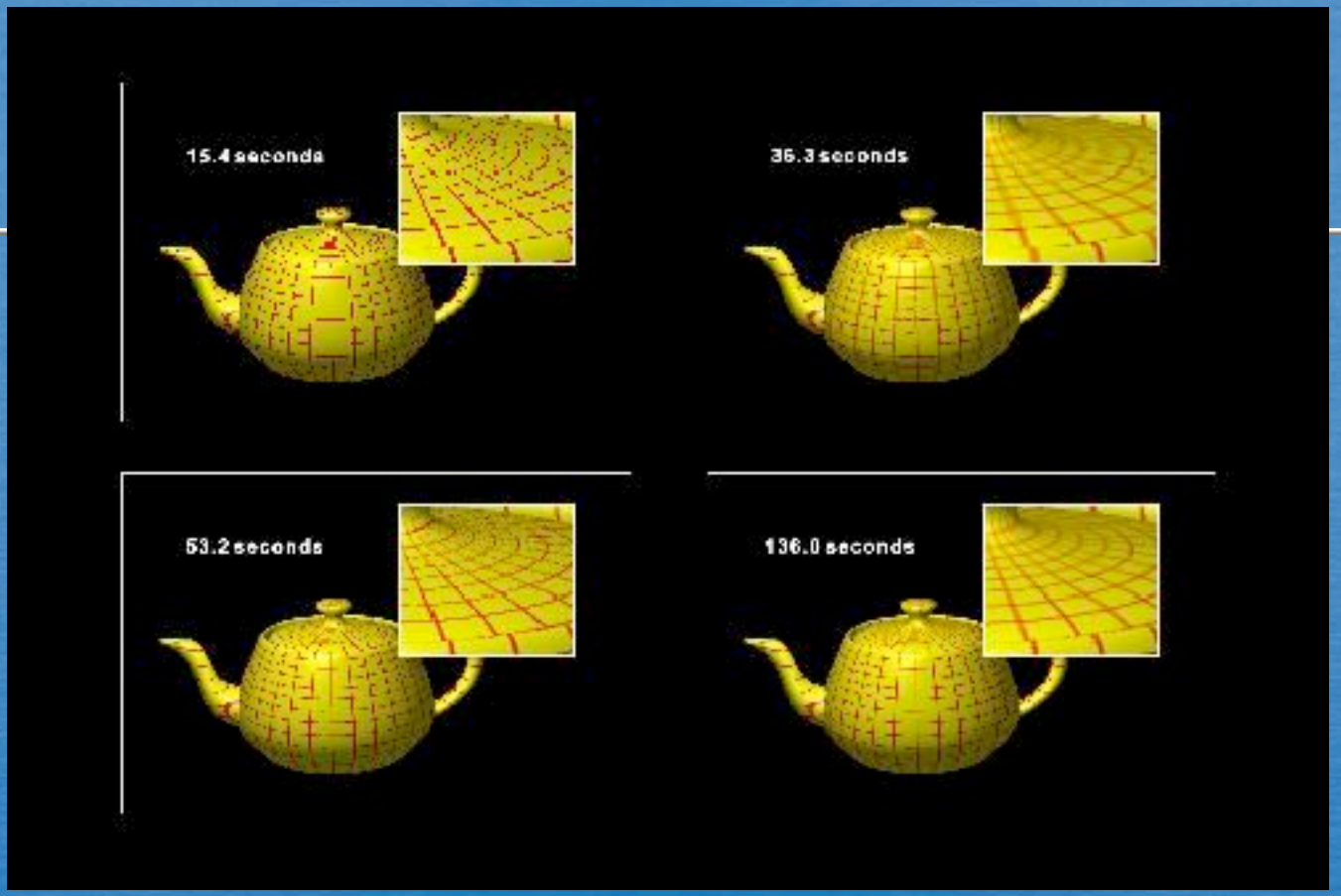
Mipmapping Process

- Compute pixel area in mipmap



- Average from two closest maps





Anti-aliasing: none, mipmapped, supersampled, supersampling and mipmapping