CS130 : Computer Graphics

Lecture 9: Texture Mapping

Tamar Shinar
Computer Science & Engineering
UC Riverside

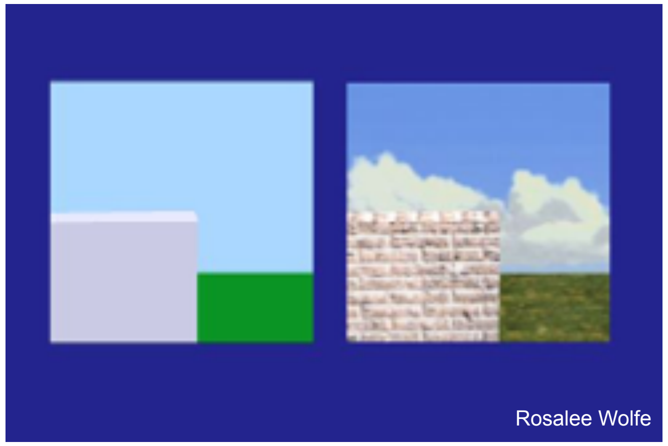
There are limits to geometric modeling

http://www.beinteriordecorator.com

National Geographic

Although modern GPUs can render millions of triangles/sec, that's not enough sometimes...

Use texture mapping to increase realism through detail



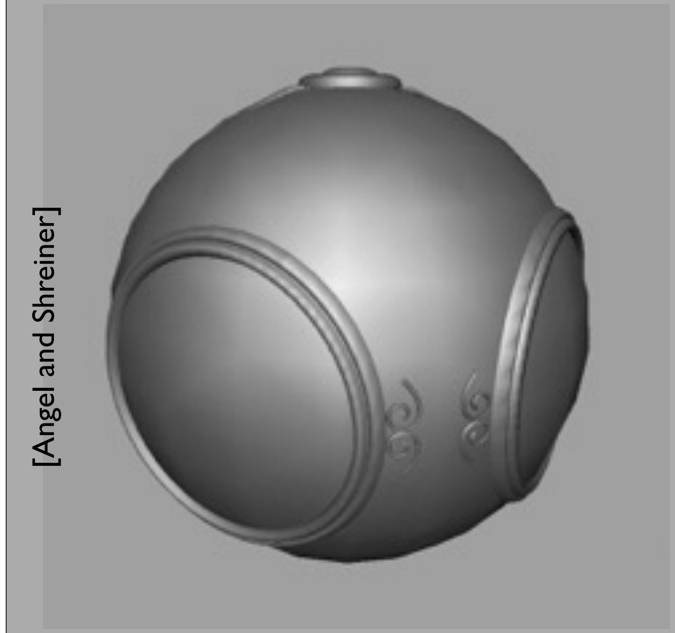
Rosalee Wolfe

This image is just 8 polygons!

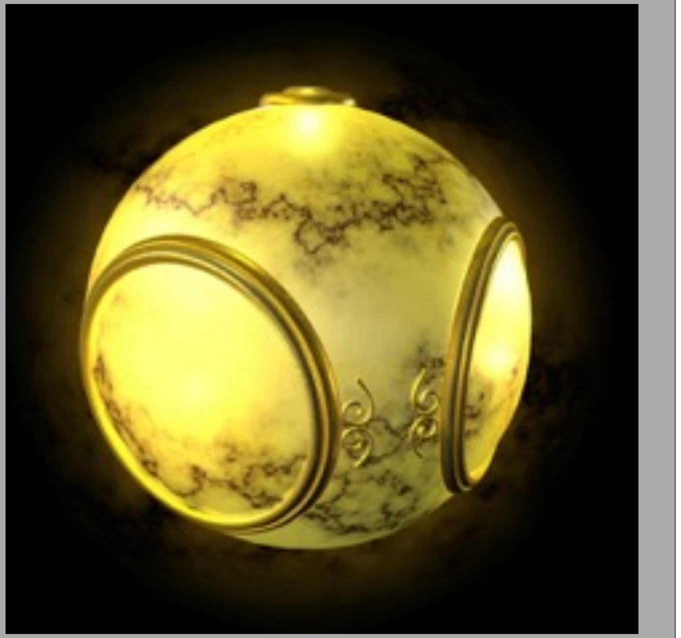
Add visual complexity.

http://www.siggraph.org/education/materials/HyperGraph/mapping/r_wolfe/r_wolfe_mapping_1.htm

[Angel and Shreiner]

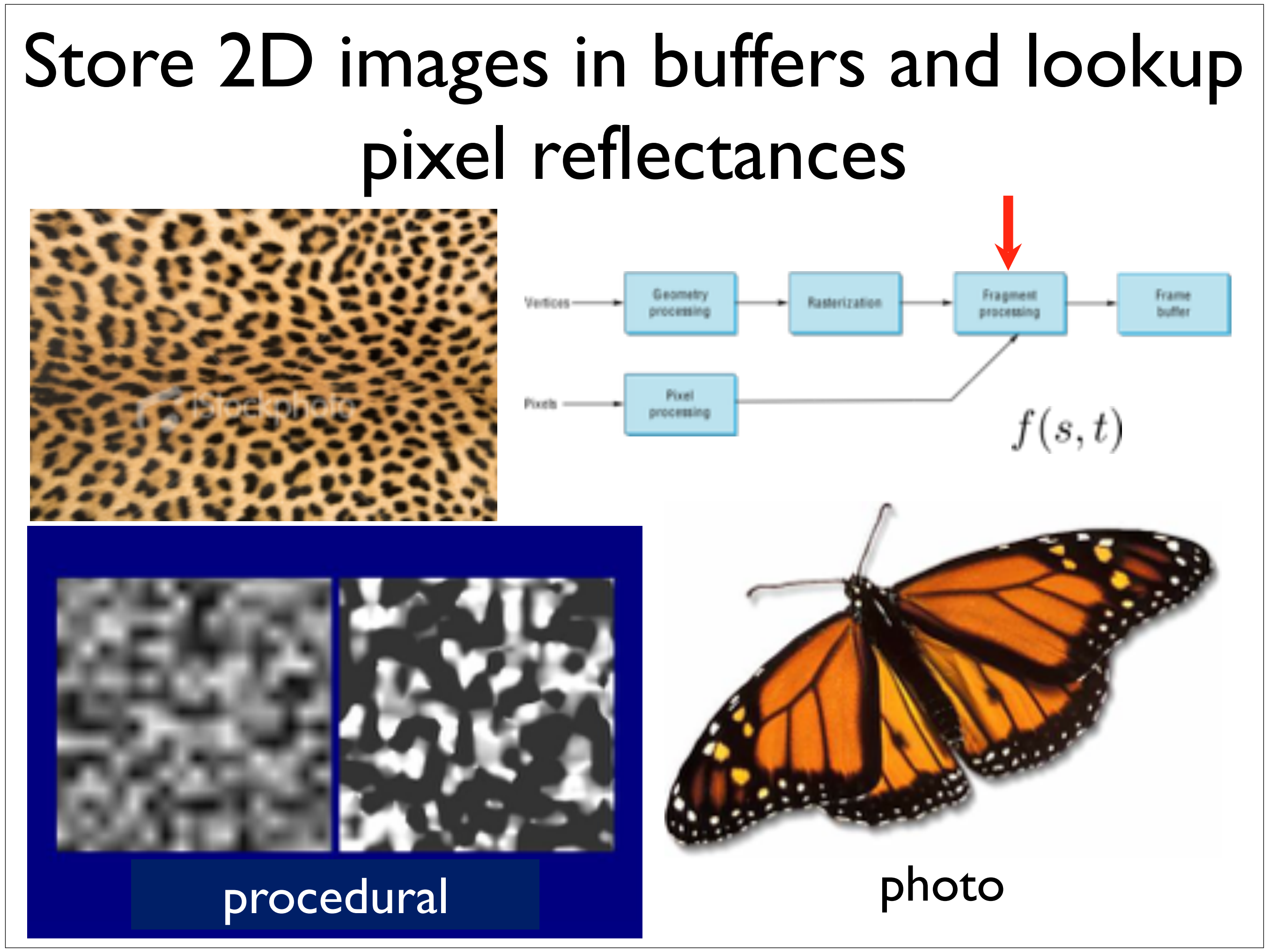


No texture



With texture

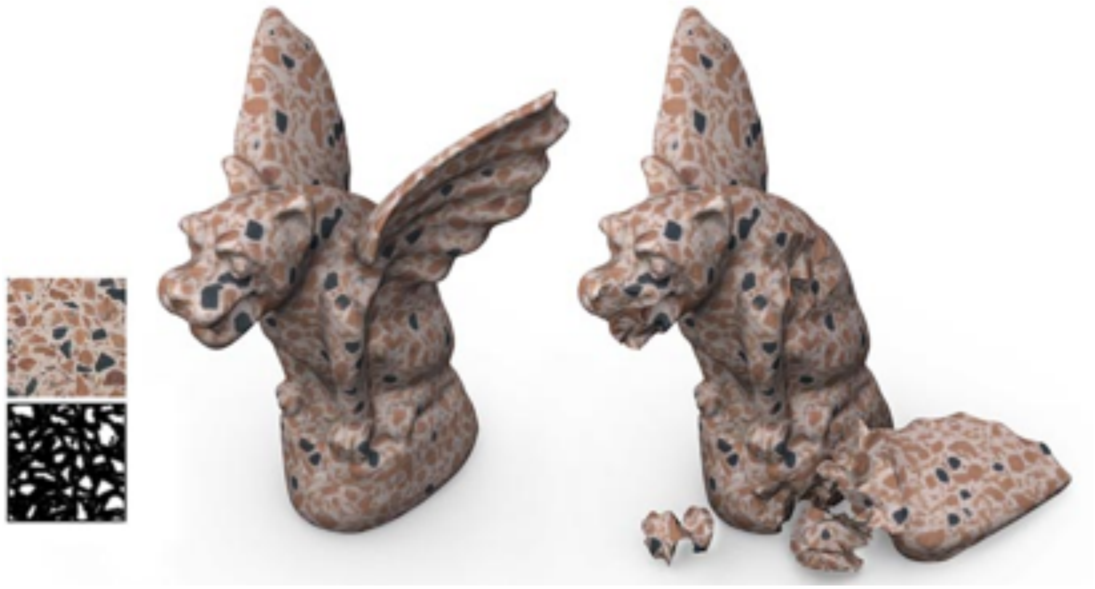
Pixar - Toy Story



Store 2D images in buffers and lookup pixel reflectances

Textures can be anything that you can lookup values in -- photo, procedurally generated, or even a function that computes a value on the fly

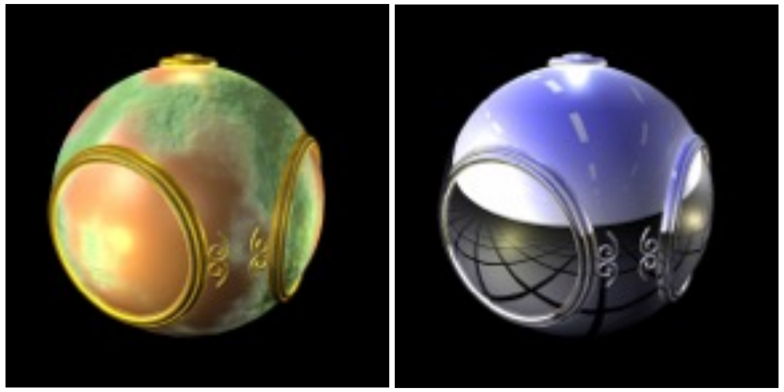
3D solid textures



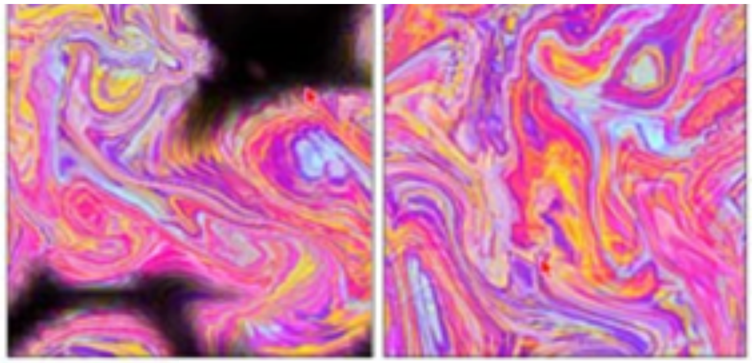
[Dong et al., 2008]

Other uses of textures...

Light maps
Shadow maps
Environment maps
Bump maps
Opacity maps
Animation

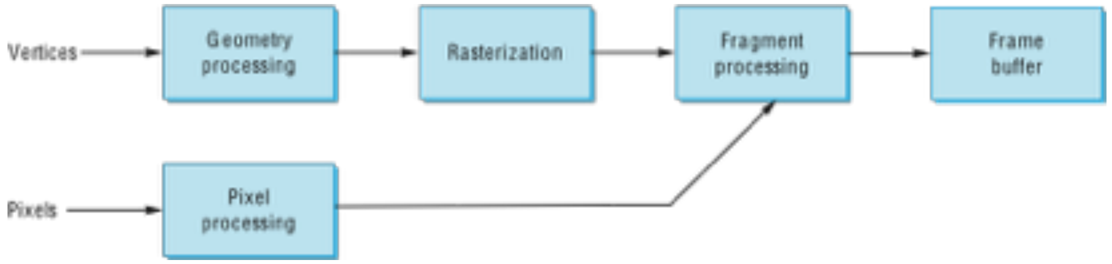


[Angel and Shreiner]



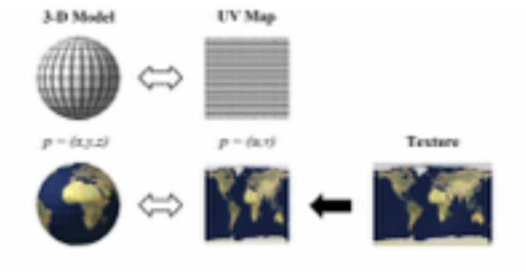
[Stam 99]

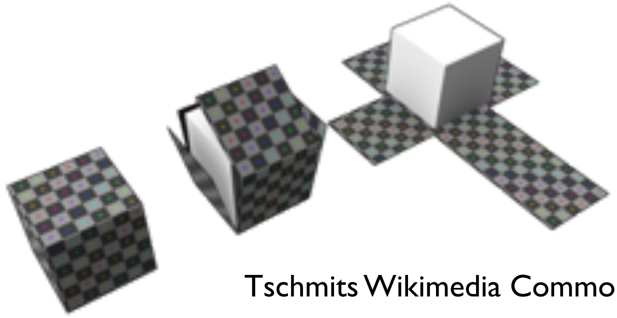
Texture mapping in the OpenGL pipeline



- Geometry and pixels have separate paths through pipeline
- meet in **fragment processing** - where textures are applied
- texture mapping applied at end of pipeline - efficient since relatively few polygons get past clipper

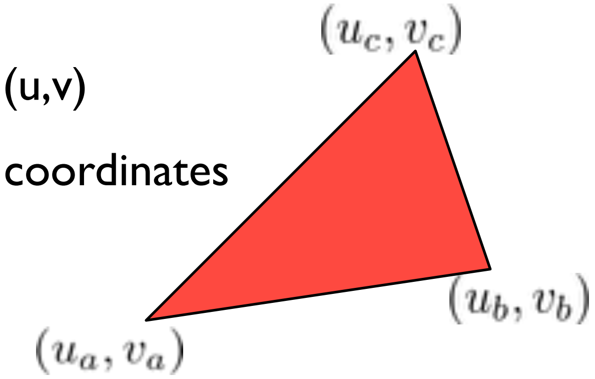
uv Mapping



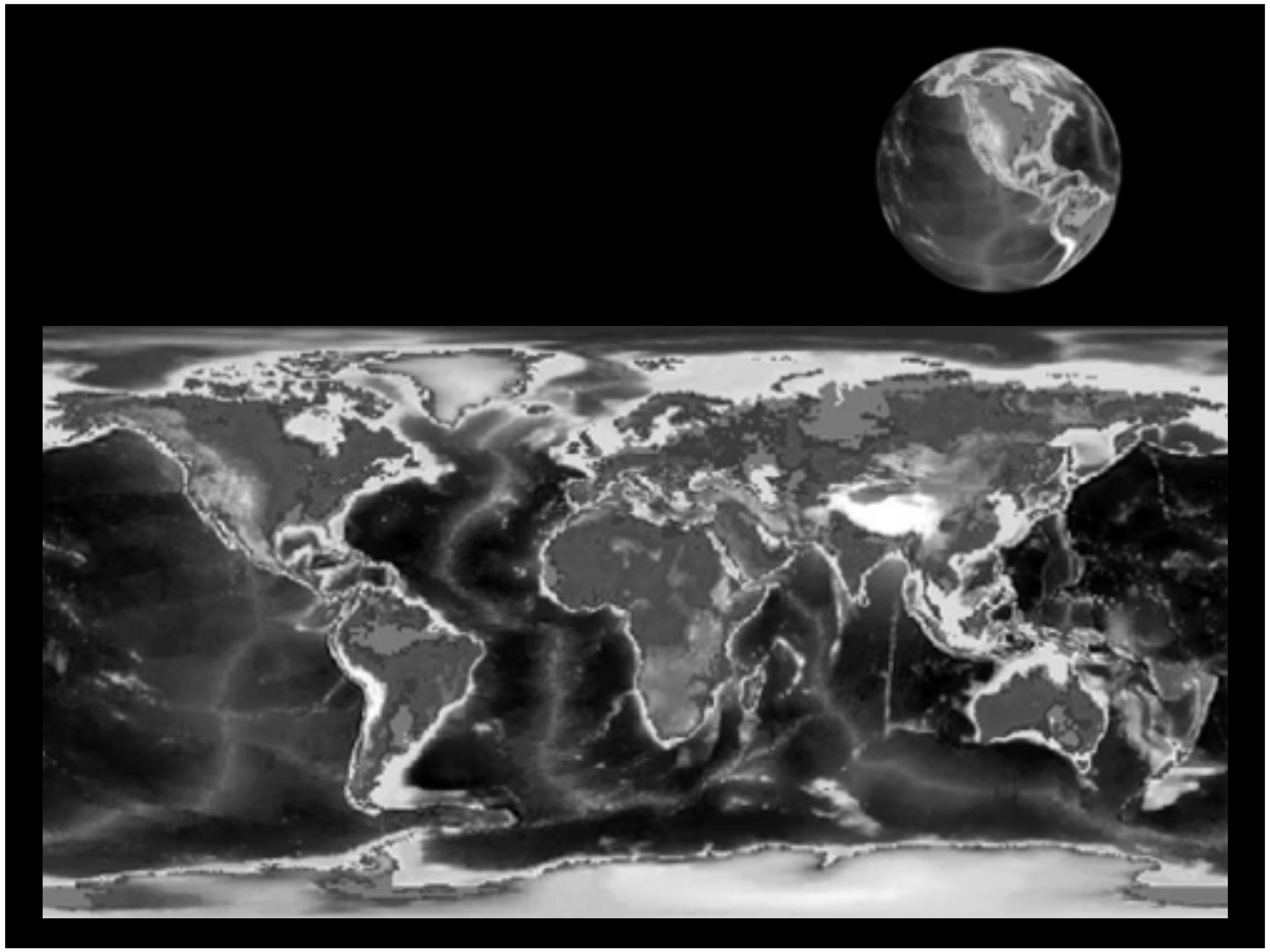


Tschmits Wikimedia Commons

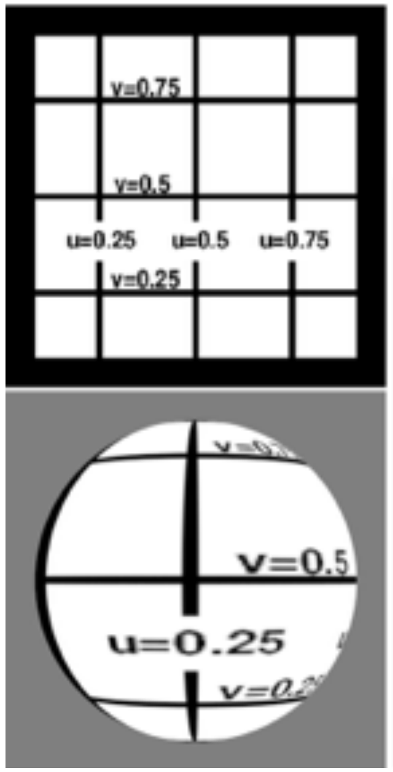
- 2D texture is parameterized by (u,v)
- Assign polygon vertices texture coordinates
- Interpolate within polygon

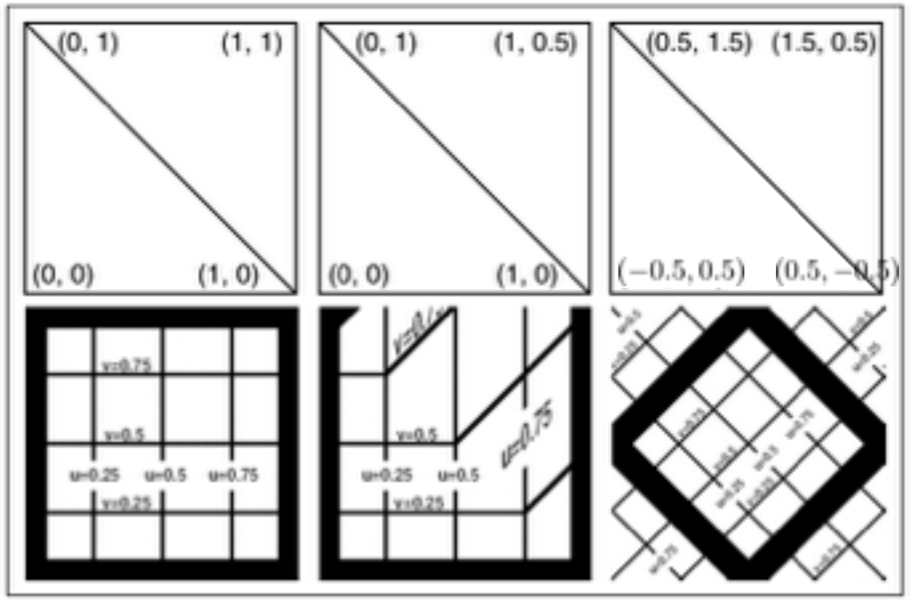


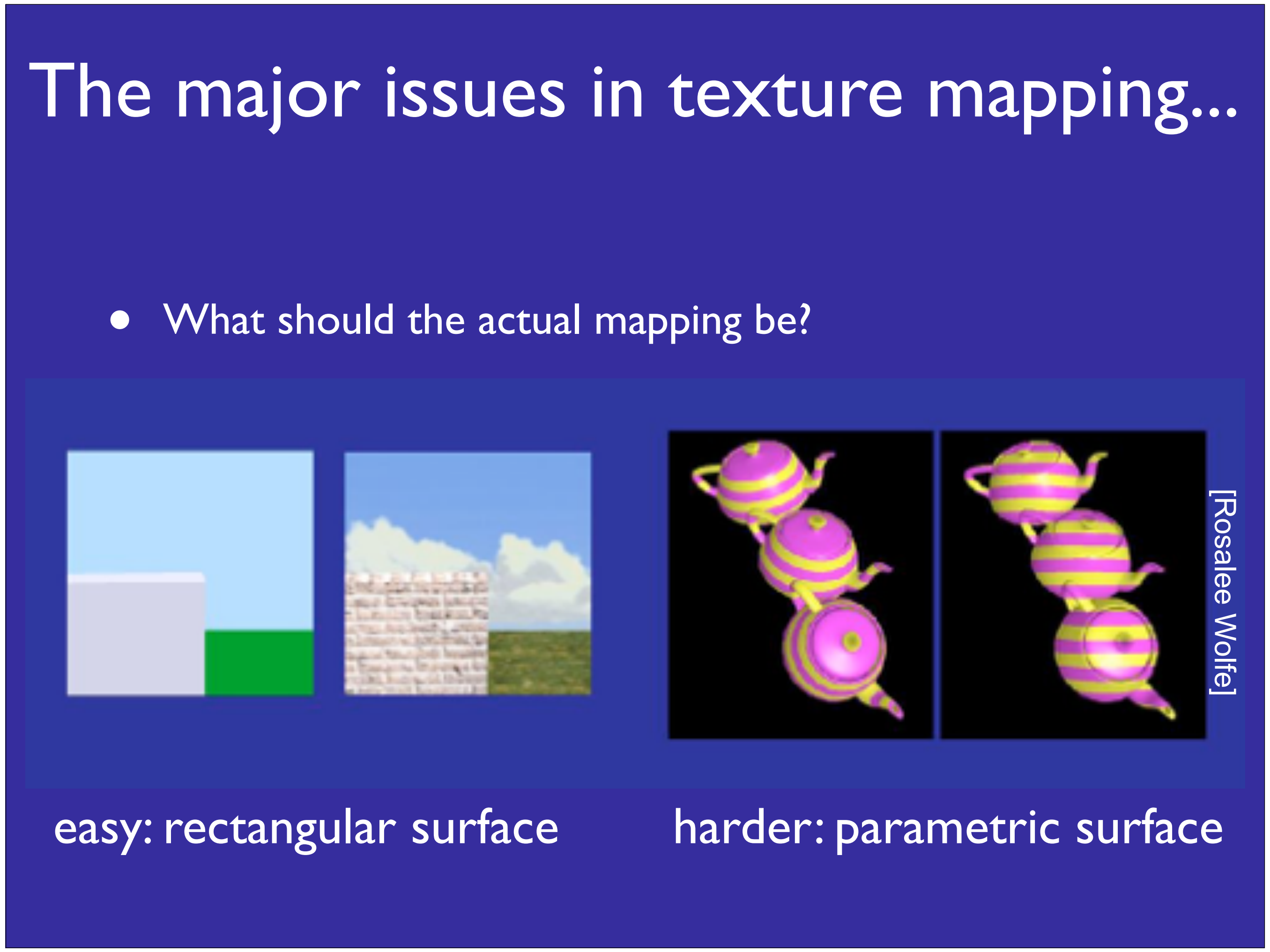
Texture coordinates are per-vertex data - a position in the (u,v) space
can interpolate tex coordinates with barycentric coordinates



Texture Calibration







The major issues in texture mapping...

- What should the actual mapping be?

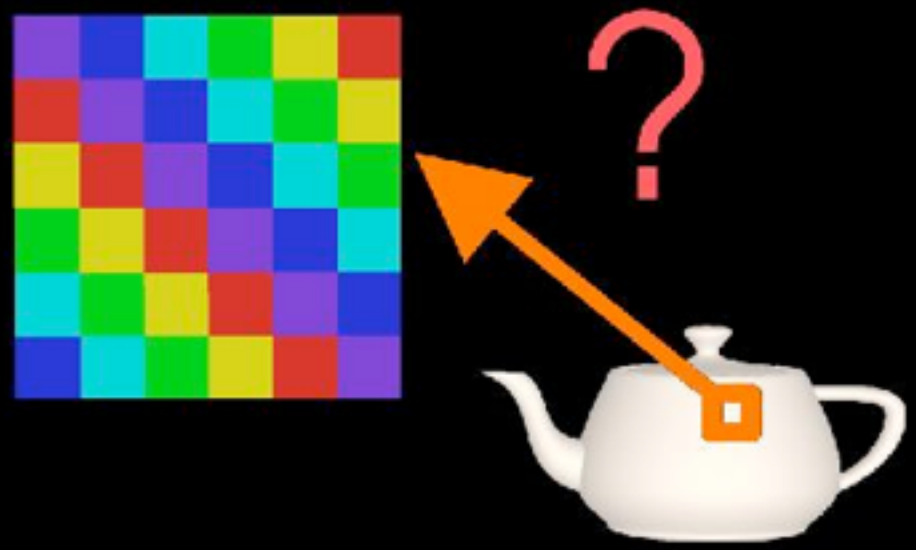
[Rosalee Wolfe]

easy: rectangular surface

harder: parametric surface

Teapot: Which image looks better? The image on the left uses **object coordinates** in the texture mapping – this makes more sense. The image on the **right** uses **world coordinates** – texture ends up changing relative to the object
want a nice map that doesn't look distorted

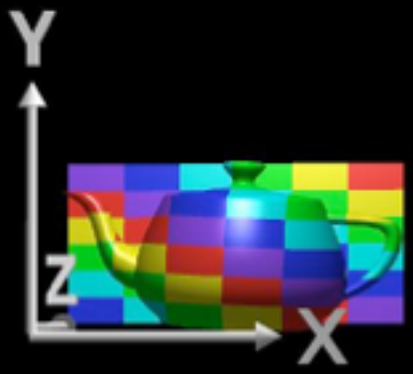
Given a point on the object **(x,y,z)**, what point **(u,v)** in the texture we use?

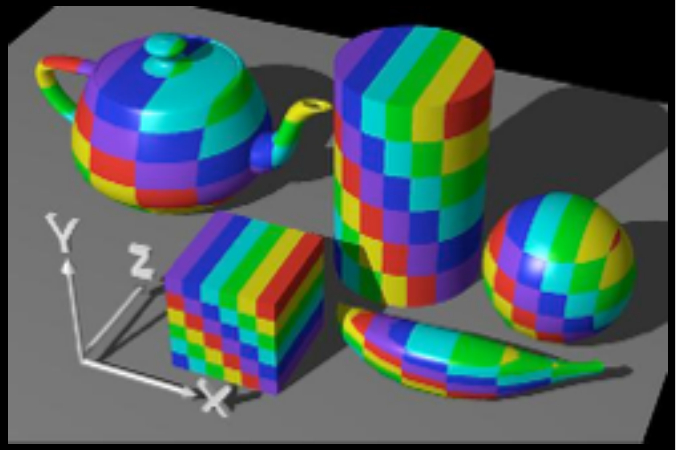


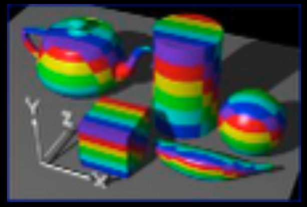
[Rosalee Wolfe]

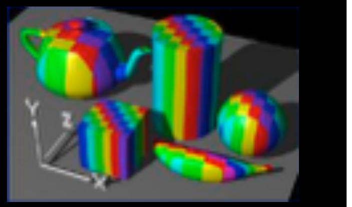
Example: planar mapping

[Rosalee Wolfe]



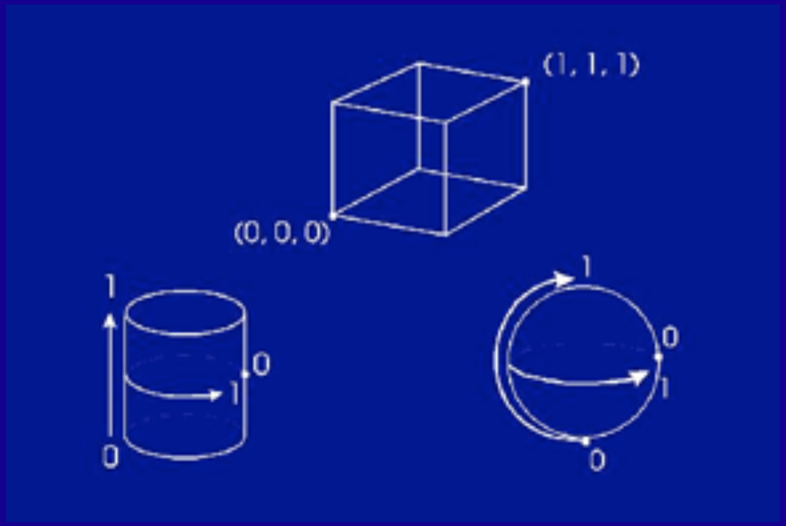






Intermediate surfaces

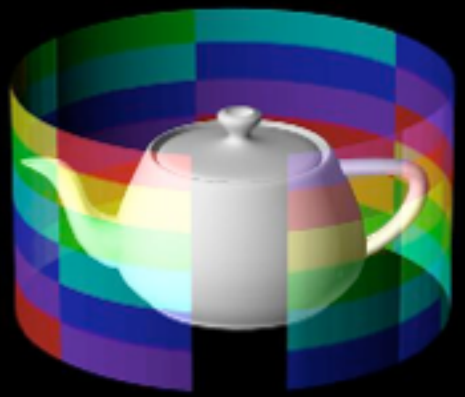
First map the texture to a simpler, intermediate surface



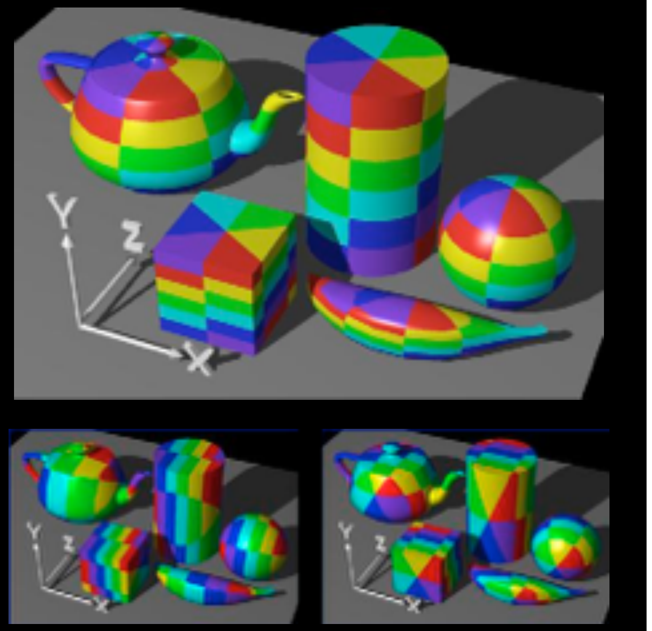
[Rosalee Wolfe]

Cylindrical mapping

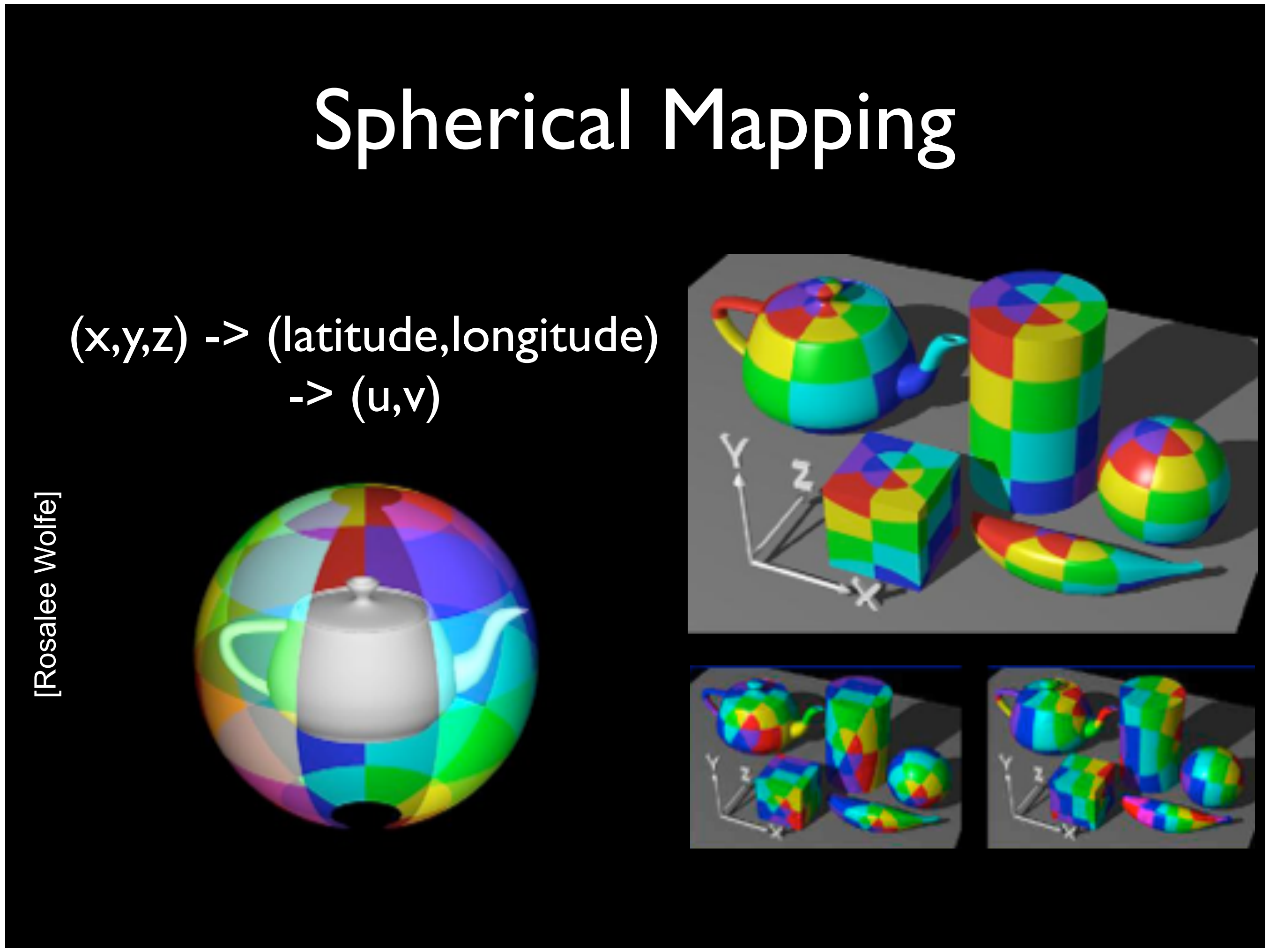
(x,y,z) -> (theta, h) -> (u,v)



[Rosalee Wolfe]



- note “pie slice” phenomena
- which coordinate axis is parallel to the cylinder axis?



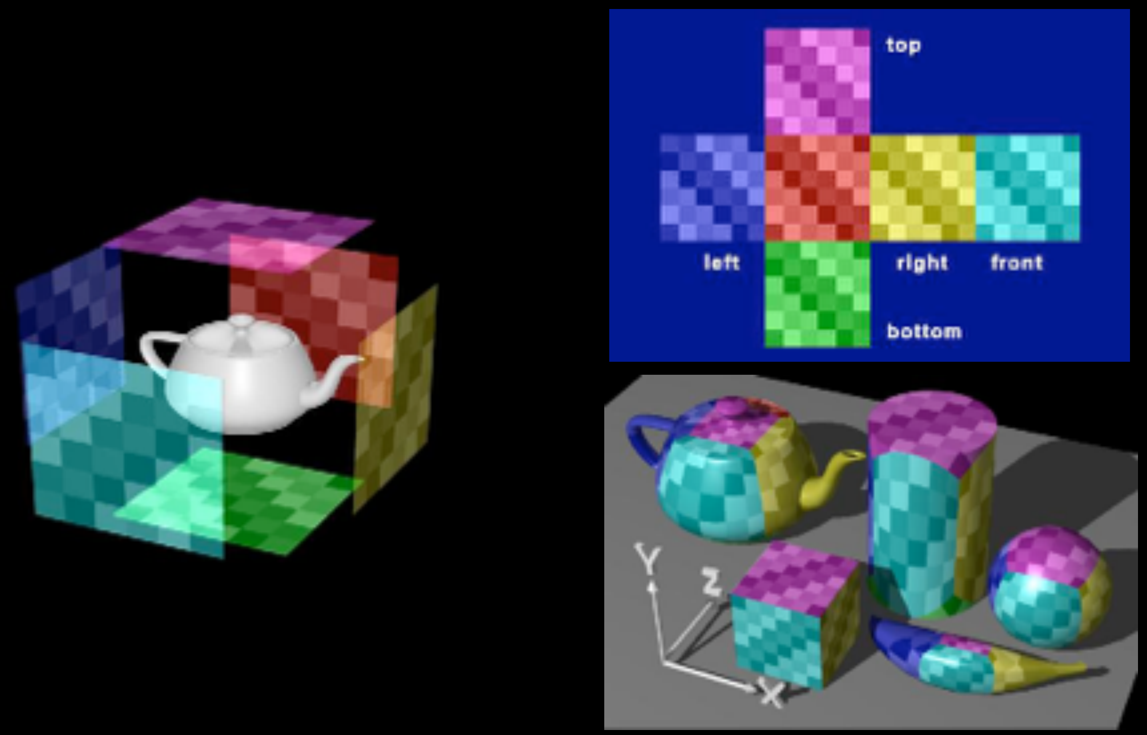
Spherical Mapping

(x,y,z) -> (latitude,longitude)
-> (u,v)

[Rosalee Wolfe]

spherical map stretches squares at equator and squeezes squares at poles

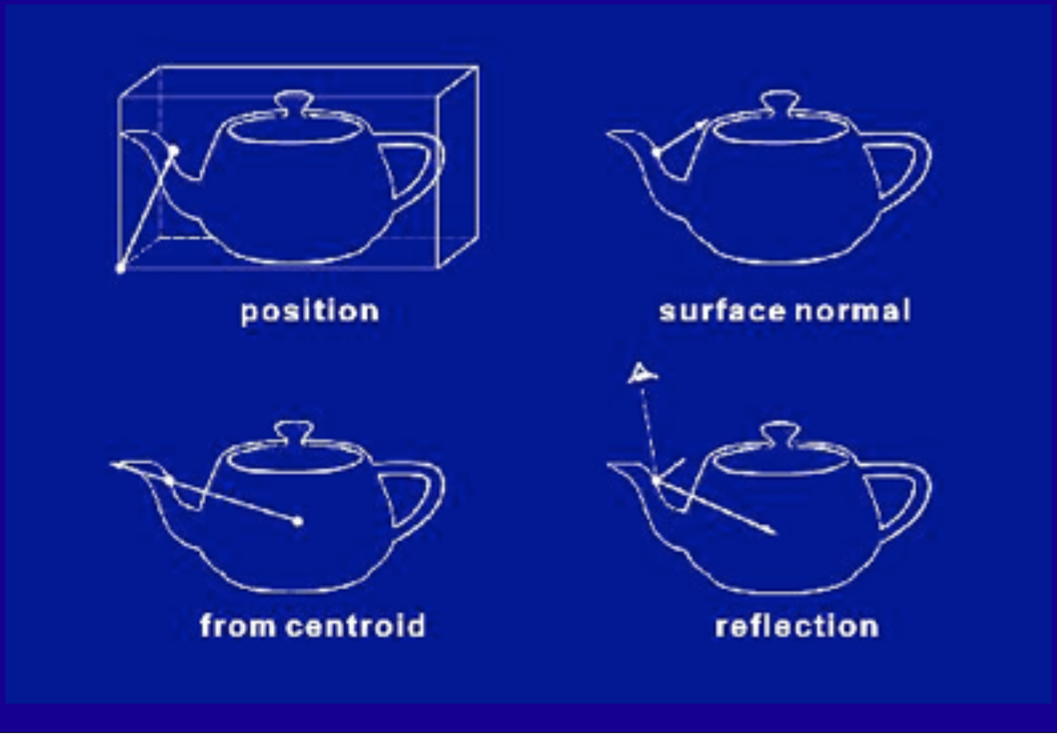
Box Mapping



[Rosalee Wolfe]

- similar to planar mapping
- planar projection -- choose which plane to project onto

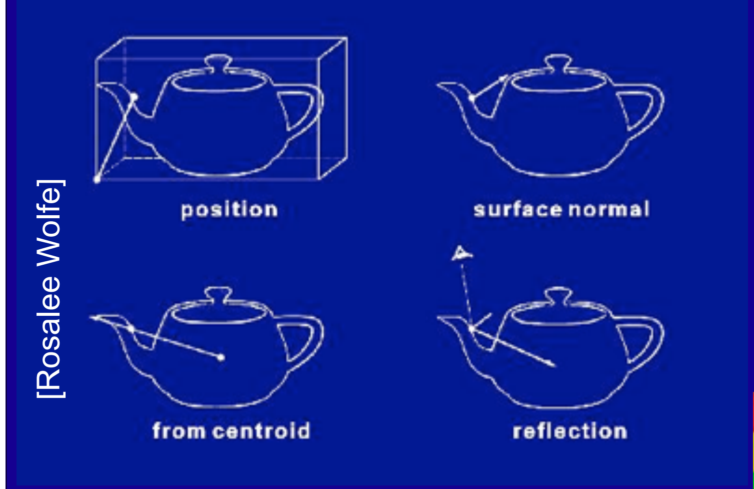
How do we map between intermediate and actual objects?



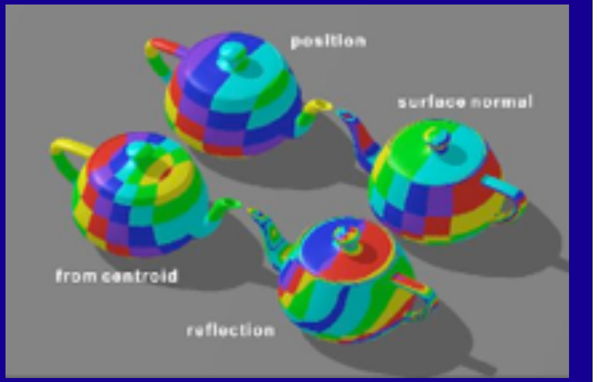
[Rosalee Wolfe]

We associated (x,y,z) on the intermediate object with the texture (u,v). But which point on the actual object is this?
We choose both the **intermediate shape** and the **mapping from the actual shape to the intermediate shape**
1. a point on the object relative to its bounding box
2. see where surface normal intersects intermediate surface
3. shoot ray from centroid through surface point to intermediate surface
4. use the reflection vector (depends on the viewer position and normal)

How do we map between intermediate and actual objects?

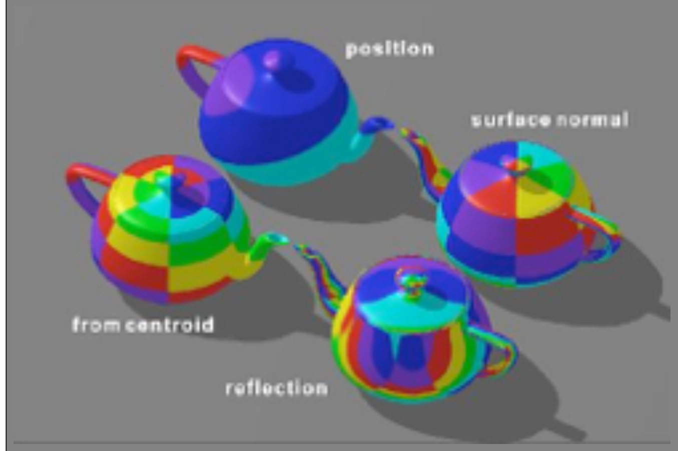


[Rosalee Wolfe]

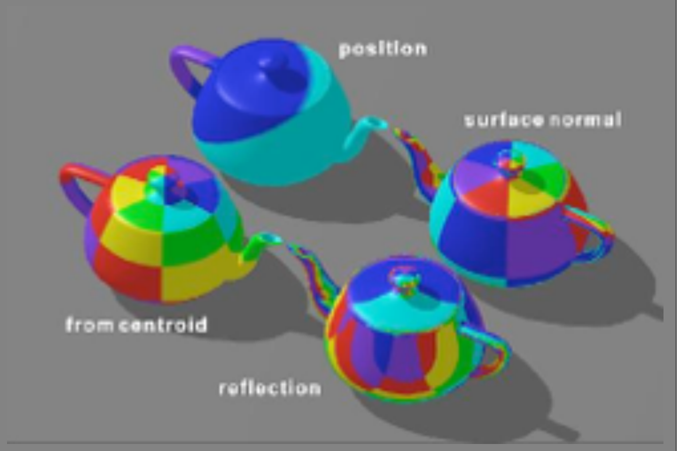


What intermediate shape was used here?

Can you tell what intermediate shape was used?
Planar map - in xy plane

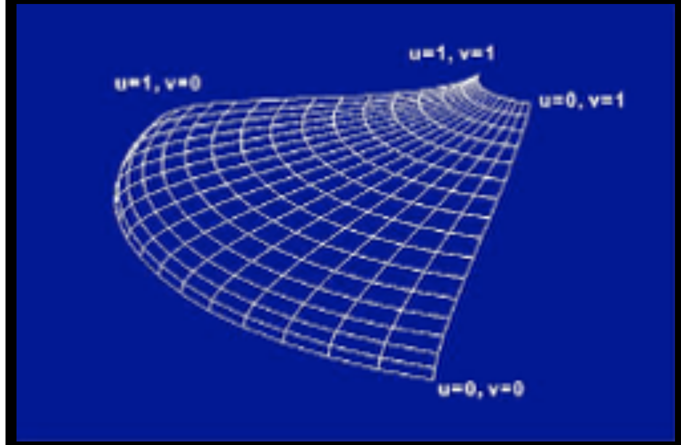


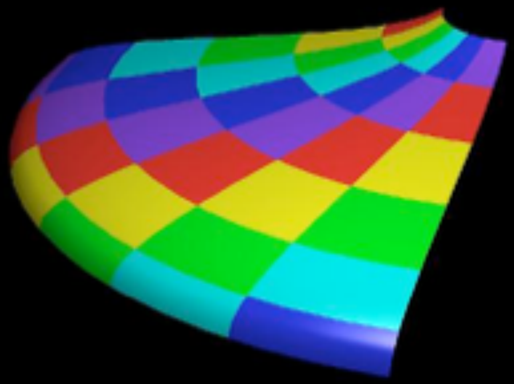
Cylindrical



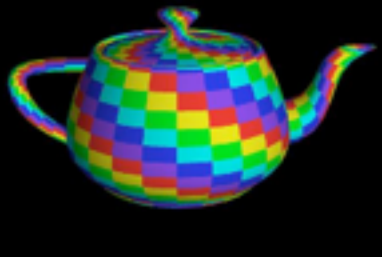
Spherical

Parametric Surfaces

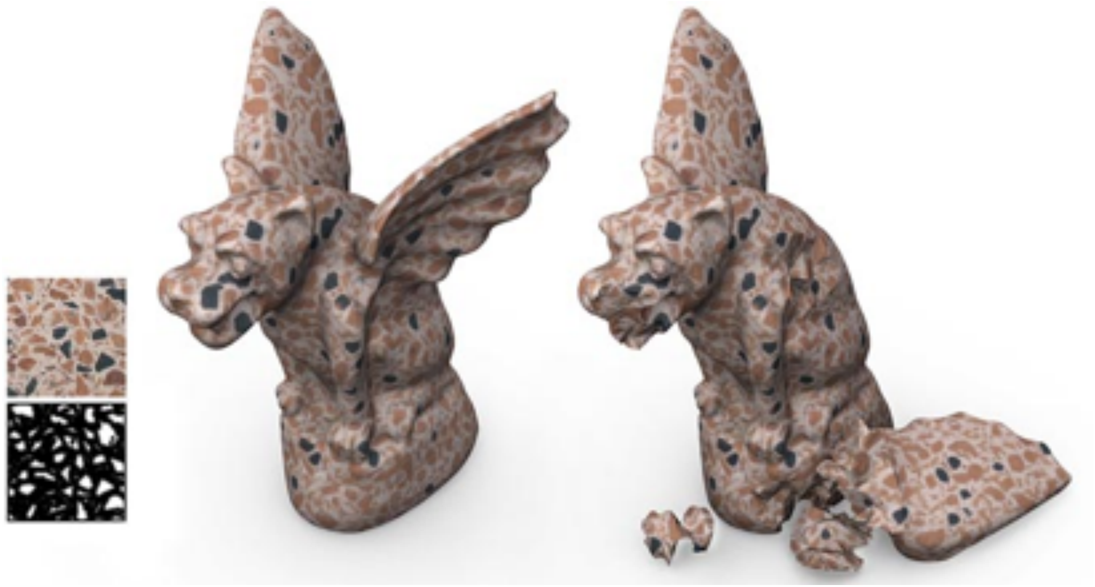




32 parametric patches



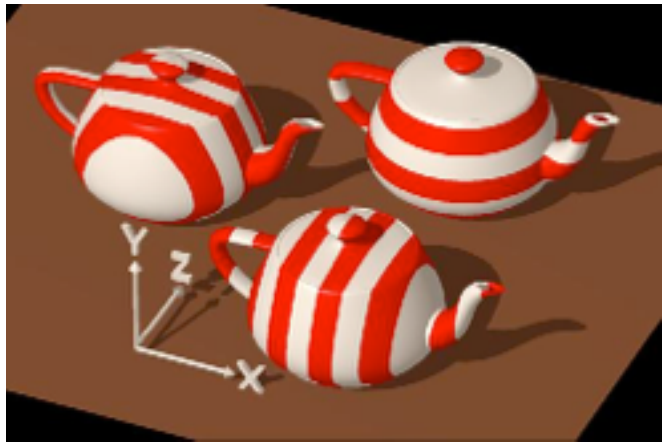
3D solid textures

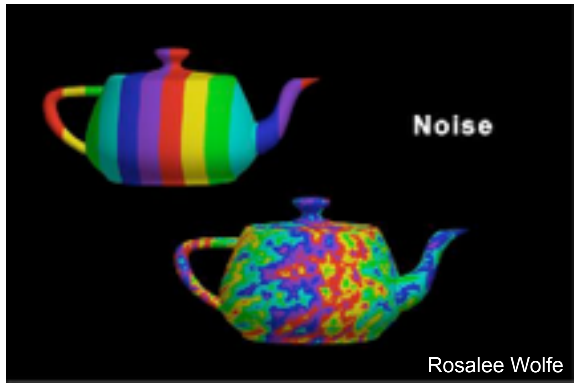


can map object (x,y,z) directly to texture (u,v,w)

[Dong et al., 2008]

Procedural textures





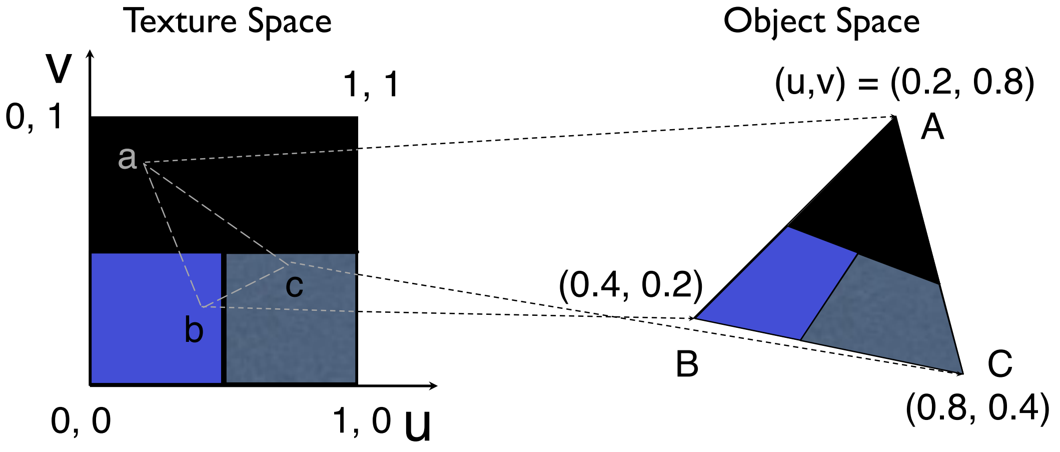
Rosalee Wolfe

e.g., Perlin noise

Triangles

Texturing triangles

- Store (u,v) at each vertex
- interpolate inside triangles using barycentric coordinates



[Angel and Shreiner]

Texturing triangles

- Store (u,v) at each vertex
- interpolate inside triangles using barycentric coordinates

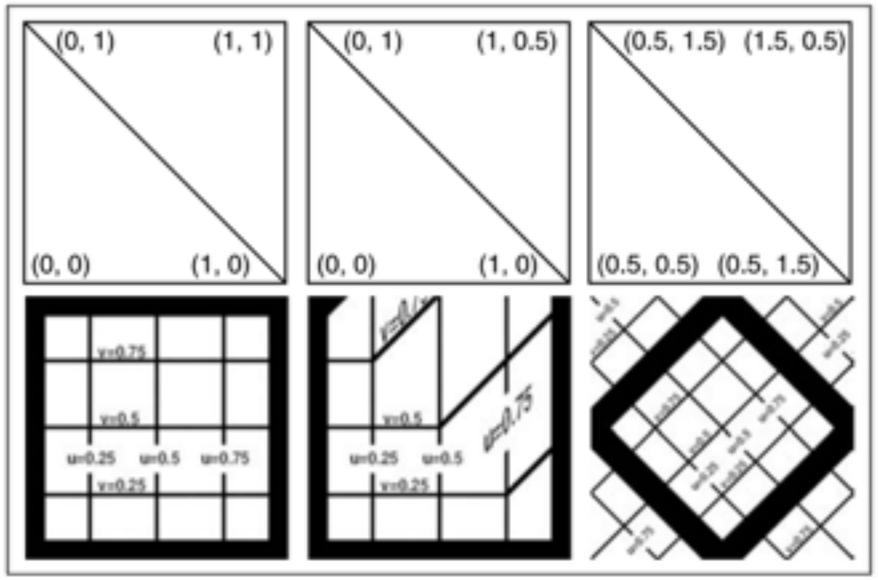
$$\mathbf{p}(\beta,\gamma) = \mathbf{a} + \beta(\mathbf{b}-\mathbf{a}) + \gamma(\mathbf{c}-\mathbf{a}).$$

$$u(\beta,\gamma) = u_a + \beta(u_b - u_a) + \gamma(u_c - u_a),$$

$$v(\beta,\gamma) = v_a + \beta(v_b - v_a) + \gamma(v_c - v_a).$$

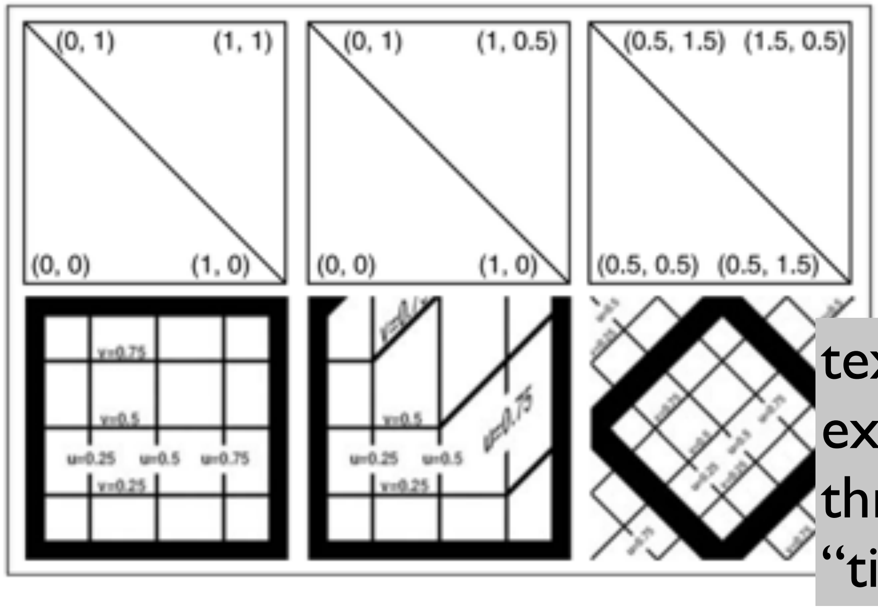
Texturing triangles

Choice of (u,v) makes big difference



Texturing triangles

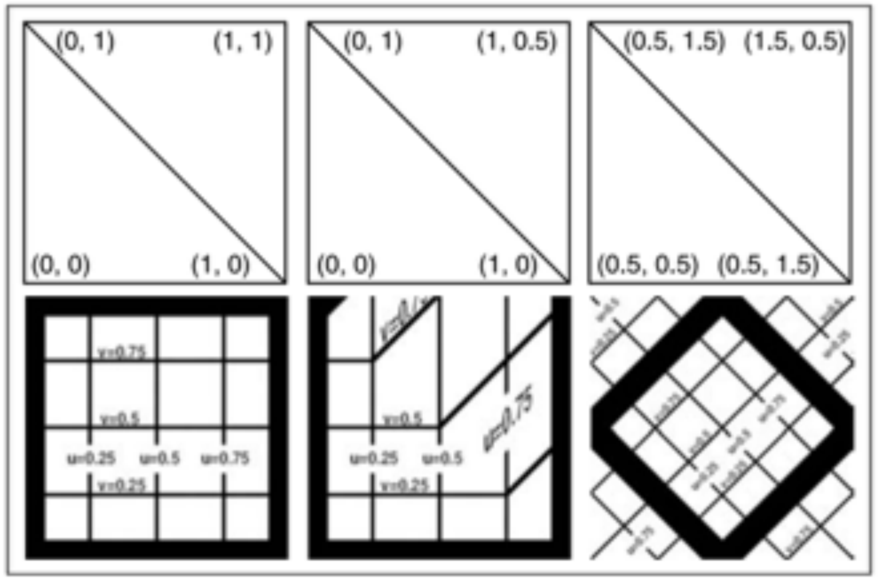
Choice of (u,v) makes big difference



texture extended through “tiling”

Texturing triangles

Choice of (u,v) makes big difference

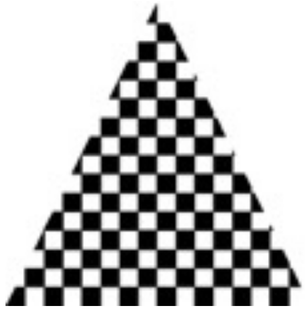


Textures in OpenGL

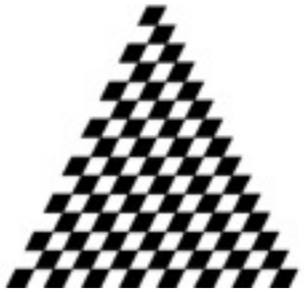
- Assign (u,v) to vertices

```
glTexCoord*()
```

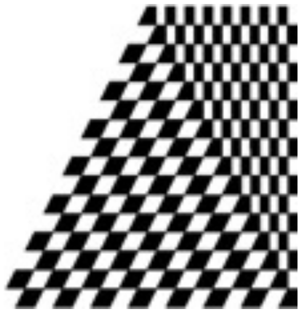
- OpenGL then uses interpolation for triangle interior



good selection
of tex coordinates

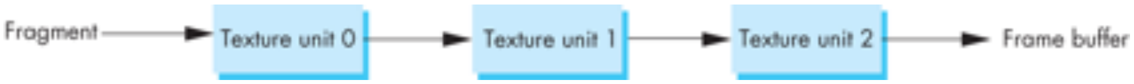


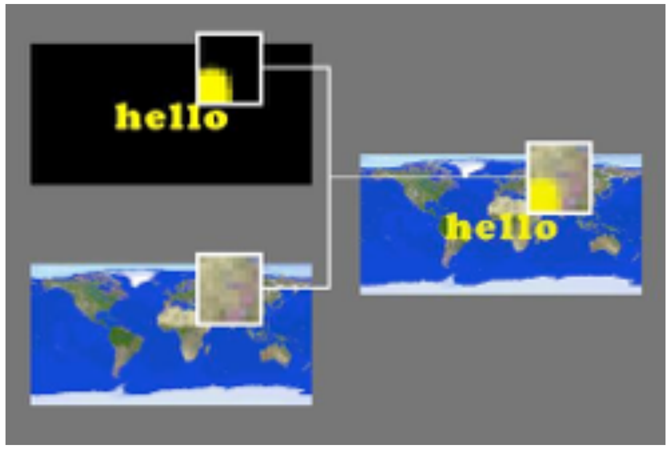
poor selection
of tex coordinates

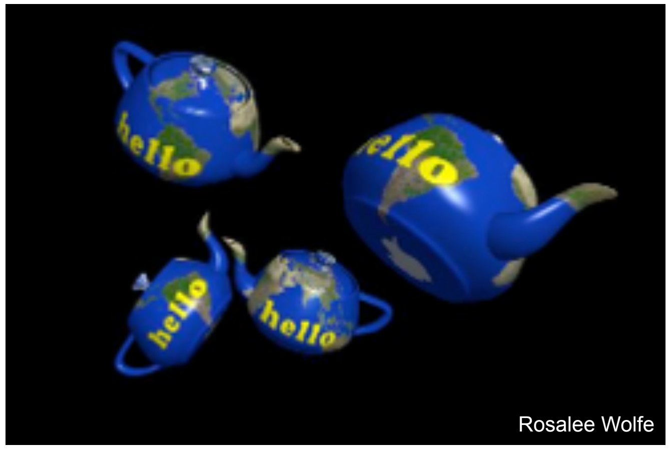


texture stretched
over trapezoid
showing effects of
bilinear interpolation

Multitexturing

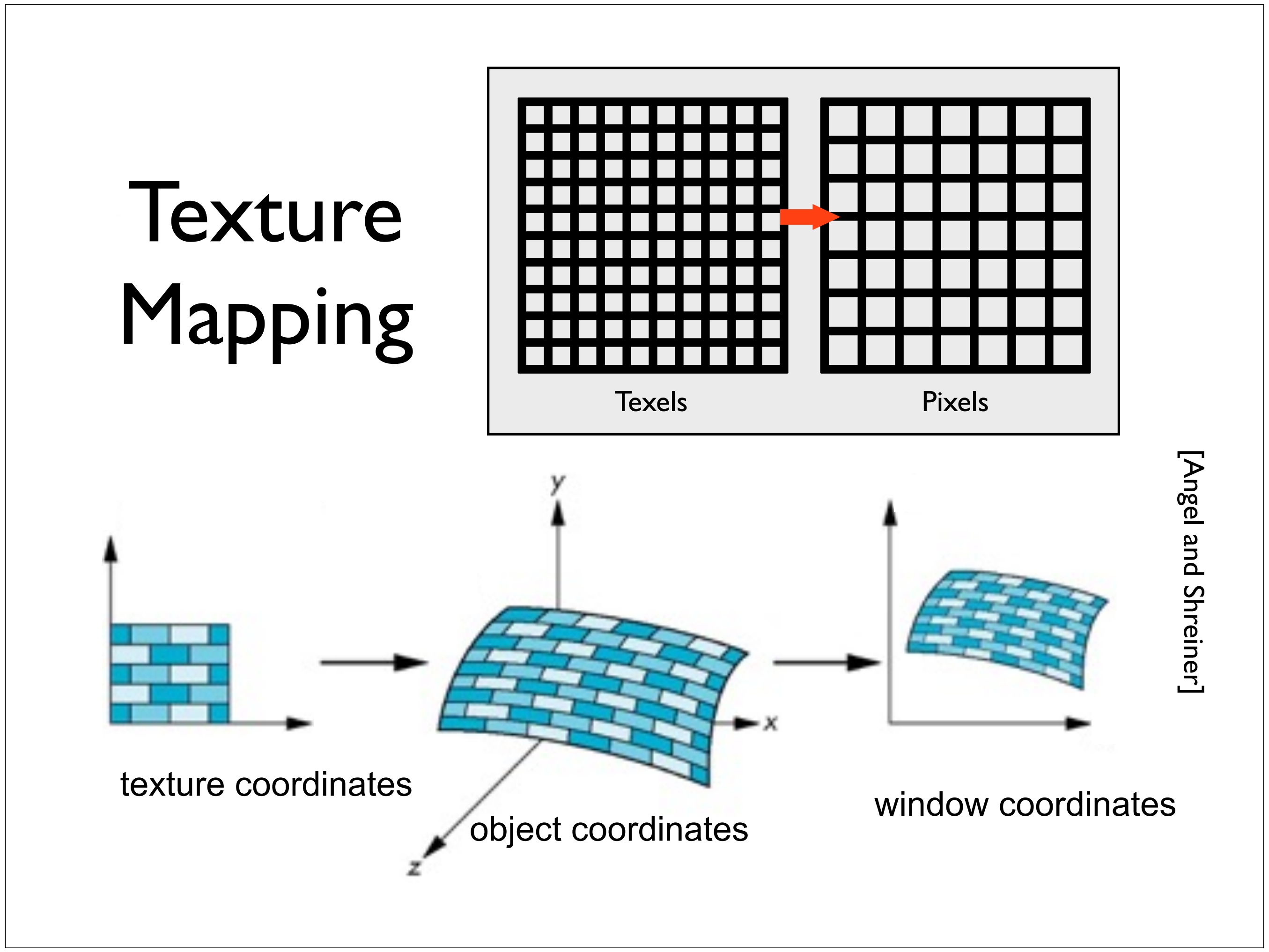






Rosalee Wolfe

Texture Sampling



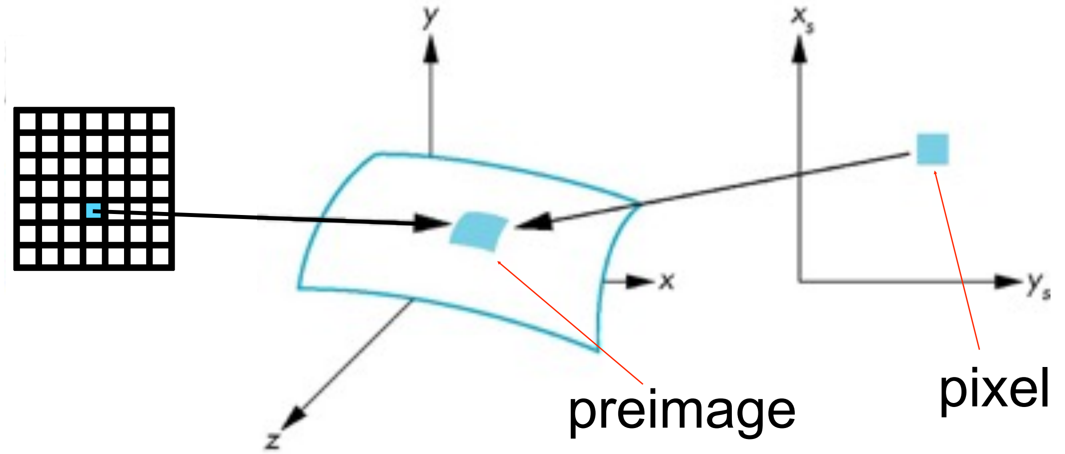
Texture Mapping

[Angel and Shreiner]

- Texture coordinates: Used to identify points in the image to be mapped
- Object Coordinates: Conceptually, where the mapping takes place
- Window Coordinates: Where the final image is really produced

Point Sampling

Map back to texture image and use the **nearest texel**



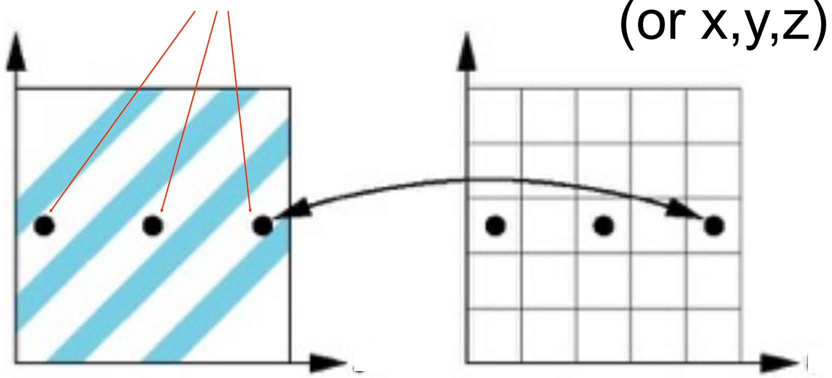
[Angel and Shreiner]

Aliasing

Point sampling of the texture can lead to aliasing artifacts

miss blue stripes

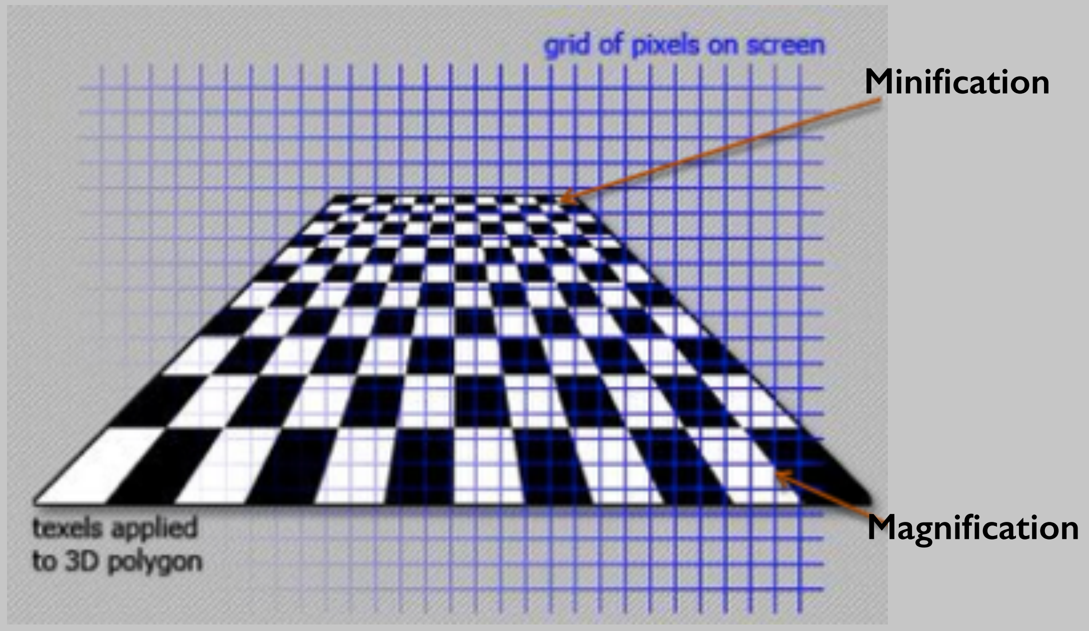
point samples in (or x,y,z) space



point samples in texture space

[Angel and Shreiner]

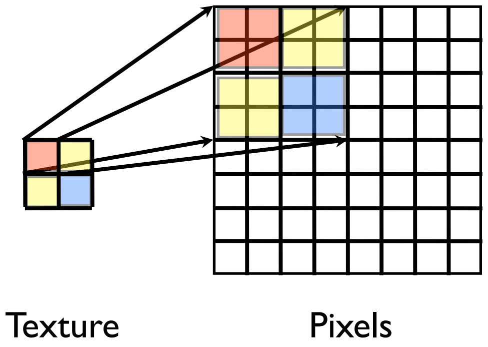
Magnification and Minification



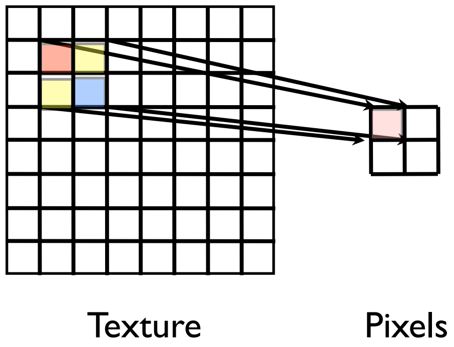
Magnification and Minification

More than one texel can cover a pixel (*minification*) or
more than one pixel can cover a texel (*magnification*)

Can use point sampling (nearest texel) or linear filtering
(2 x 2 filter) to obtain texture values

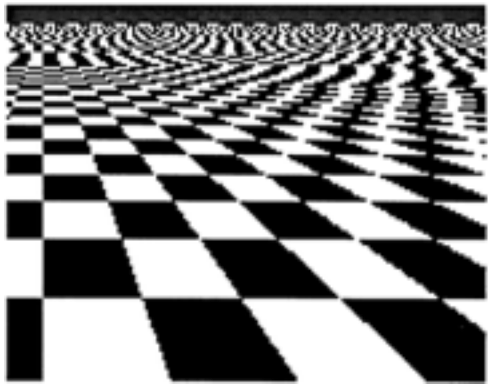


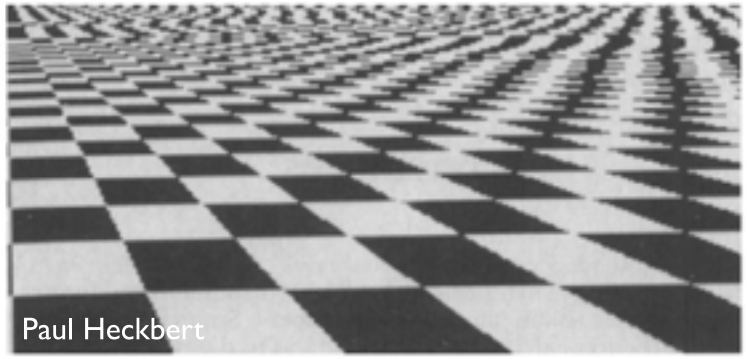
Magnification



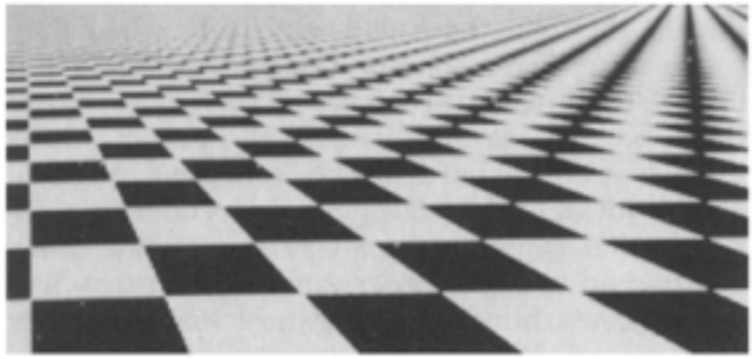
Minification

Aliasing artifacts





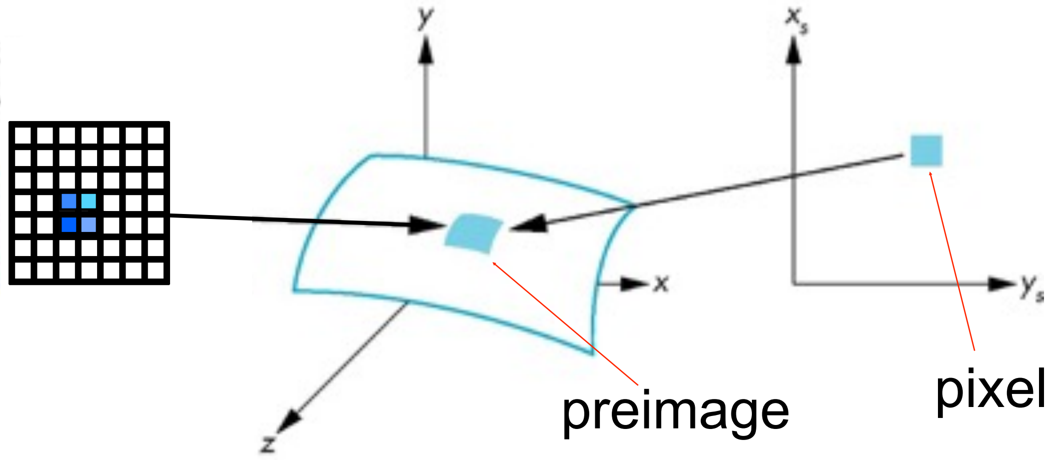
Paul Heckbert



We apply **filtering** to reduce aliasing artifacts

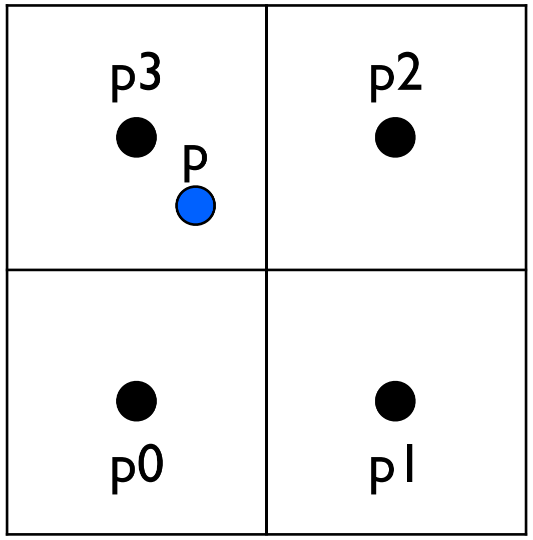
Area Averaging

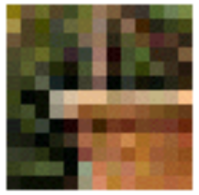
A better but slower option is to use **area averaging**



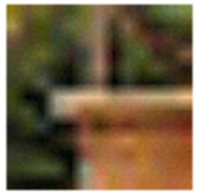
[Angel and Shreiner]

Use bilinear filtering

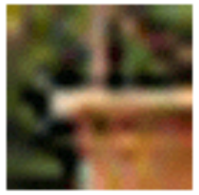




nearest neighbor



bilinear



Wikipedia

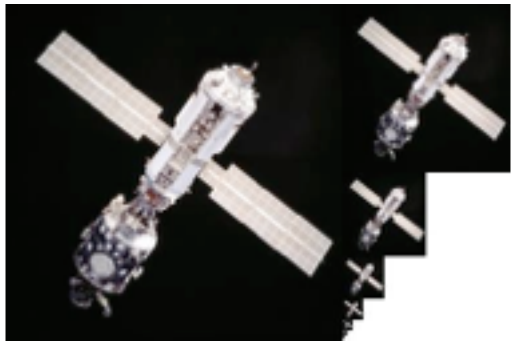
bicubic

mitigate magnification artifacts

p = ?

smooths out the texture – no sharp boundaries

Mipmapping



Togikun, Wikimedia Commons

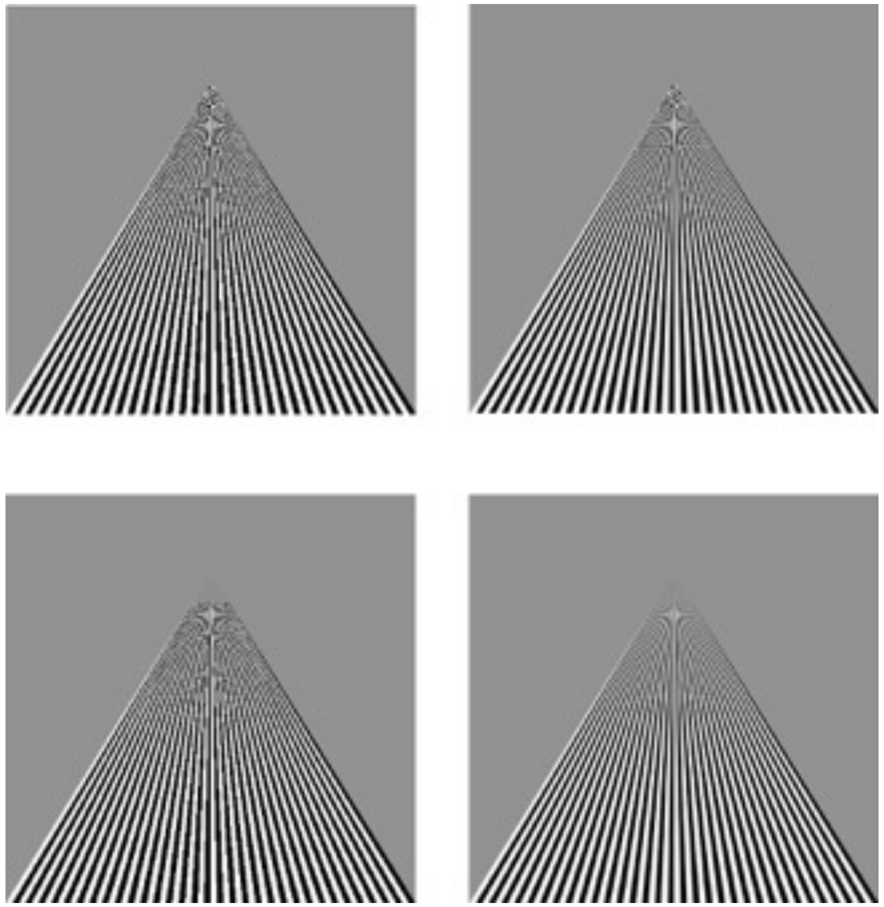
128×128, 64×64, 32×32, 16×16, 8×8, 4×4, 2×2, 1×1

Reduce minification artifacts

Prefilter the texture to obtain reduced resolutions

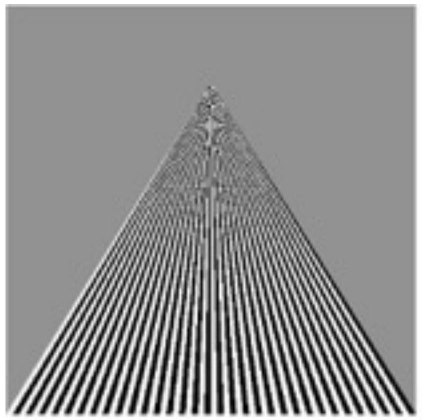
Requires 1/3 more space

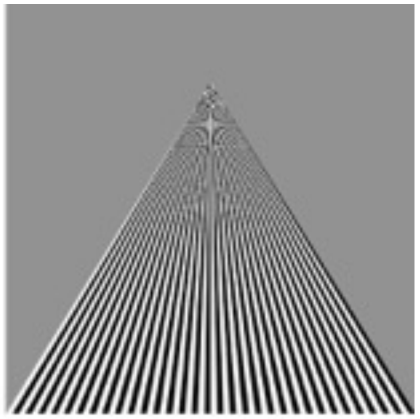
Get a texture hierarchy indexed by level



[Angel and Shreiner]

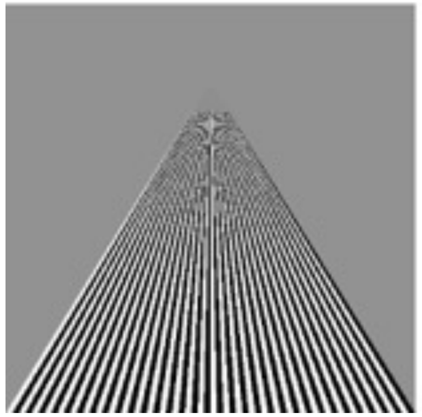
point
sampling

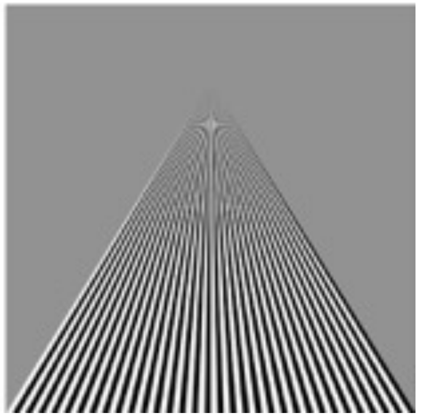




linear
filtering

mipmapped
point
sampling





mipmapped
linear
filtering

[Angel and Shreiner]