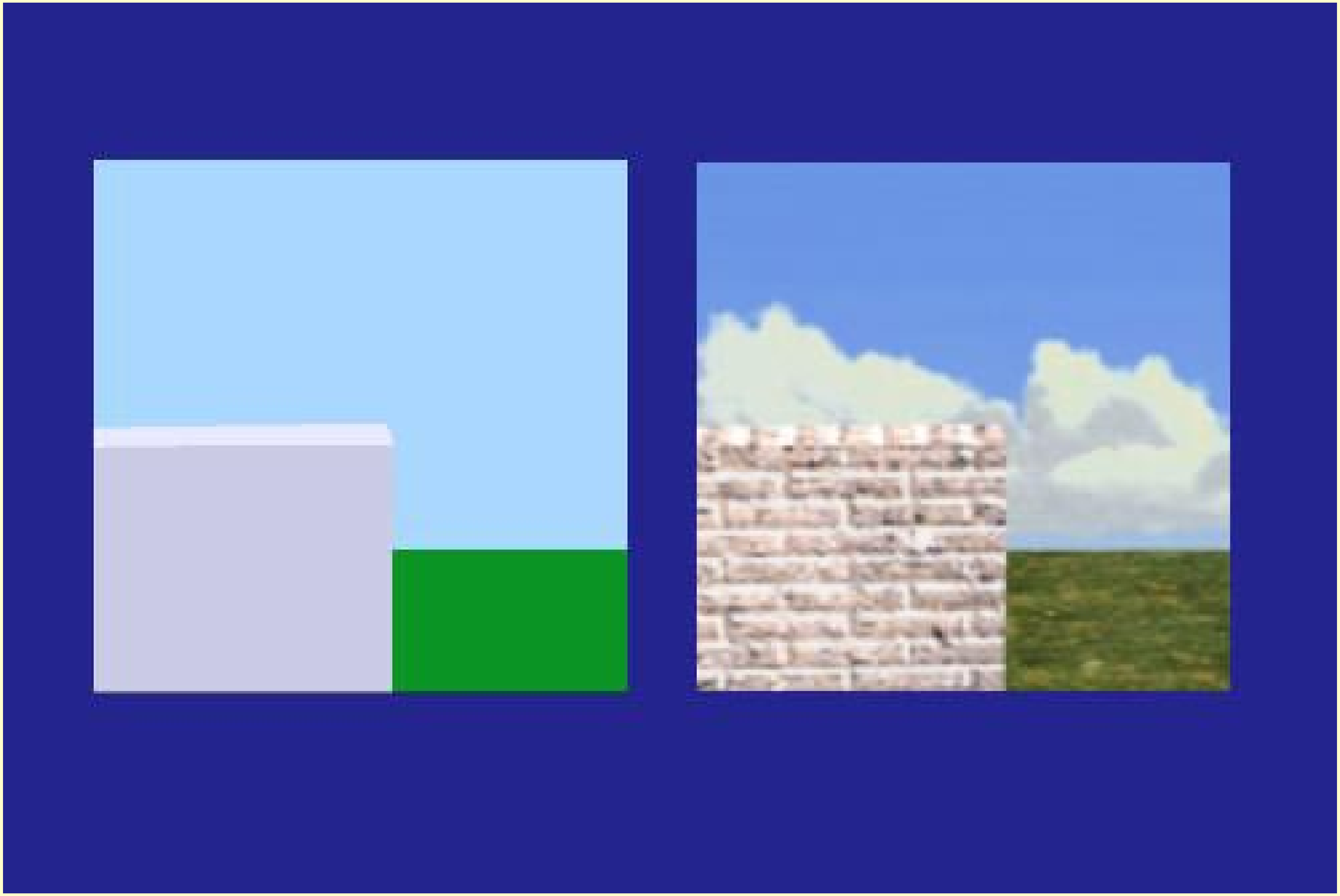
CMSC 435/634

Texture

Based on SIGGRAPH 97 education slide set
Rosalee Wolfe, DePaul University

Texturing Adds Richness



Texture Mapping

Mapping a function onto a surface

Function can be

- 1D, 2D or 3D
- Sampled (i.e. an image) or a 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

- Position
- Surface parameterization
- Manually defined *texture coordinates*

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

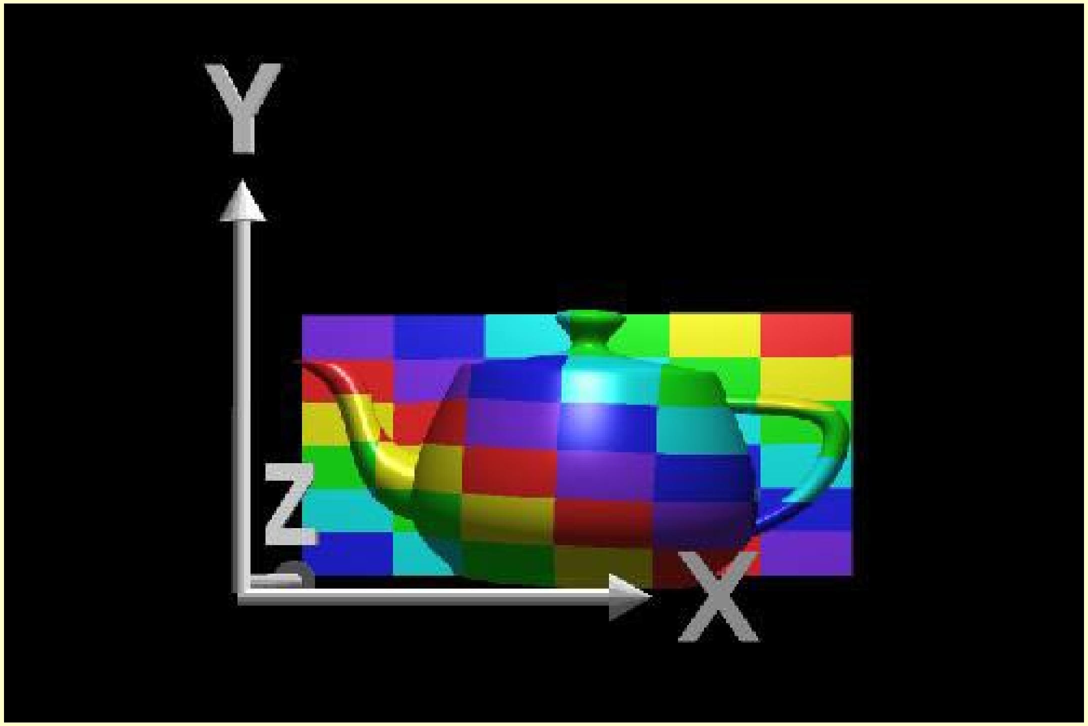
- Two-stage mapping
- Reverse projection

Planar Mapping

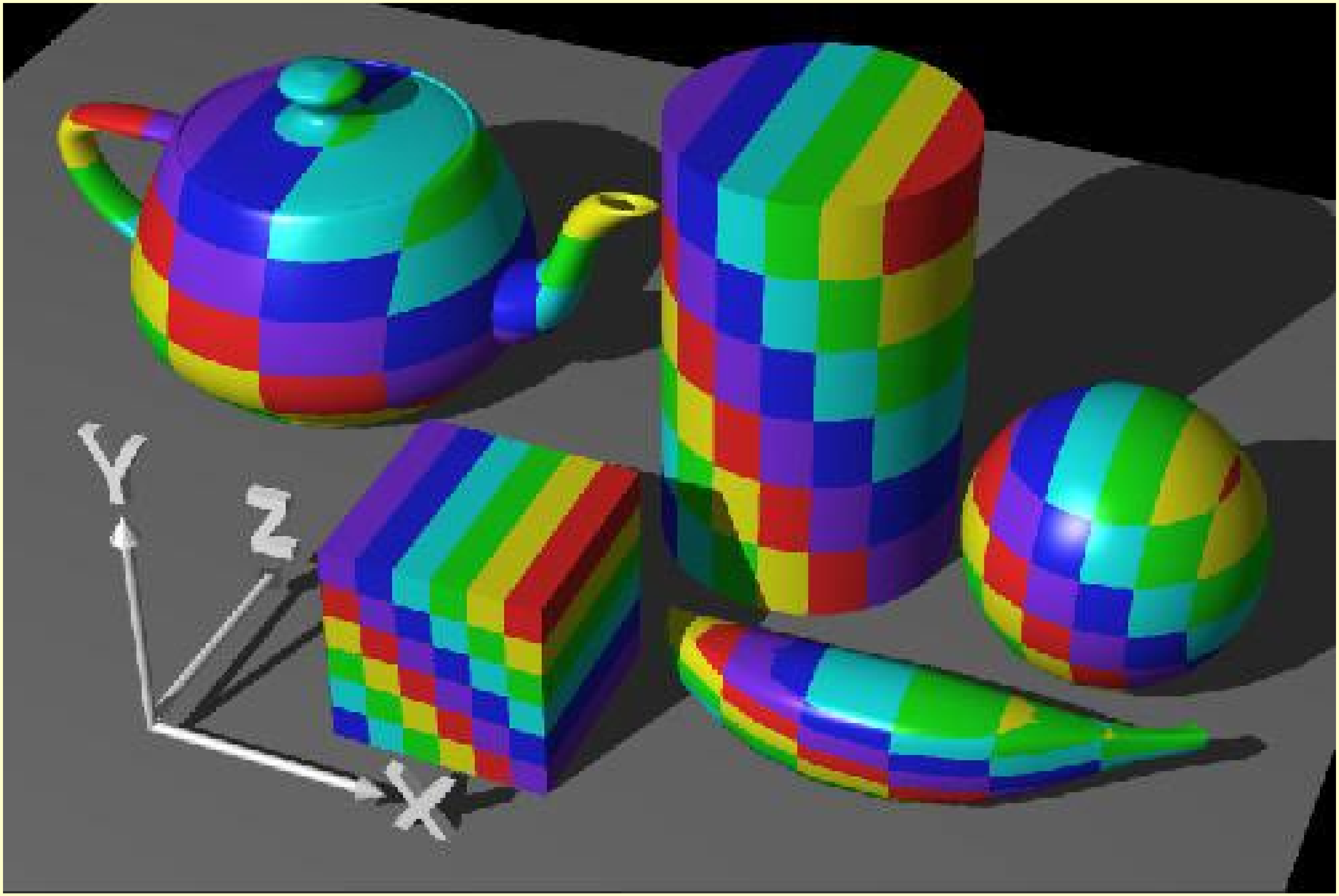
Parallel Projection of Texture

Reverse Map Position to Texture Coordinate

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



Planar Mapping

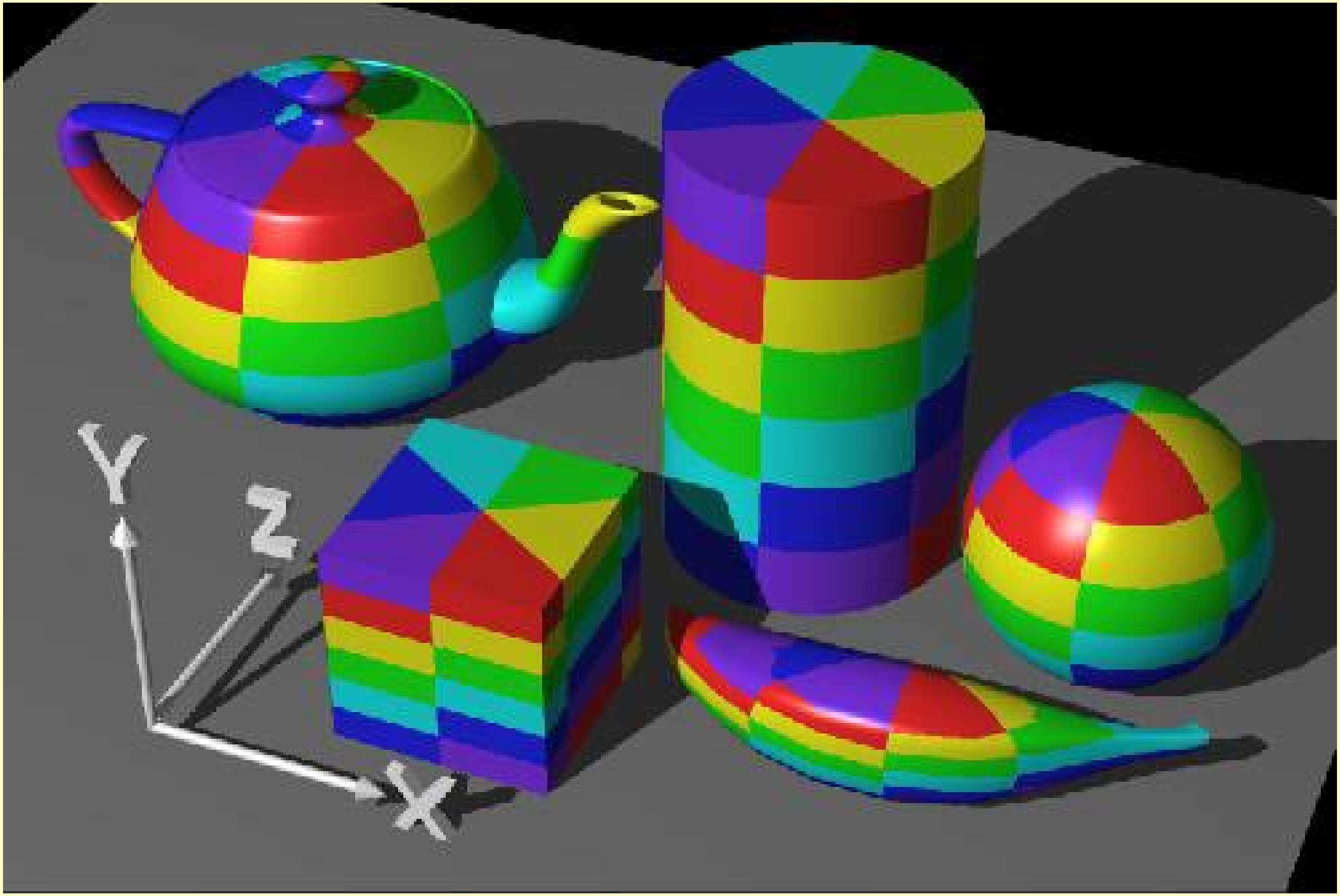


Cylindrical Mapping

For cylinder with point $(r \sin\theta, r \cos\theta, h\ z)$

Texture coordinate $(u, v) = (\theta/2\pi, z)$

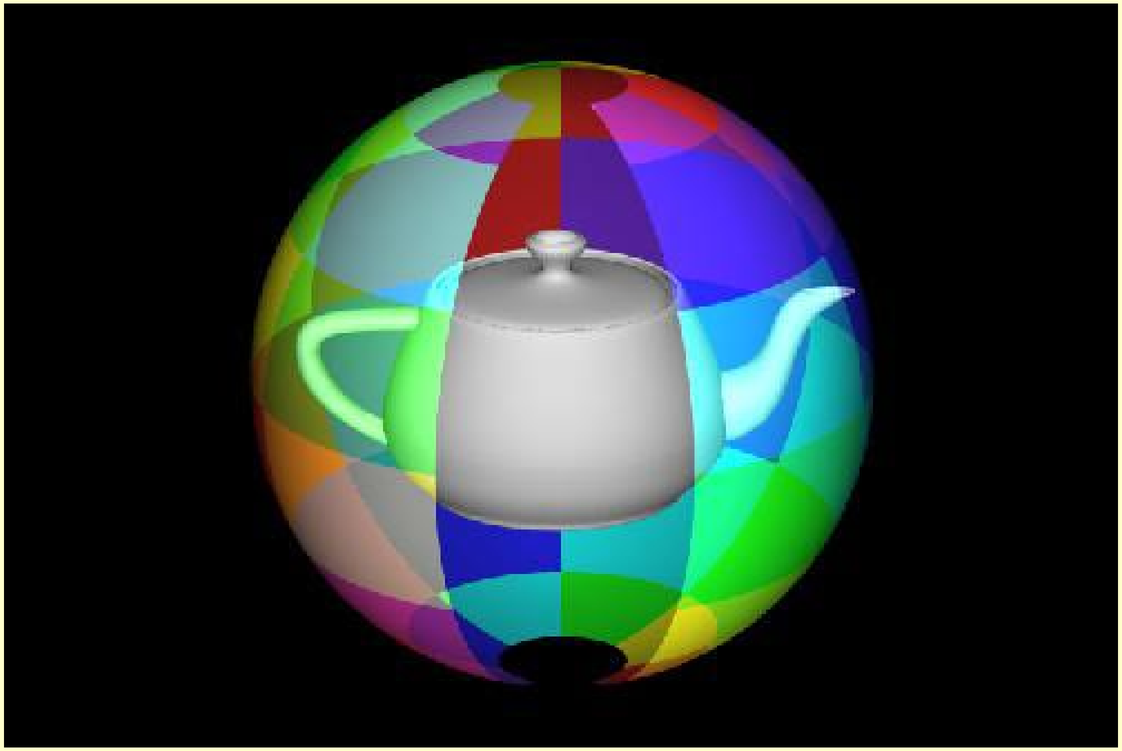
Cylindrical Mapping



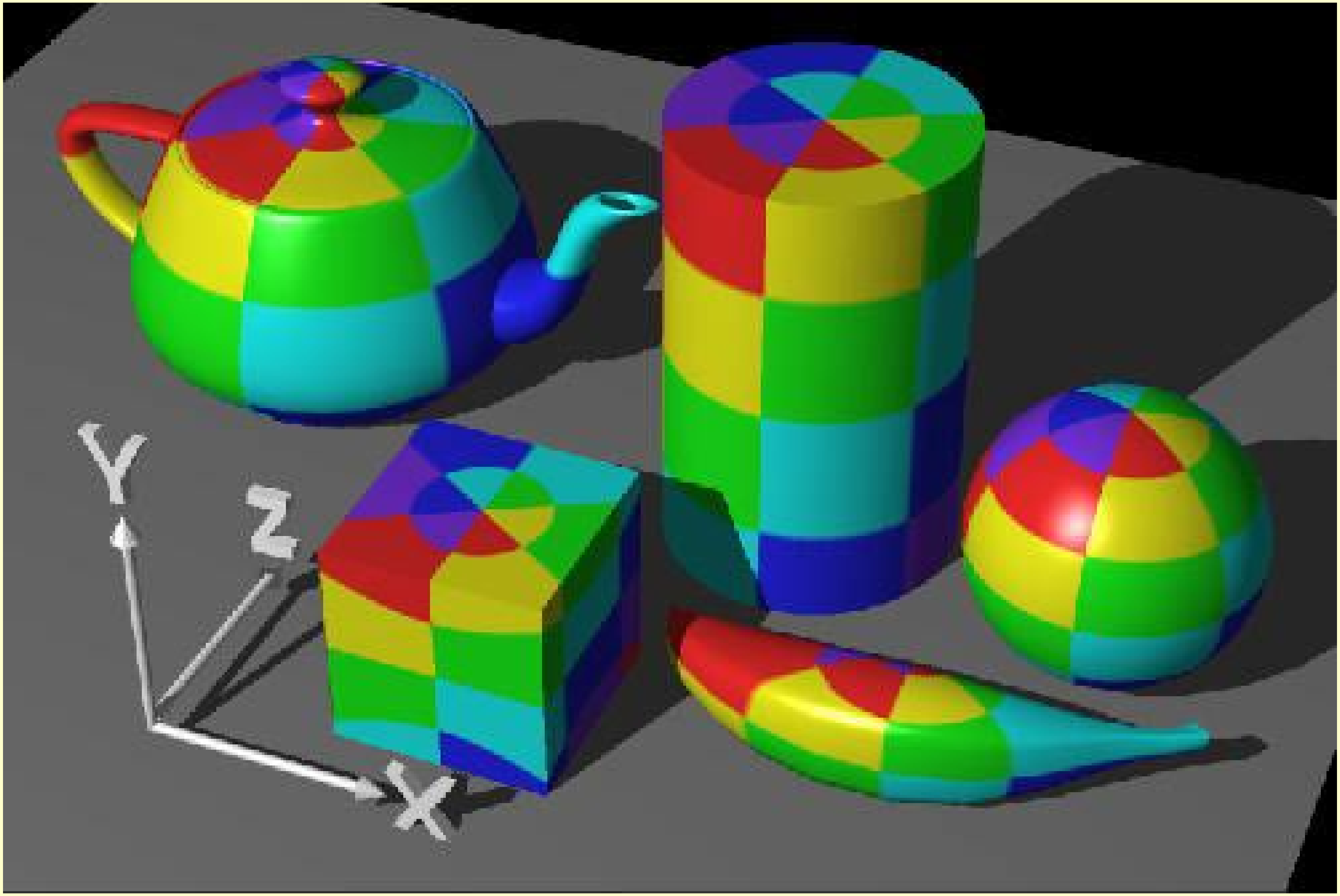
Spherical Mapping

Sphere point $(r\cos\theta\sin\phi, r\sin\theta\sin\phi, r\cos\phi)$

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



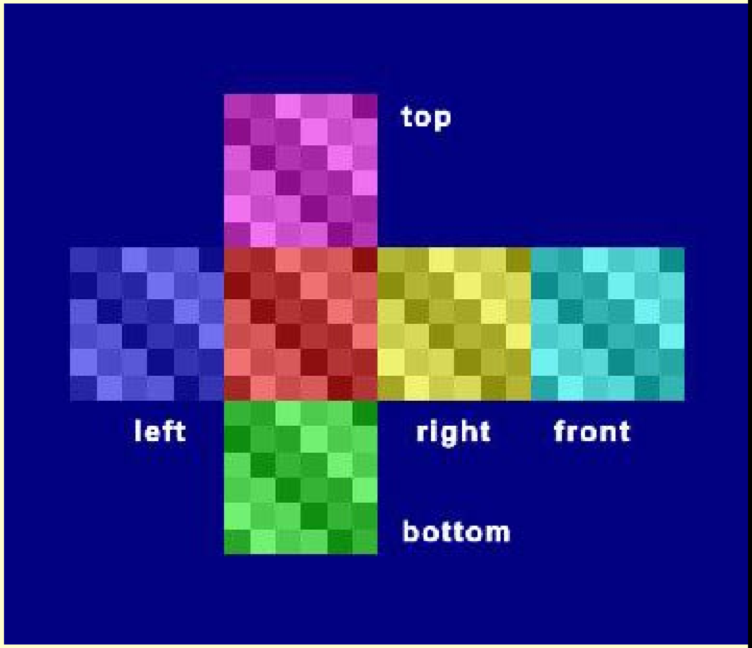
Spherical Mapping

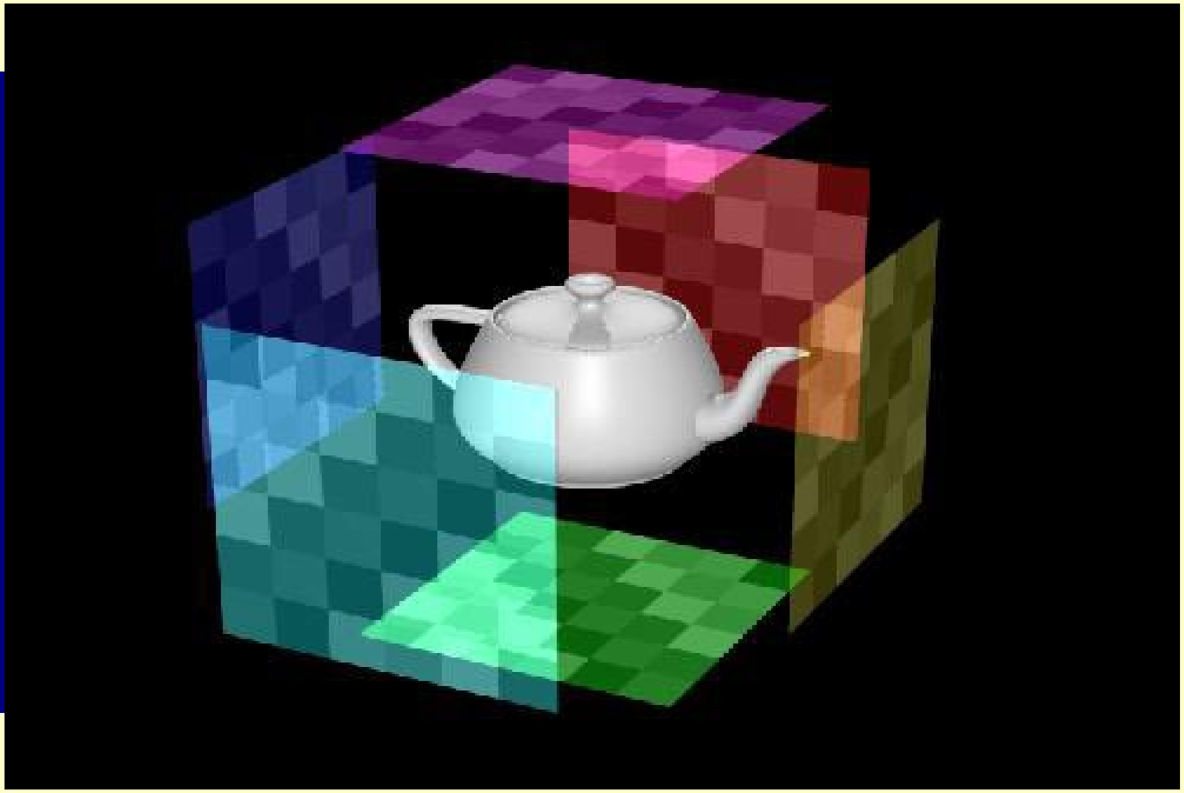


Environment Mapping

Map *vector direction* to texture coordinates

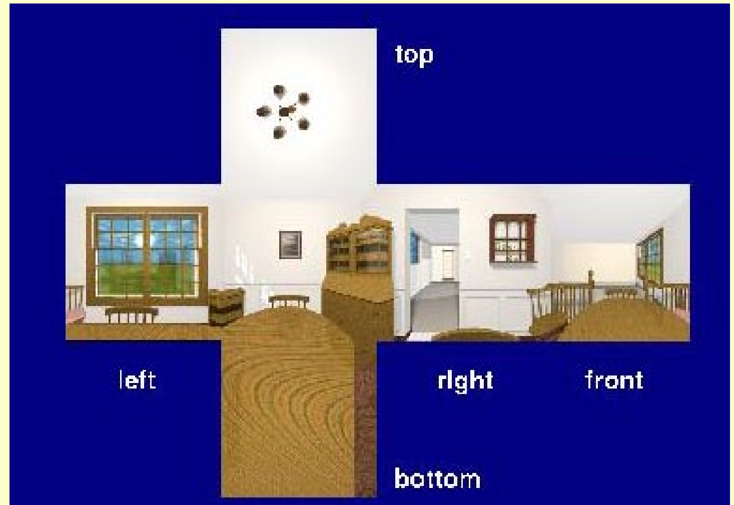
- Any mapping of *vector* to *texture* will work
- *Cube map* uses texture on six faces of cube

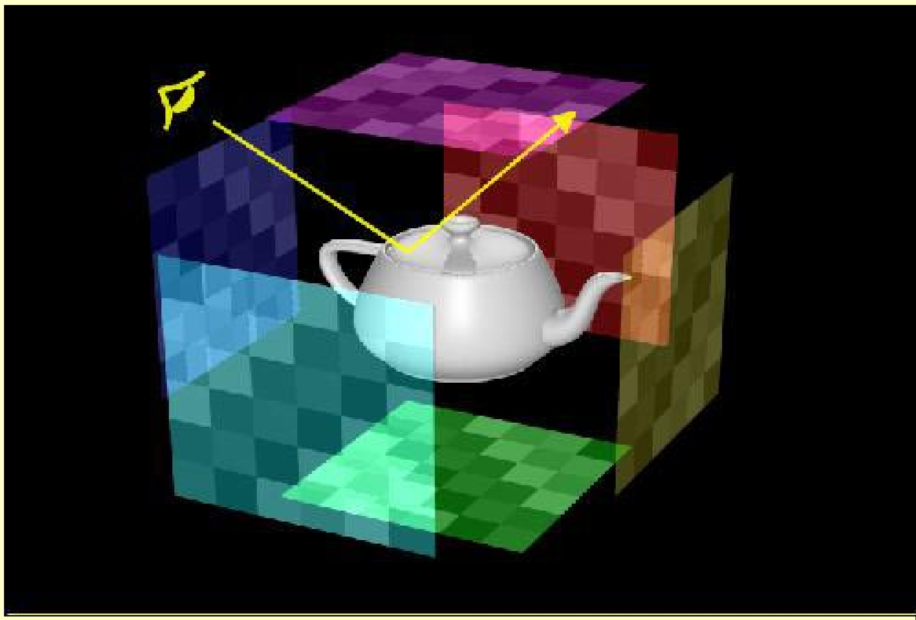




Reflection Mapping

Map based on *reflection direction*



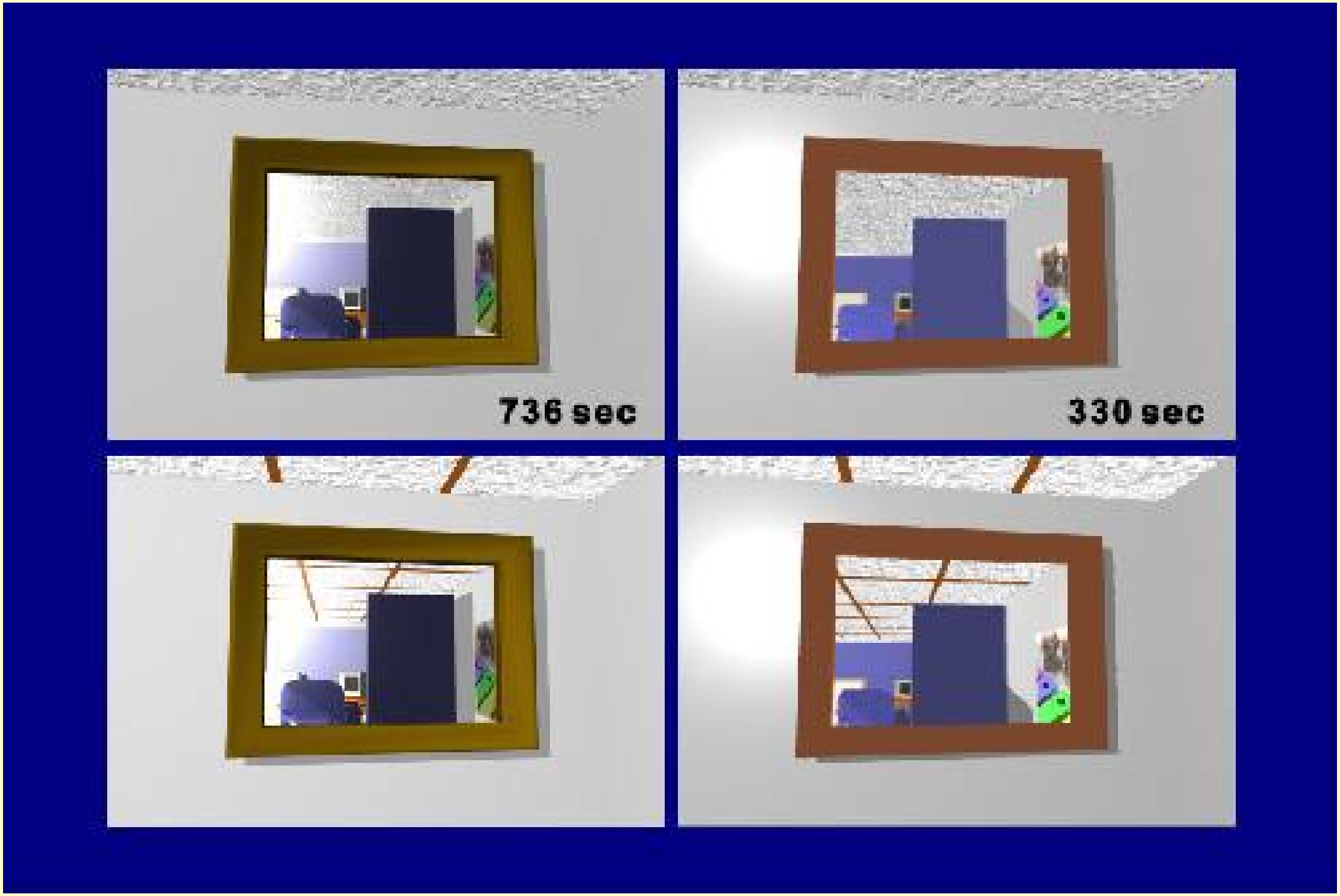


Reflection Mapping

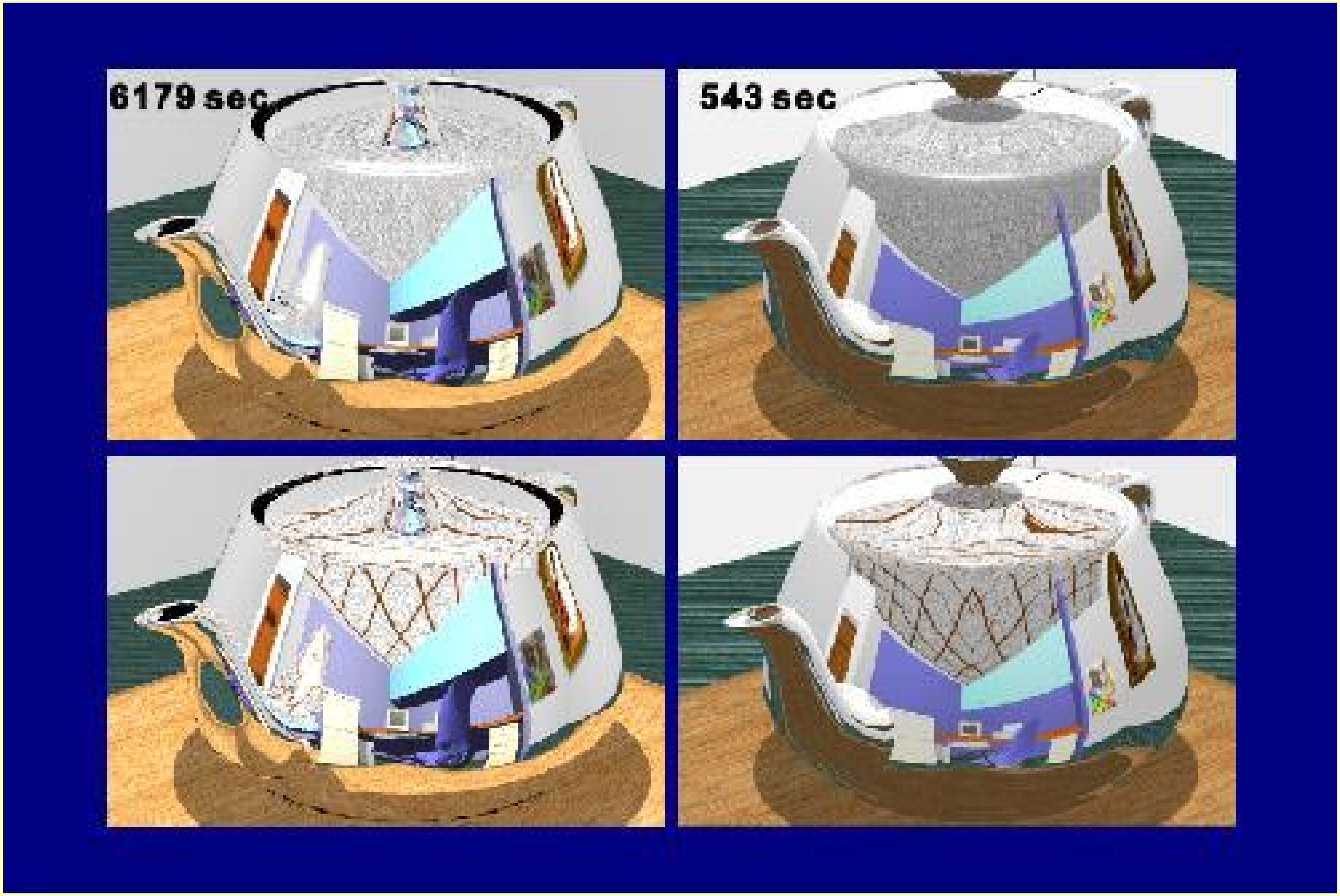
Ray Tracing

Reflection Mapping

Reflection Mapping



Reflection Mapping



Cube Environment Map

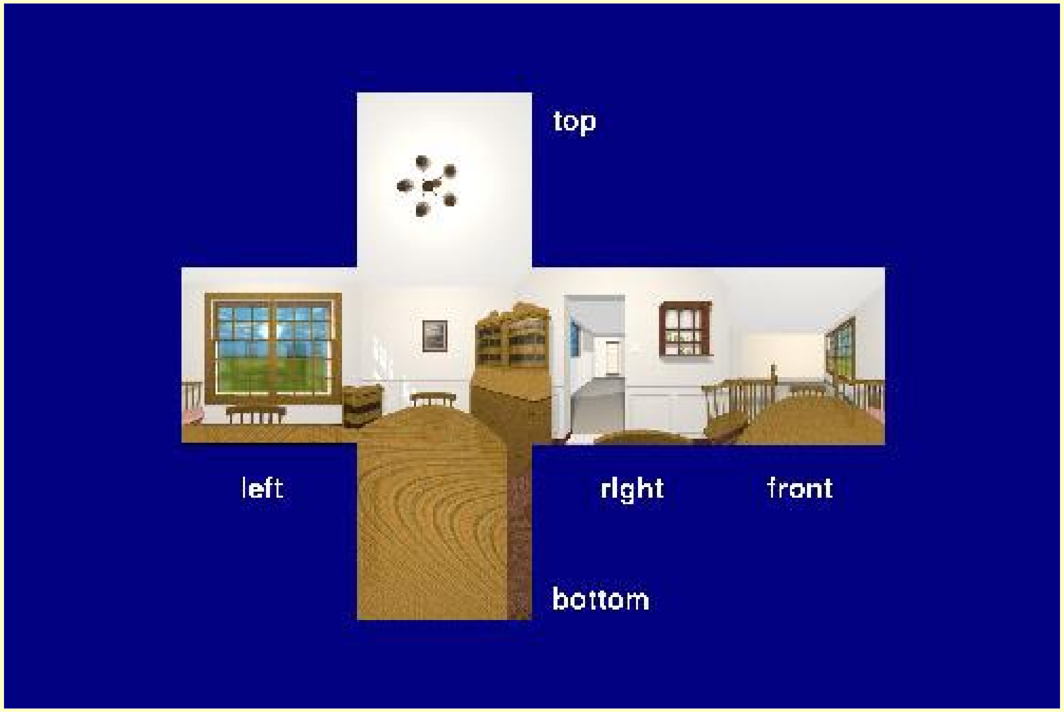
Choose cube face from largest vector component

+x: +x > ±y, ±z	+y: +y > ±z, ±x	+z: +z > ±x, ±y
–x: –x > ±y, ±z	–y: –y > ±z, ±x	–z: –z > ±x, ±y

Perspective project onto face

+x: y/x, z/x

Transform resulting –1 to 1 into texture space



Sphere Environment Map

Texture = reflection from shiny sphere

- Normal for reflection = $\hat{\mathbf{h}} = (\hat{\mathbf{v}} + \hat{\mathbf{r}}) \,/\, |\hat{\mathbf{v}} + \hat{\mathbf{r}}|$
- Transform x and y components to texture coord

Advantages

- Easy to aquire
- Easy to use

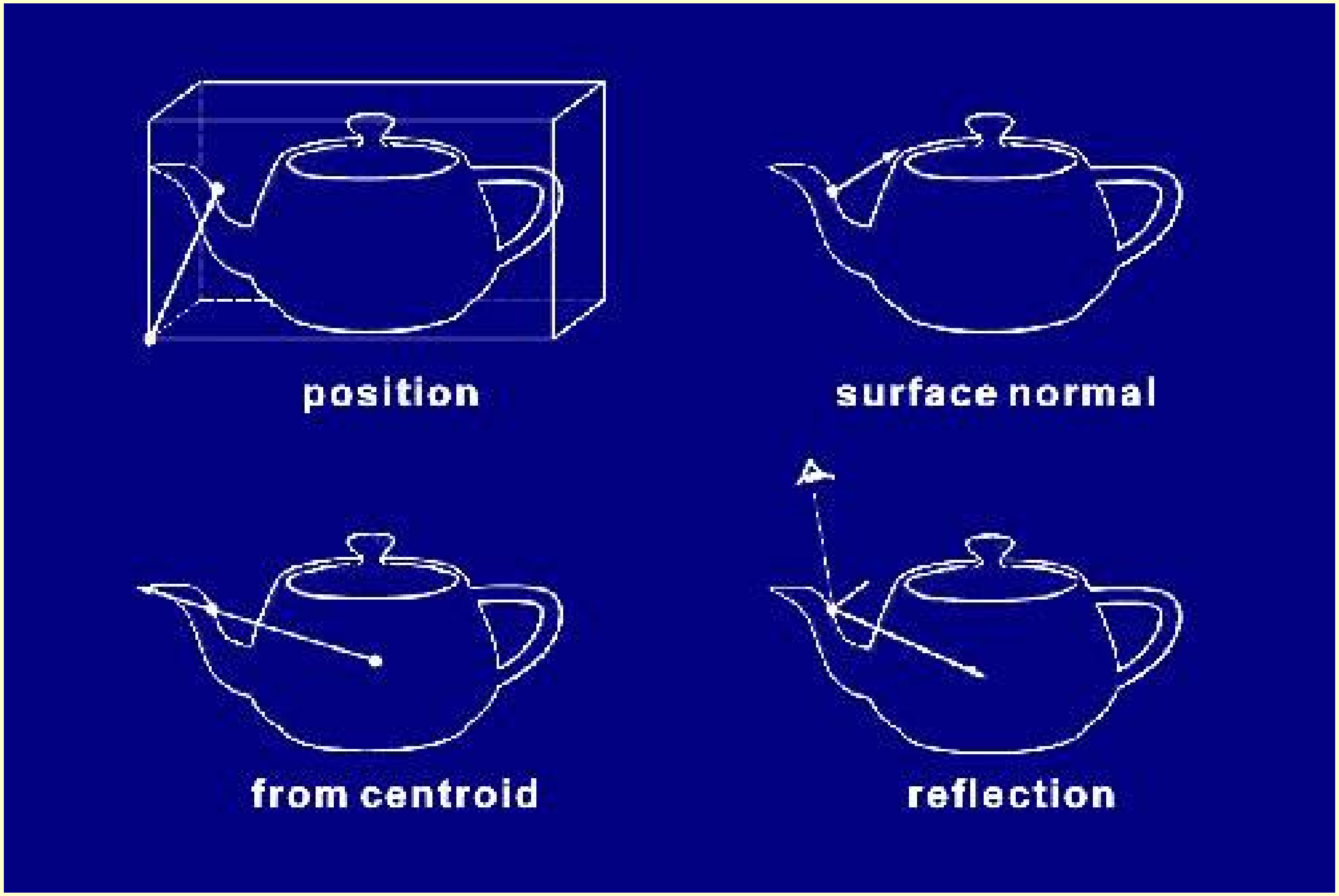
Disadvantages

- Fixed view
- Poor sampling near edge

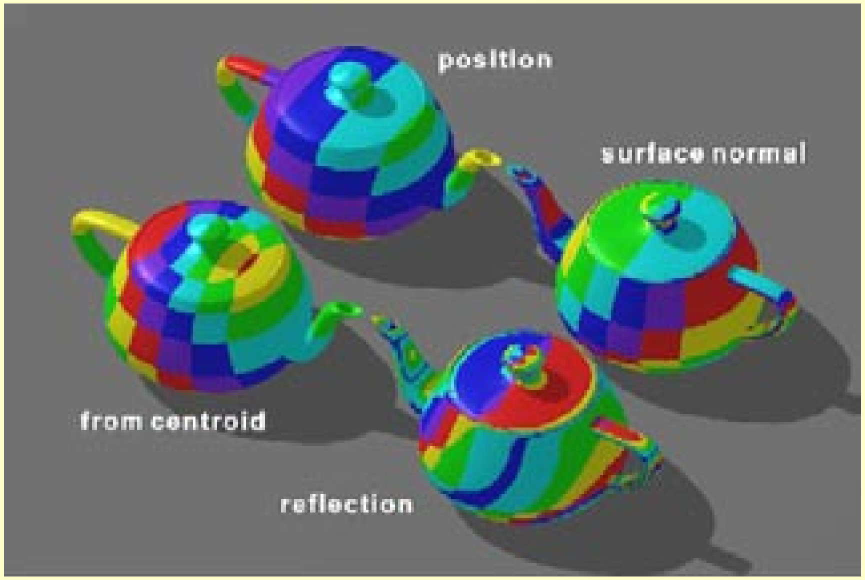
Map Entity vs. Map Shape

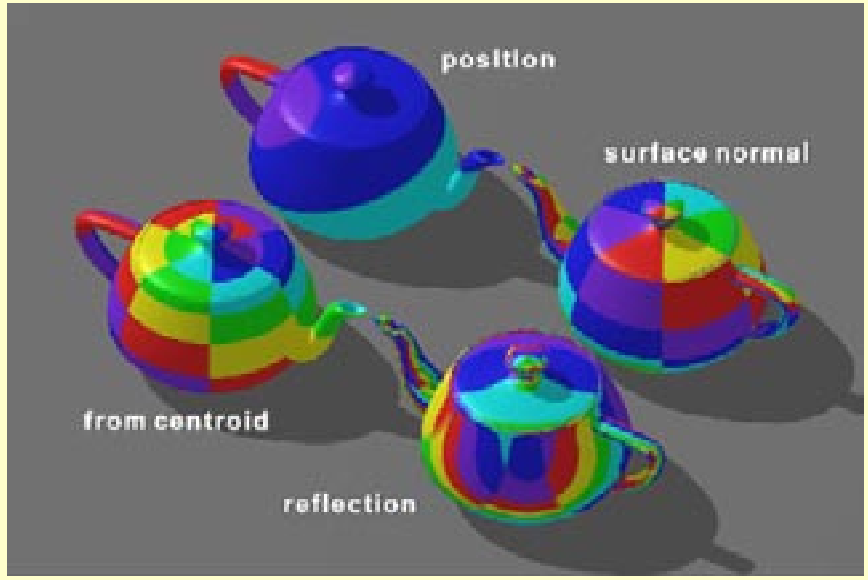
Map shape: sphere, cylinder, cube, plane, ...

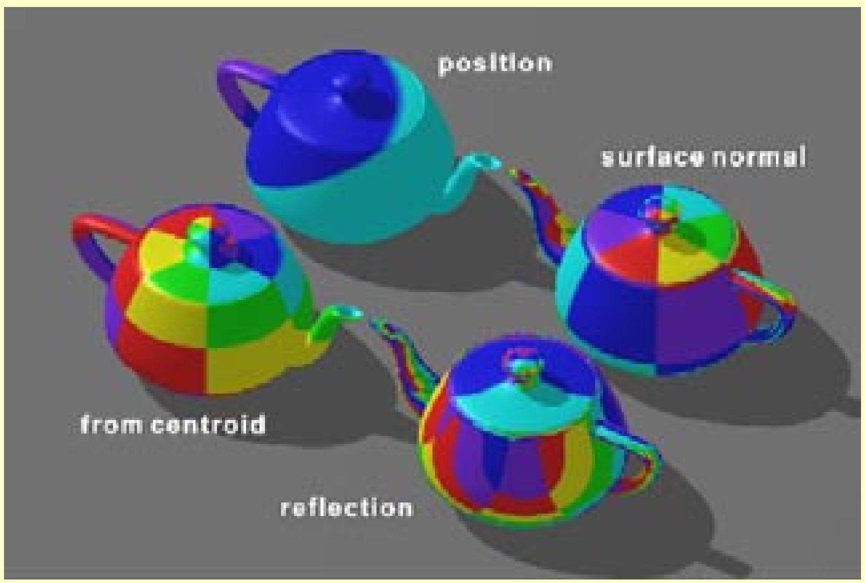
Map entity: position, normal, reflection vector, ...

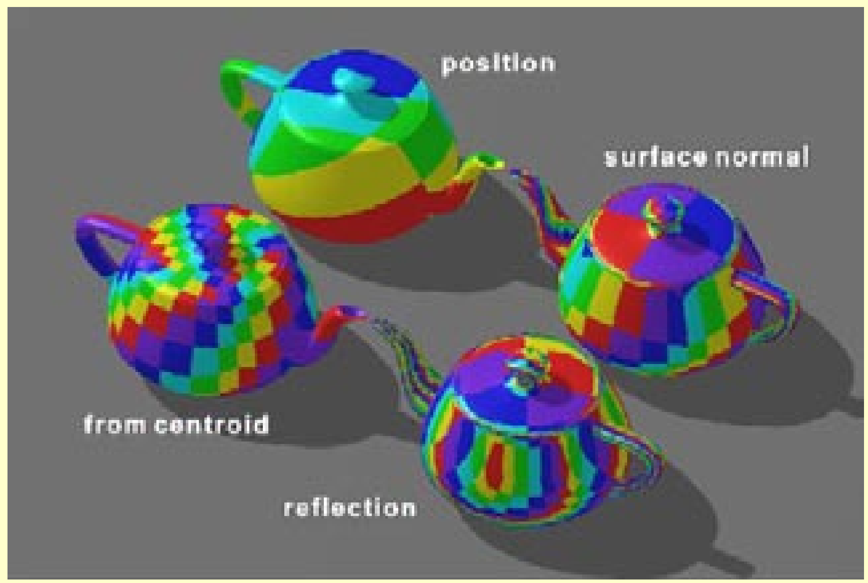


Map Entities



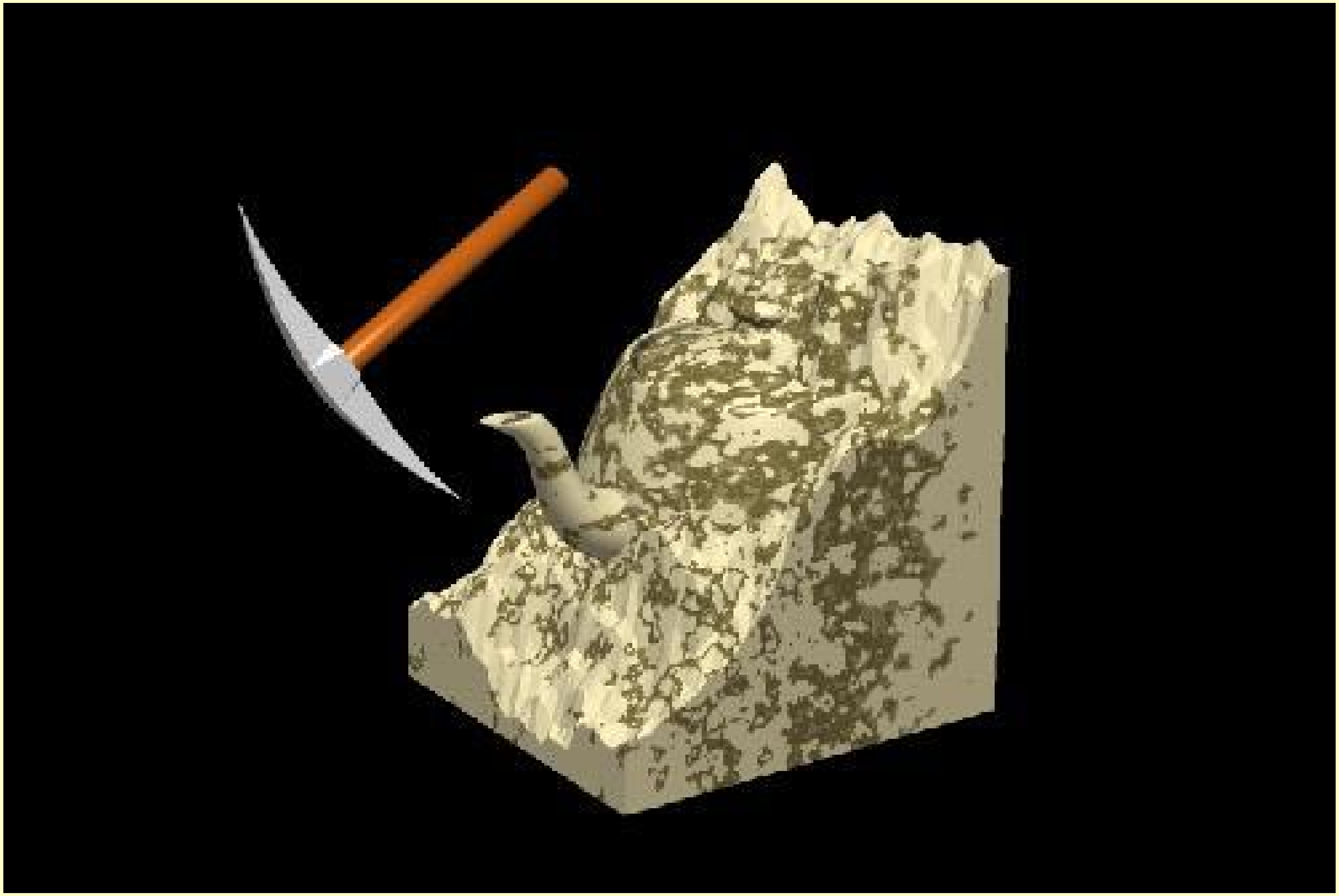






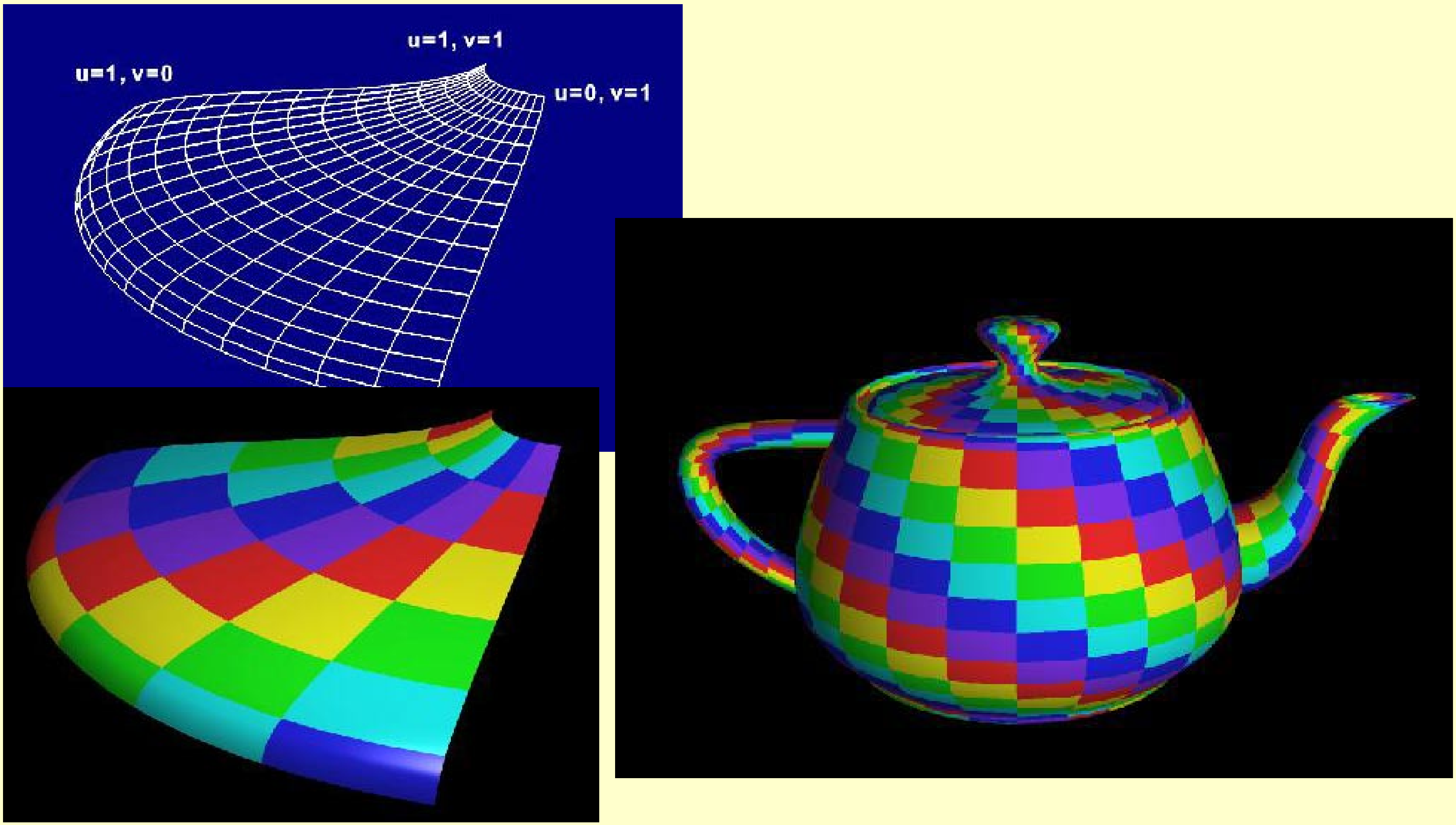
Solid/3D texture

3D Position in 3D volume texture



Parametric Patches

Use scaled surface u,v for texture u,v



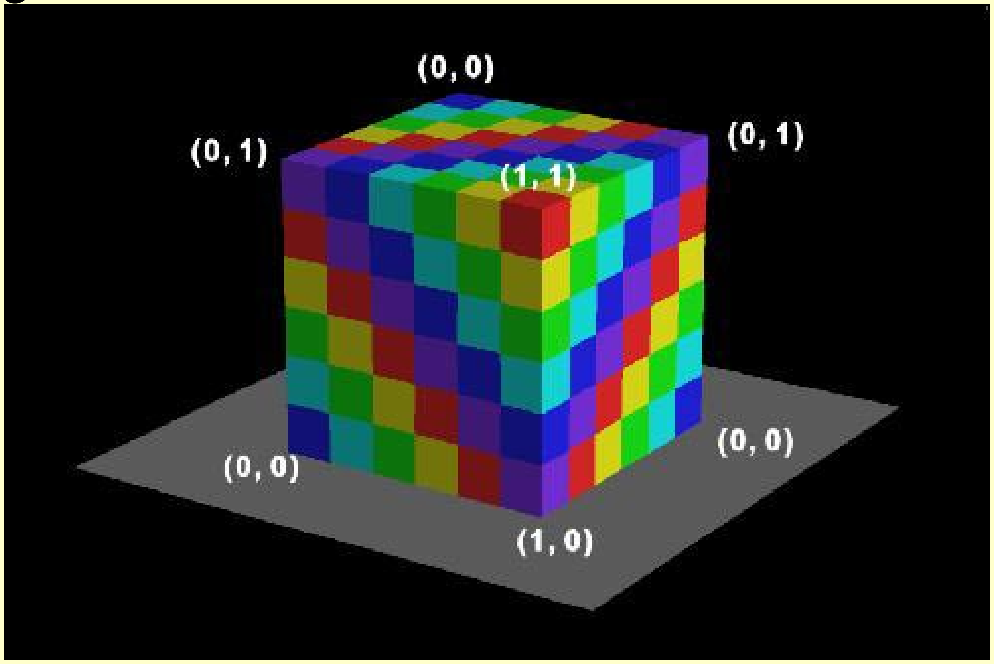
Mapping onto Polygons

Like parametric surfaces, but use explicit vertex texture coords

Interpolation issues

- screen space interp results in errors from nonlinearity and lack of rotational invariance
- use small polygons to minimize artifacts

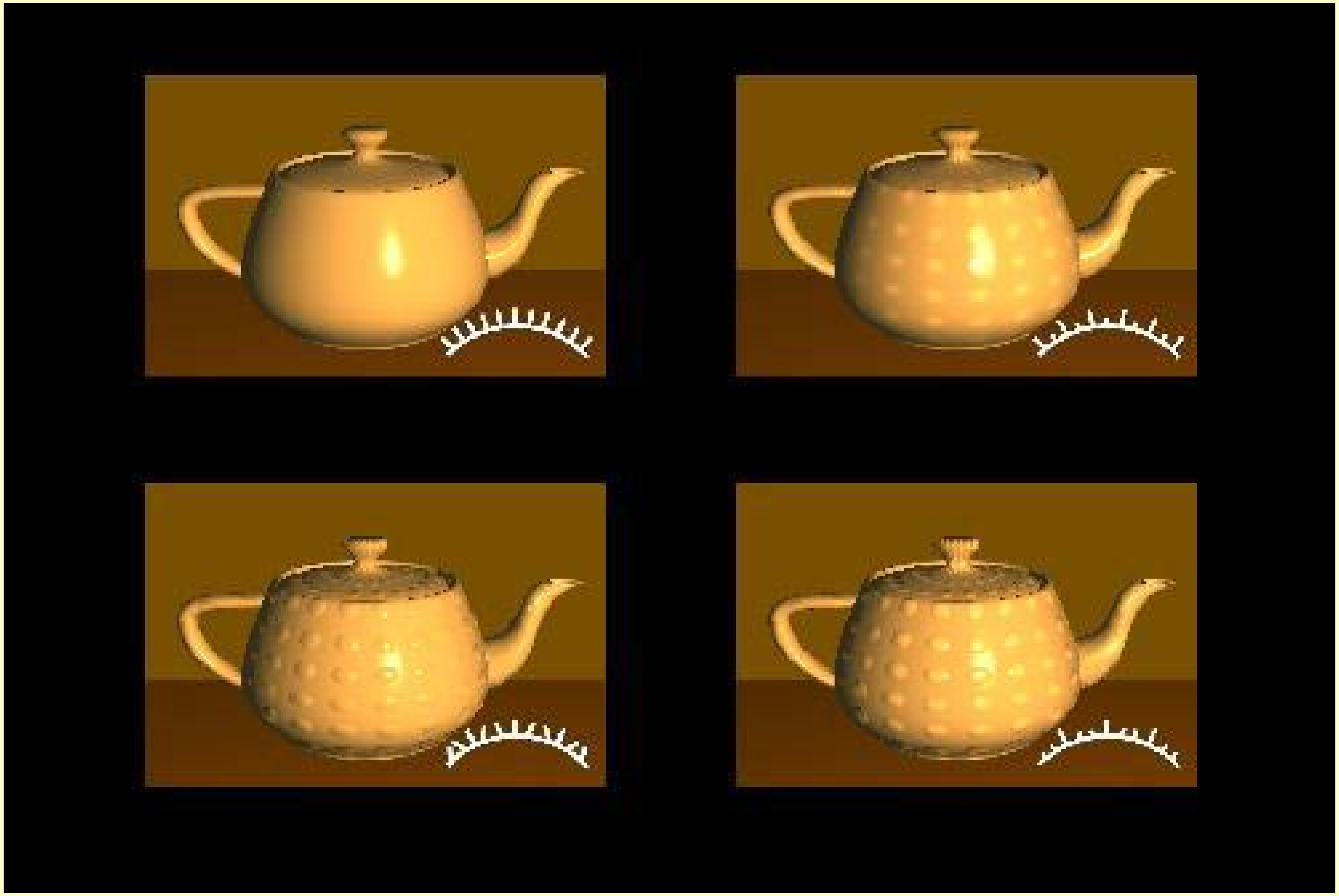
Correct solution: actual projection at each pixel



Bump Mapping

Purterb surface normal

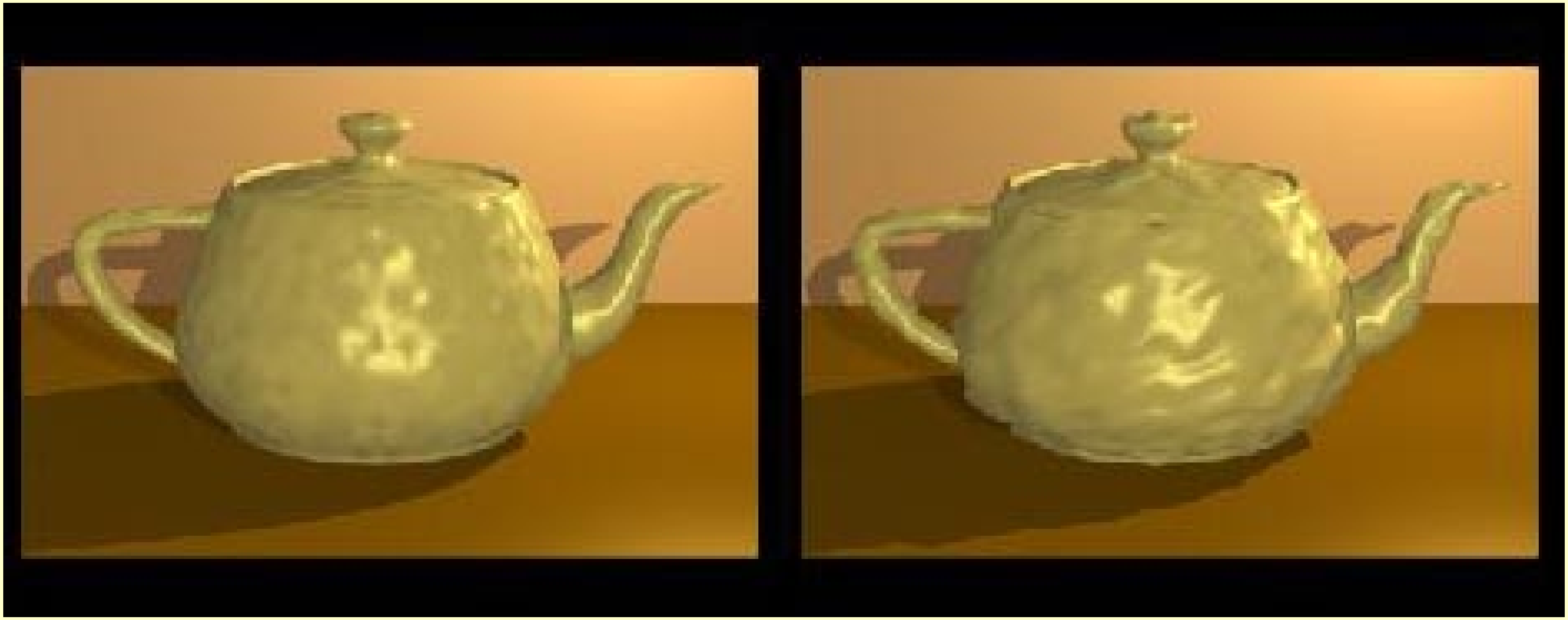
- Simulate minor shape variations



Bump Mapping

Displacement Mapping

Actually perturb surface



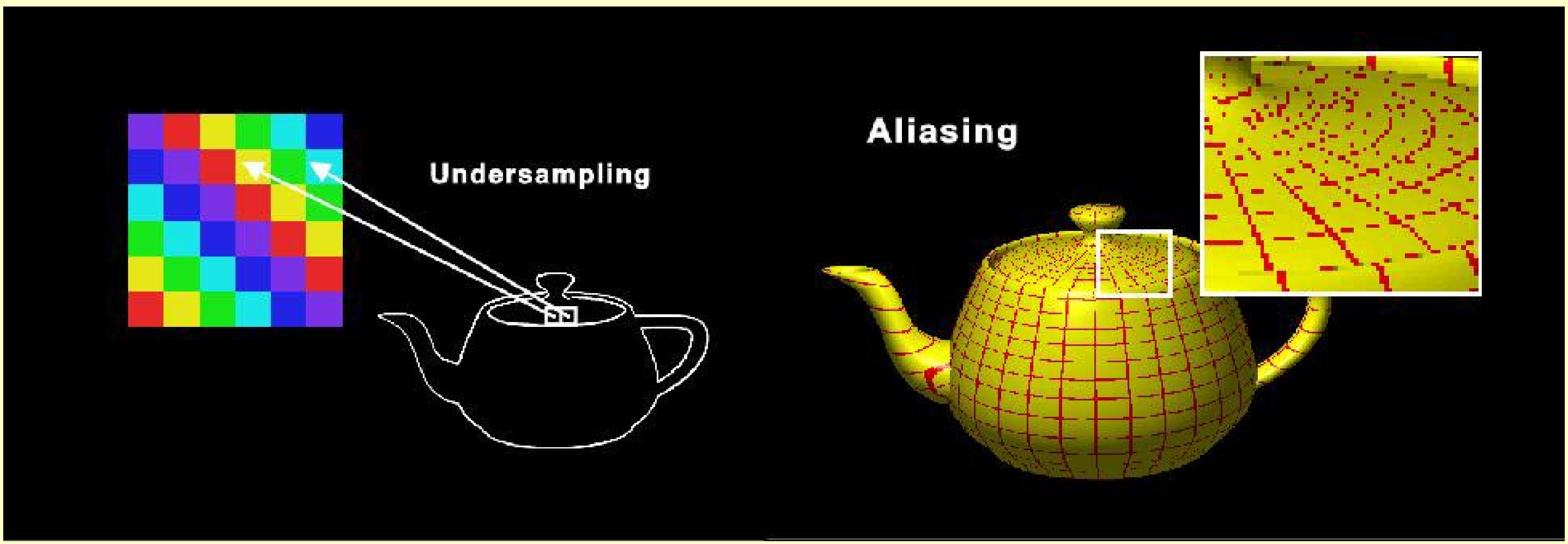
Texture Sampling

Oversampling: shows limited texture resolution

- Magnification

Undersampling: aliasing

- Minification

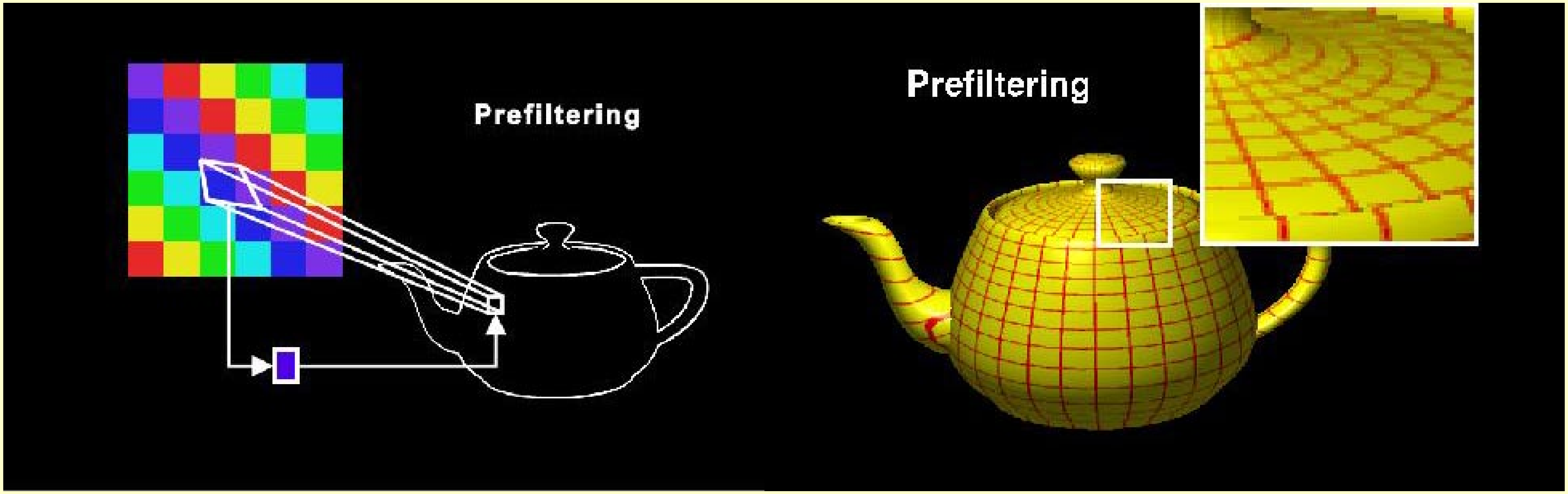


Texture Filtering

Similar to pixel antialiasing

BUT texture is known in advance

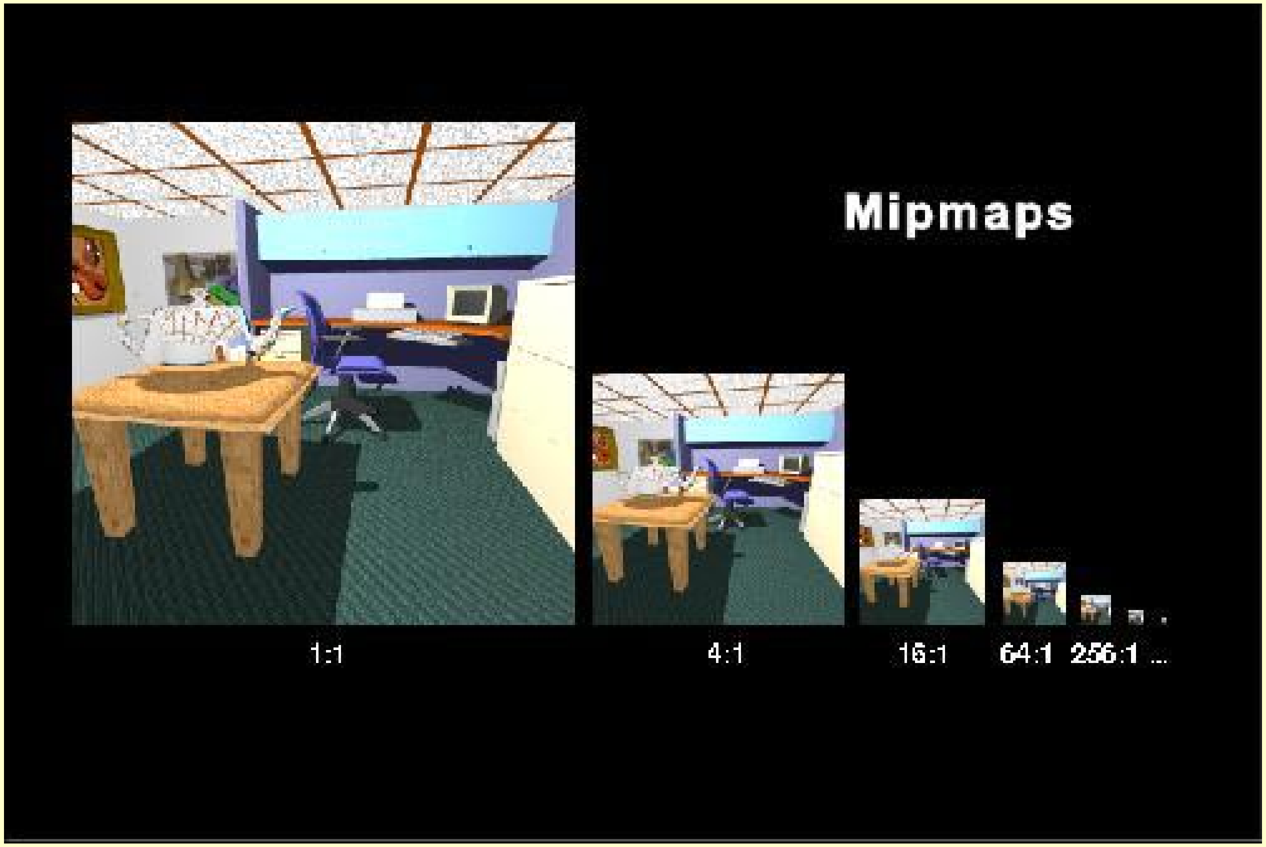
- Can improve overall antialiasing by filtering texture first
- Do faster and better by using prefiltering
  - Project pixel to texture, filter in texture space



Texture Prefiltering

MIPMap (Williams 78)

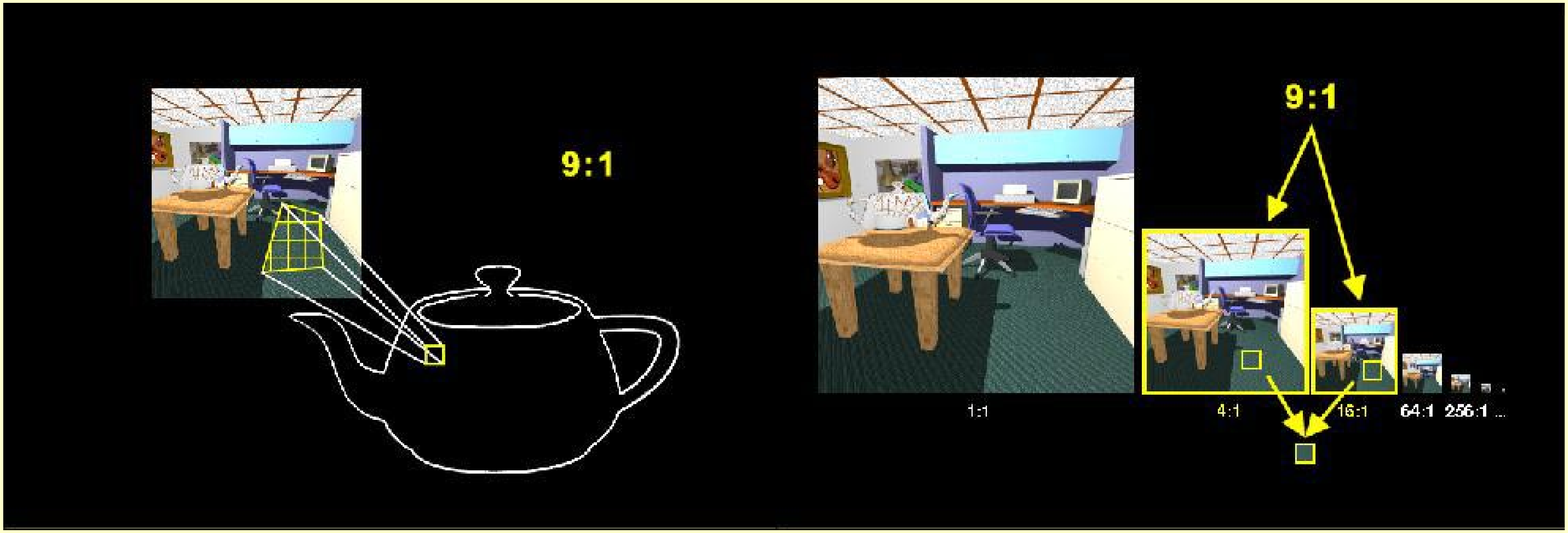
- Pre-filter at a sequence of resolutions
- Use one or several closest to correct filter size



Mipmap Reconstruction

Trilinear filtering:

- Find area in texture
- Average texels within two closest mipmaps
- Average between two closest mipmaps



Texture + Pixel Antialiasing