Texture Mapping

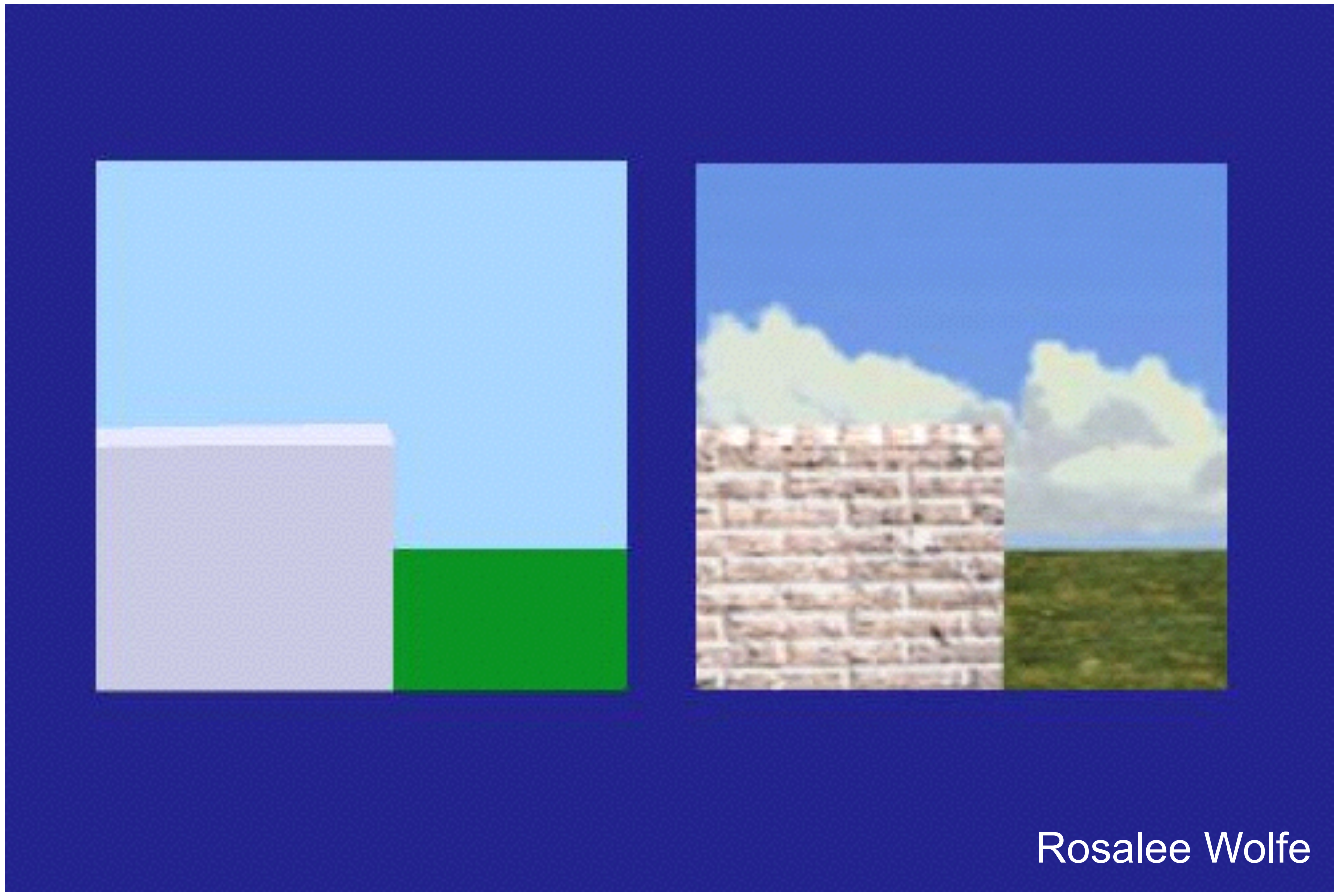
There are limits to geometric modeling

http://www.beinteriordecorator.com

National Geographic

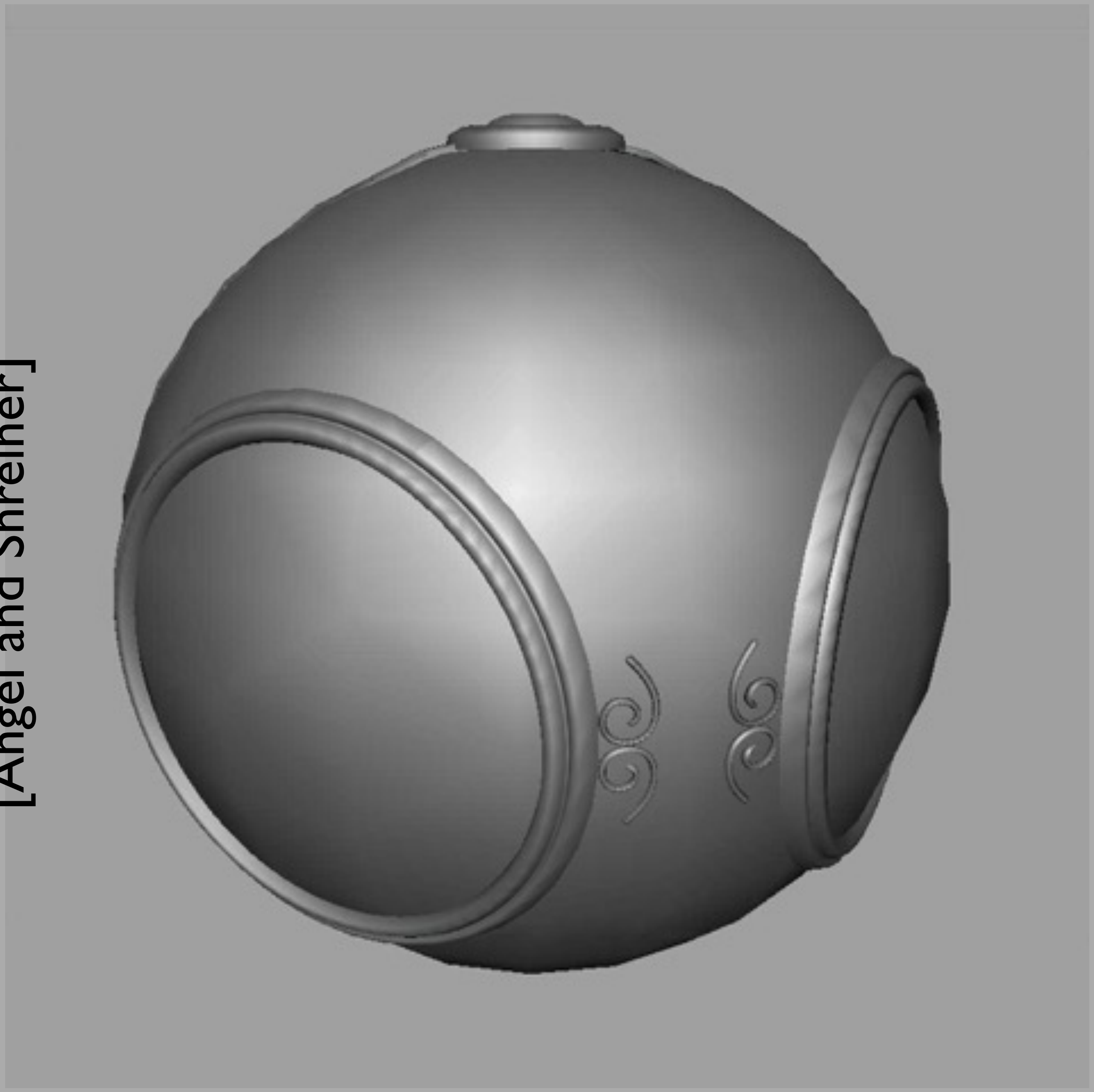
Although modern GPUs can render millions of triangles/sec, that's not enough sometimes...

Use texture mapping to increase realism through detail

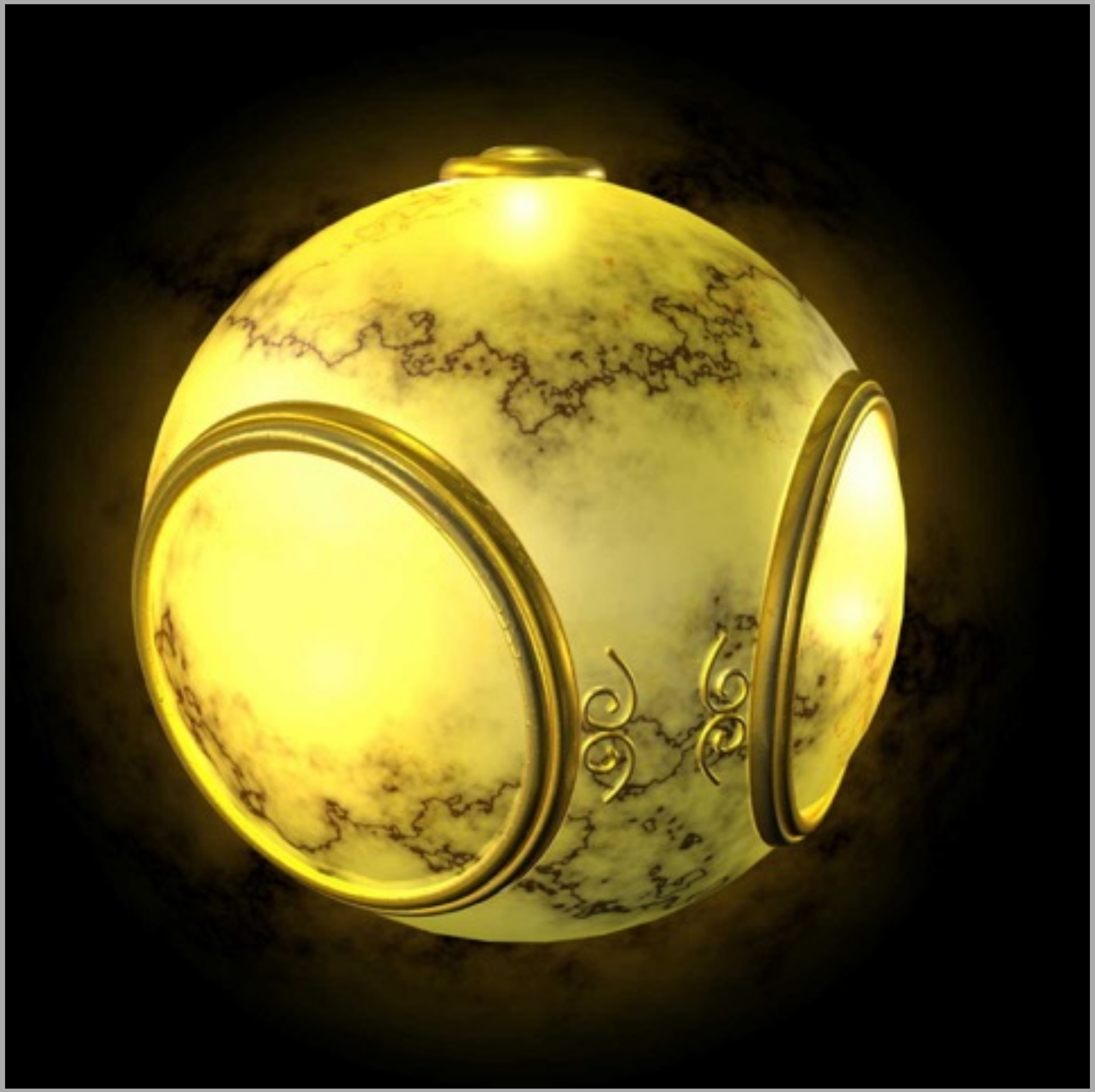


This image is just 8 polygons!

[Angel and Shreiner]



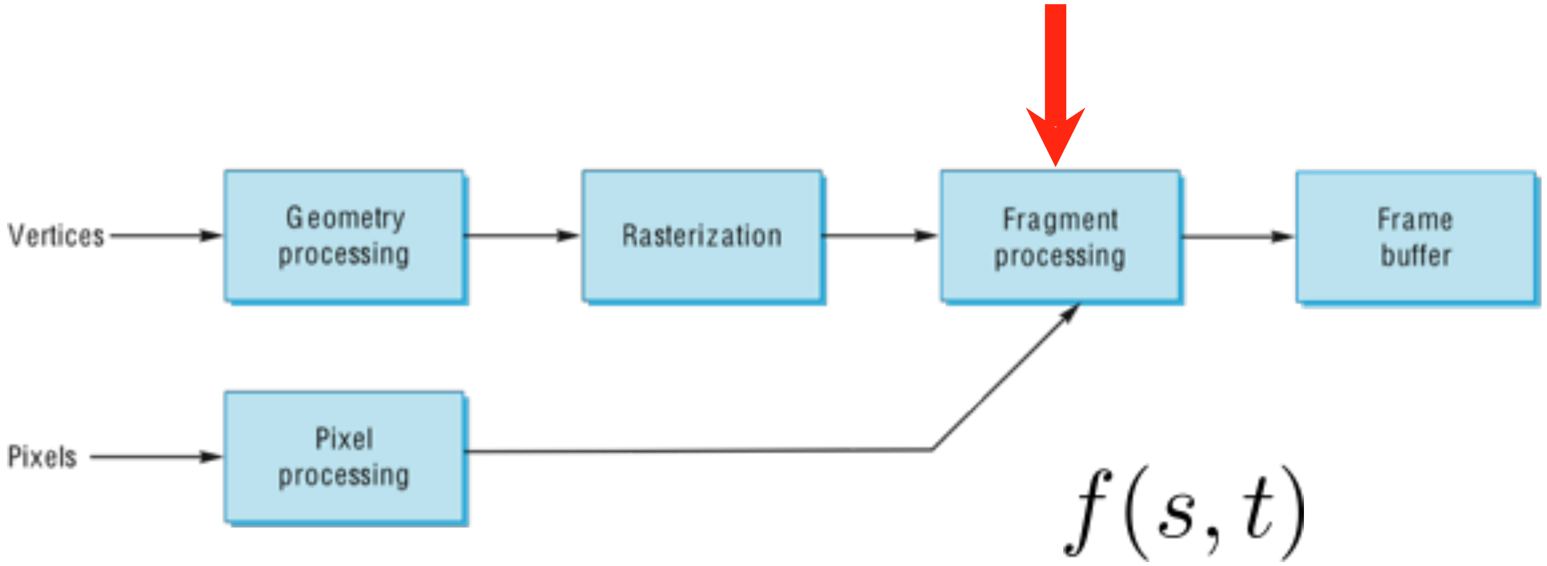
No texture



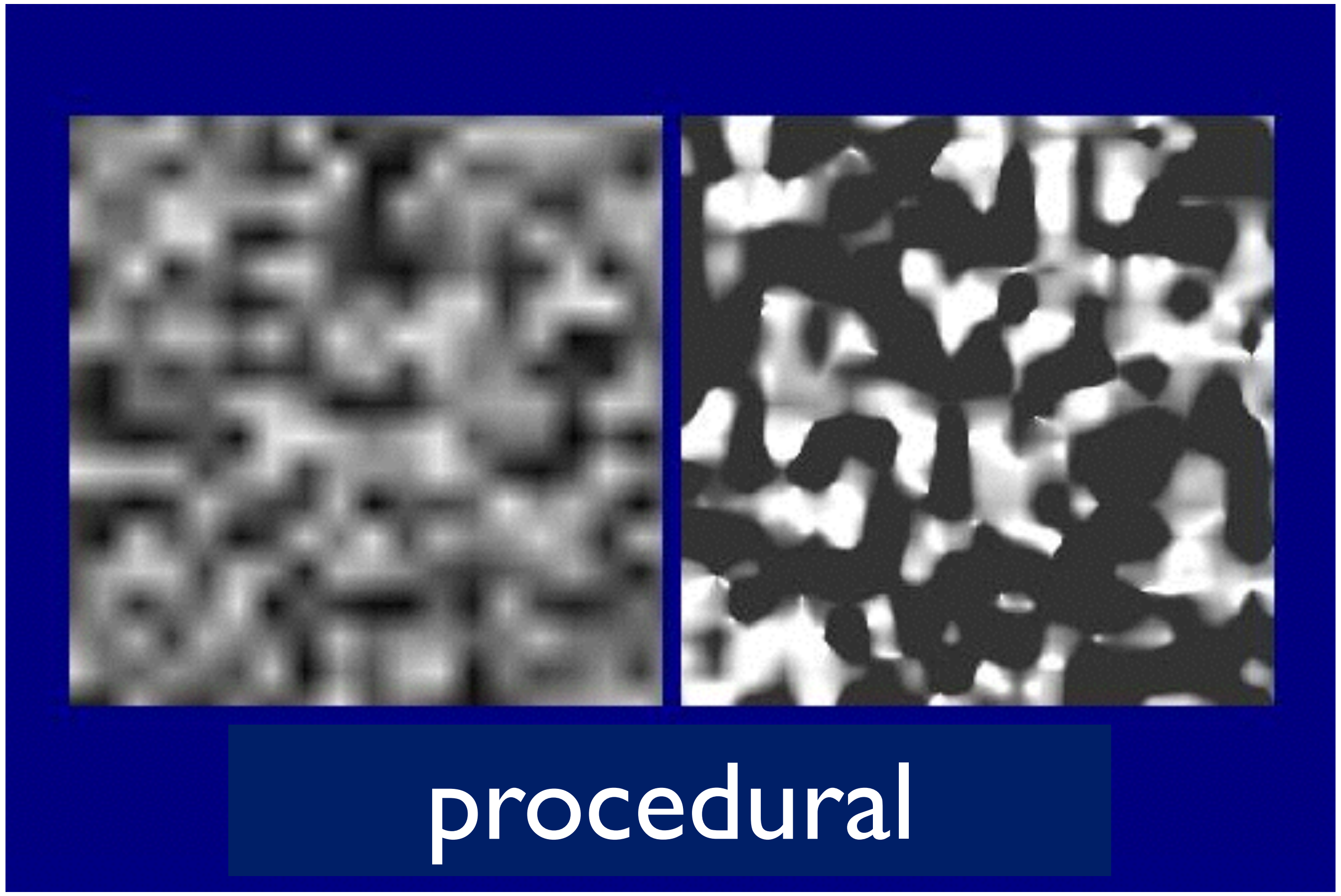
With texture

Pixar - Toy Story

Store 2D images in buffers and lookup pixel reflectances



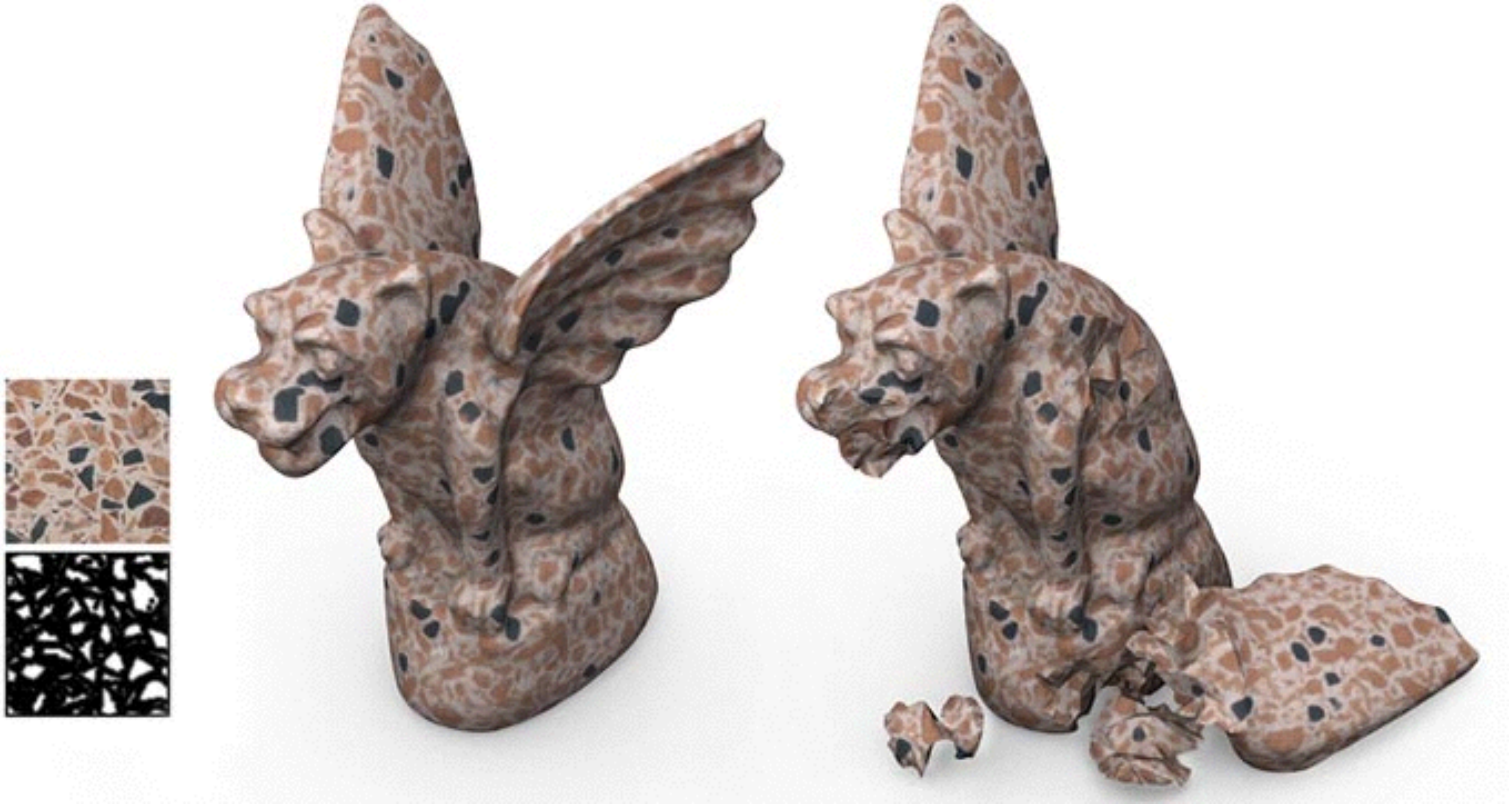
$f(s,t)$



procedural

photo

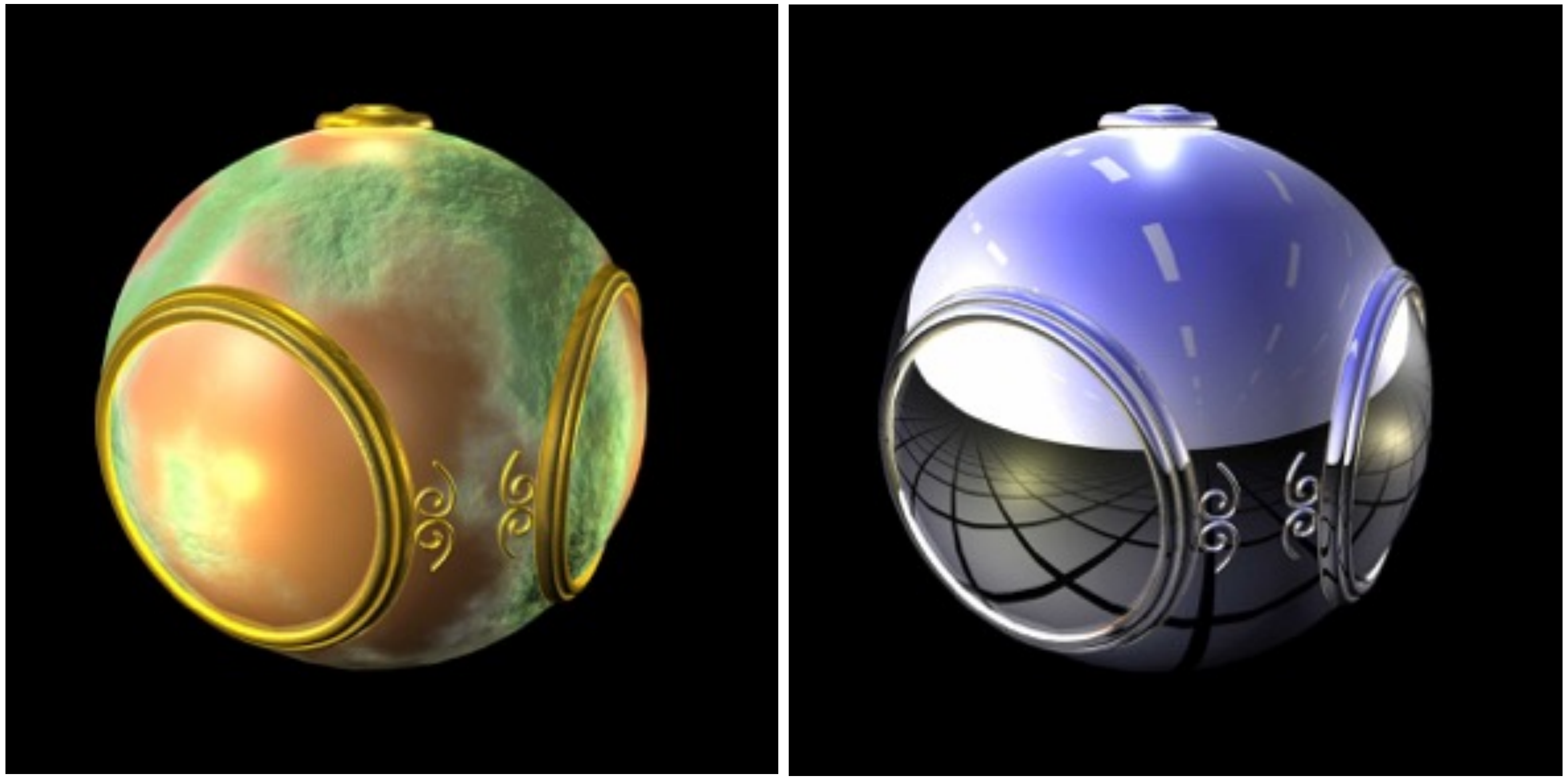
3D solid textures



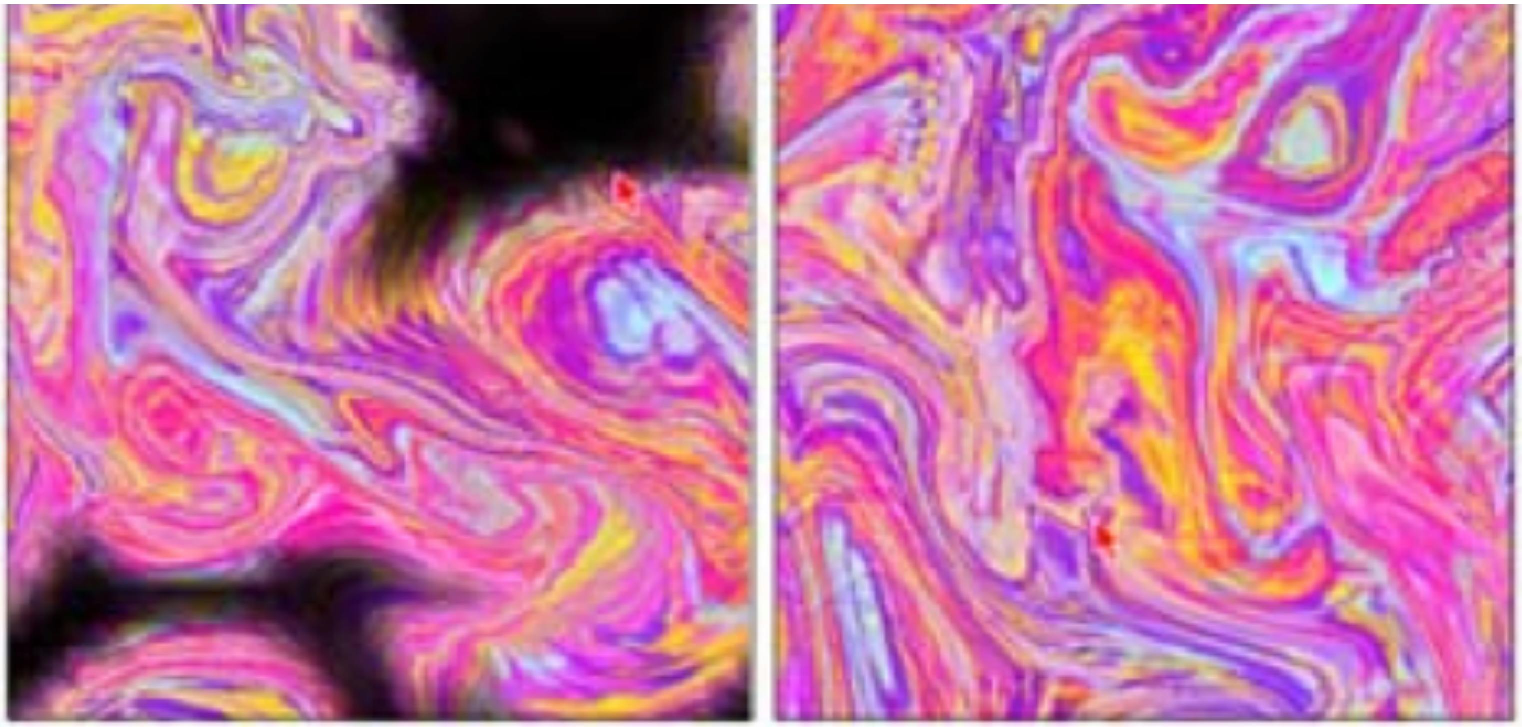
[Dong et al., 2008]

Other uses of textures...

Light maps
Shadow maps
Environment maps
Bump maps
Opacity maps
Animation

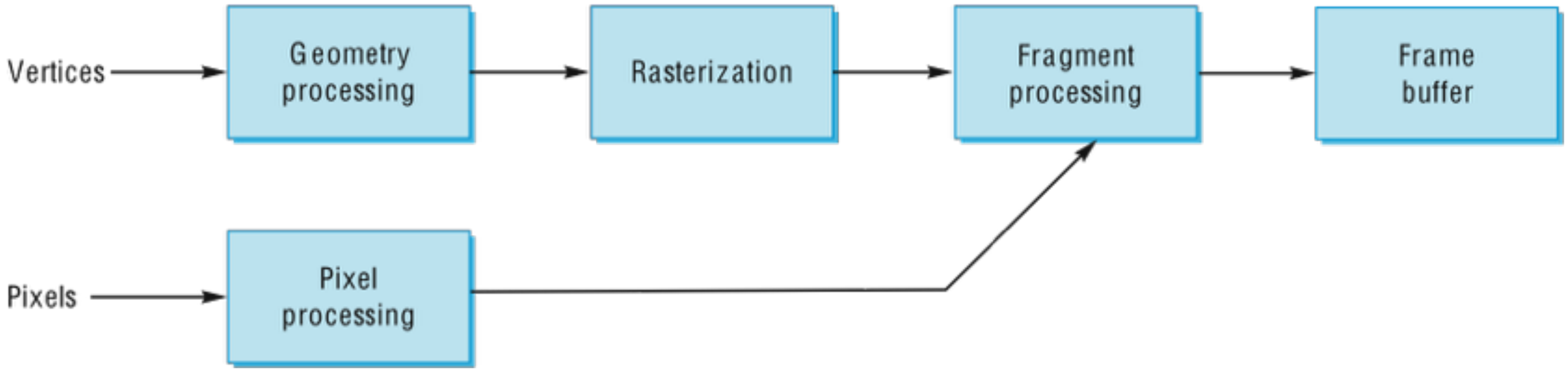


[Angel and Shreiner]



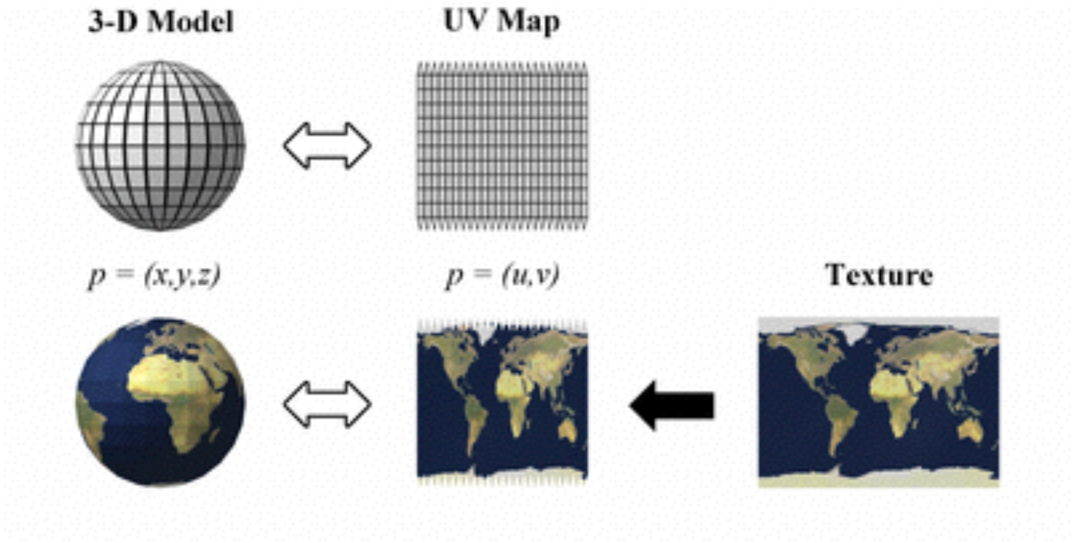
[Stam 99]

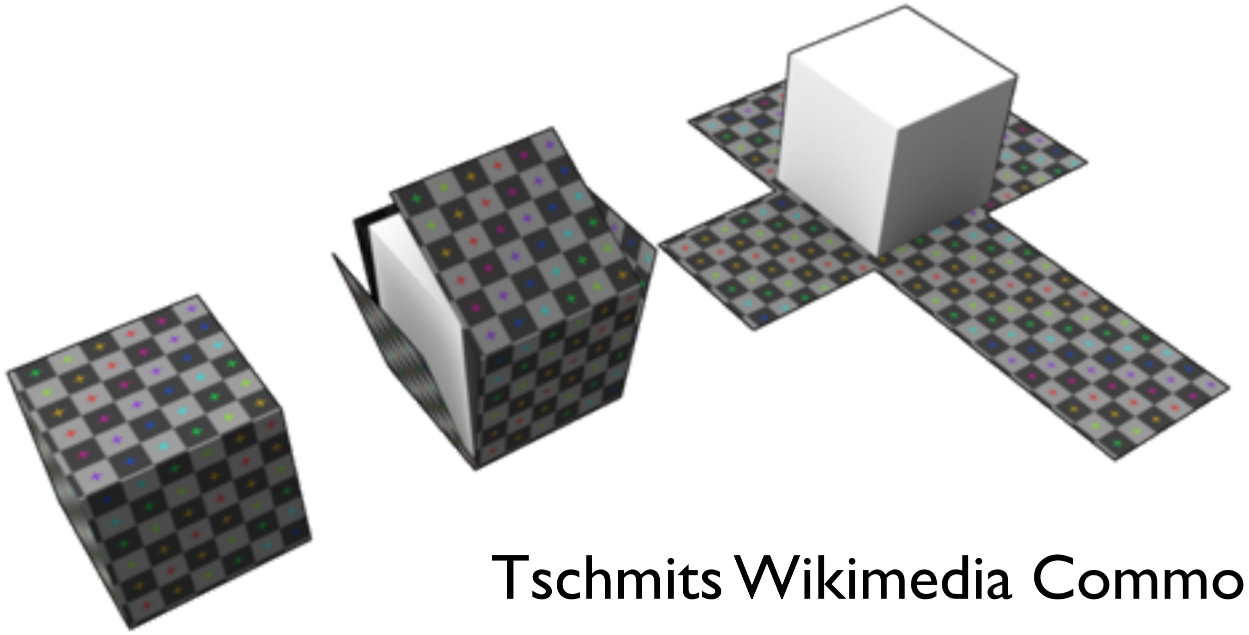
Texture mapping in the OpenGL pipeline



- 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

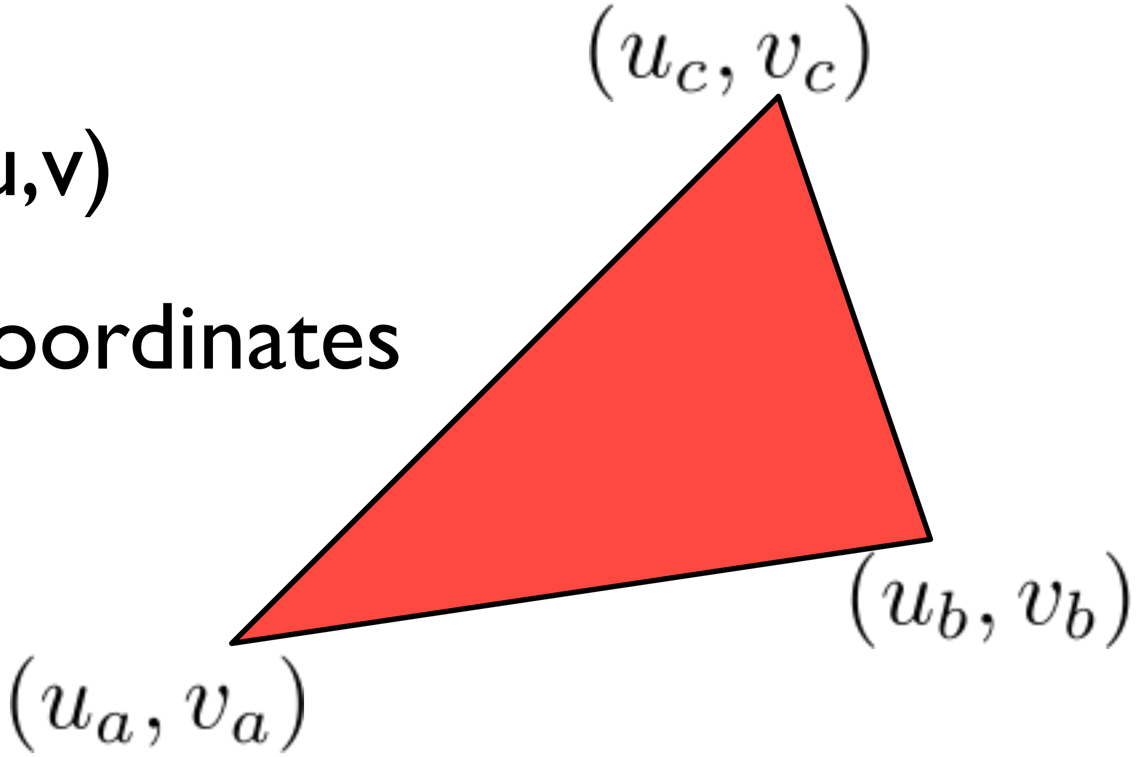
uv Mapping

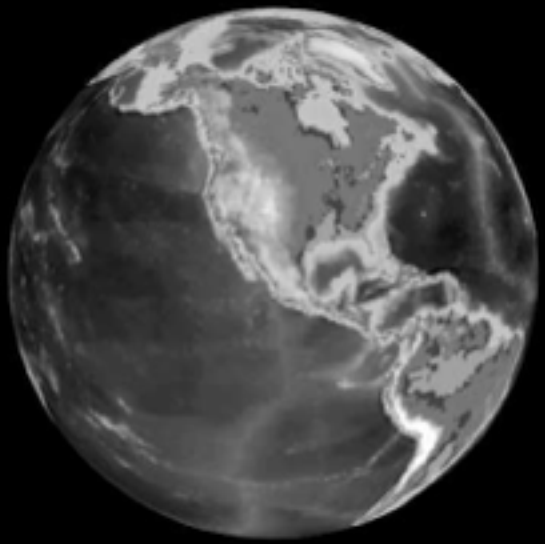


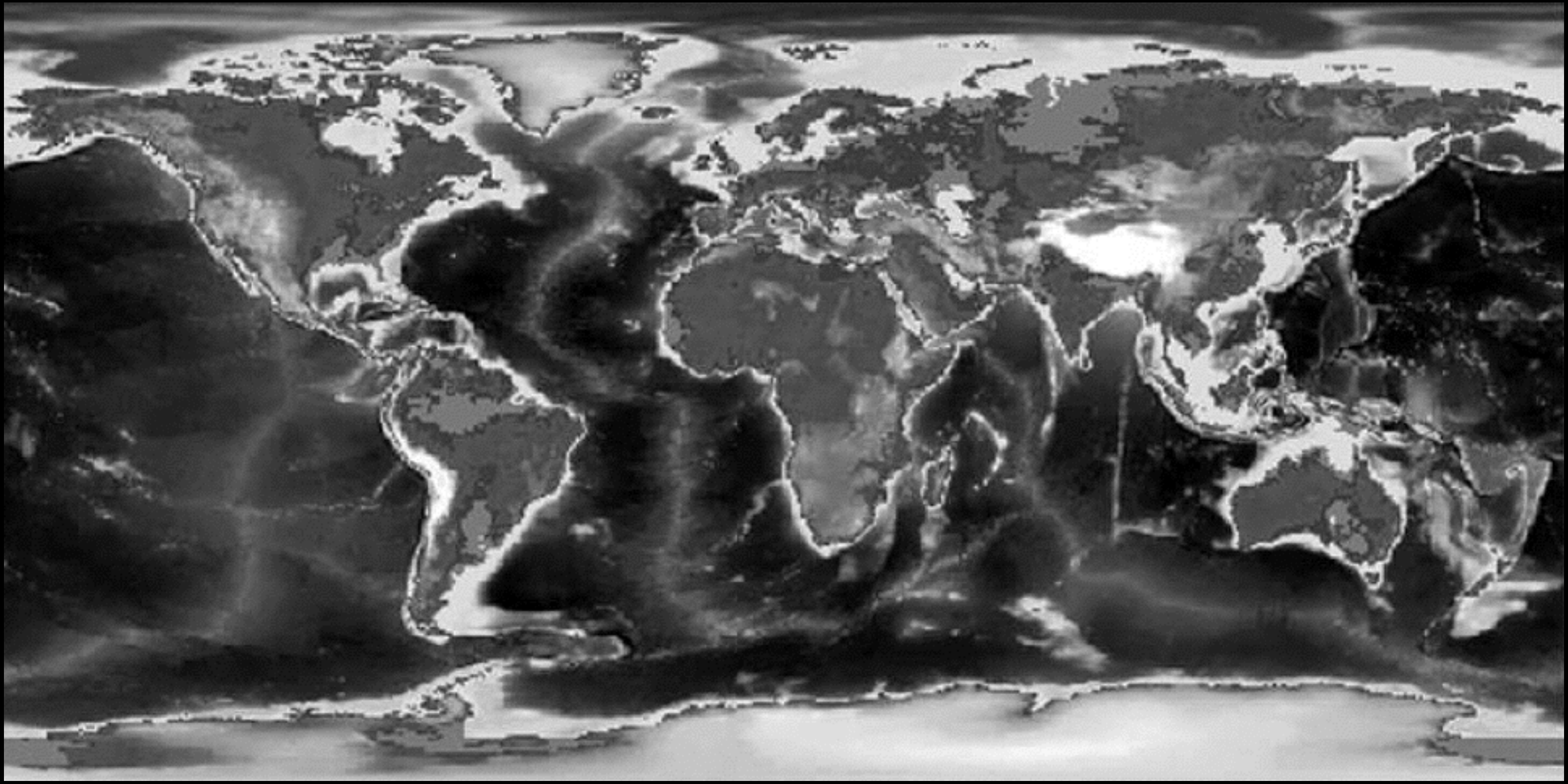


Tschmits Wikimedia Commons

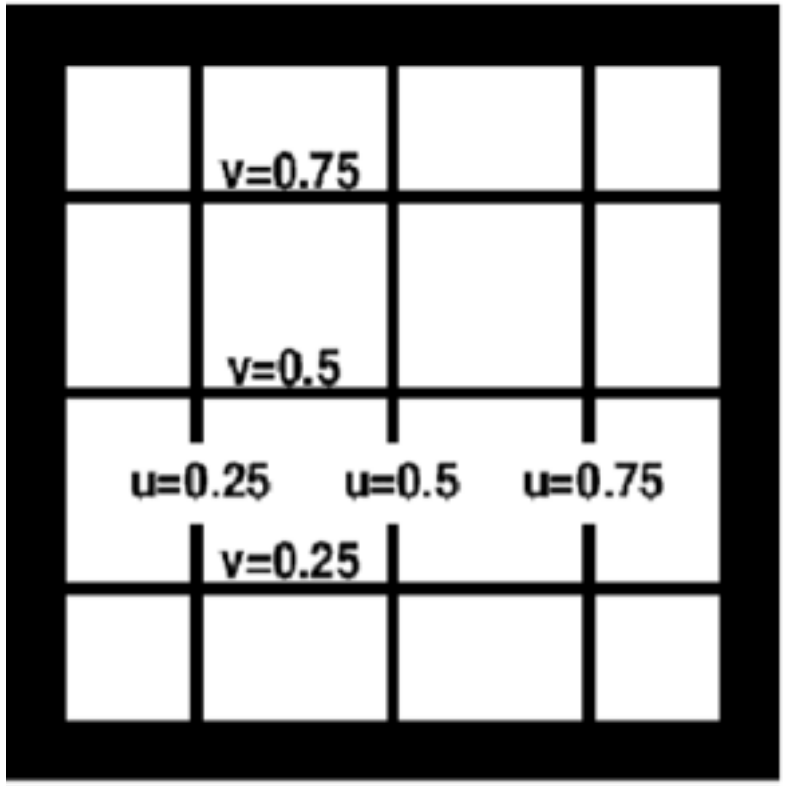
- 2D texture is parameterized by (u,v)
- Assign polygon vertices texture coordinates
- Interpolate within polygon

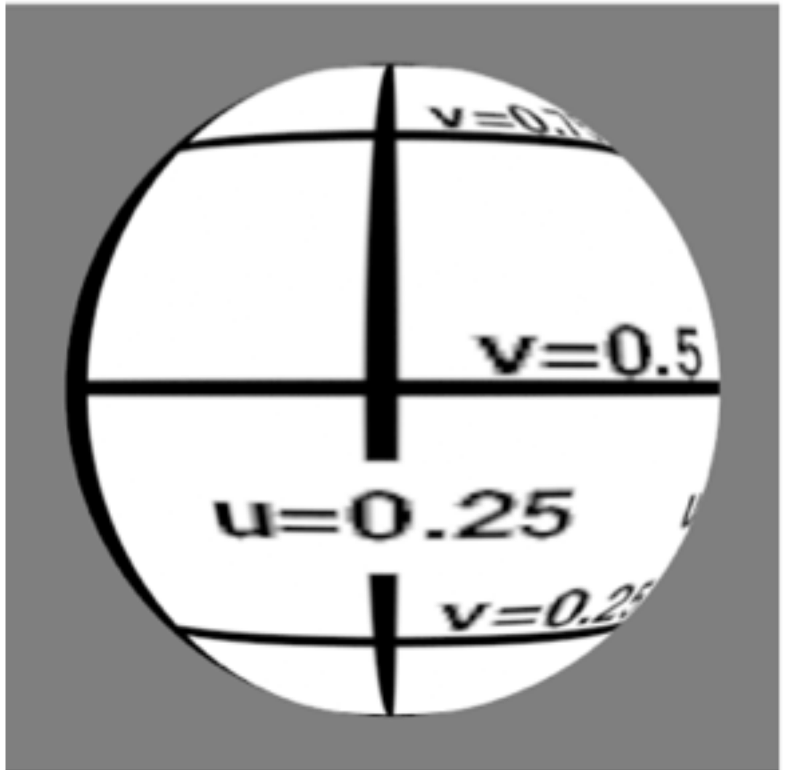


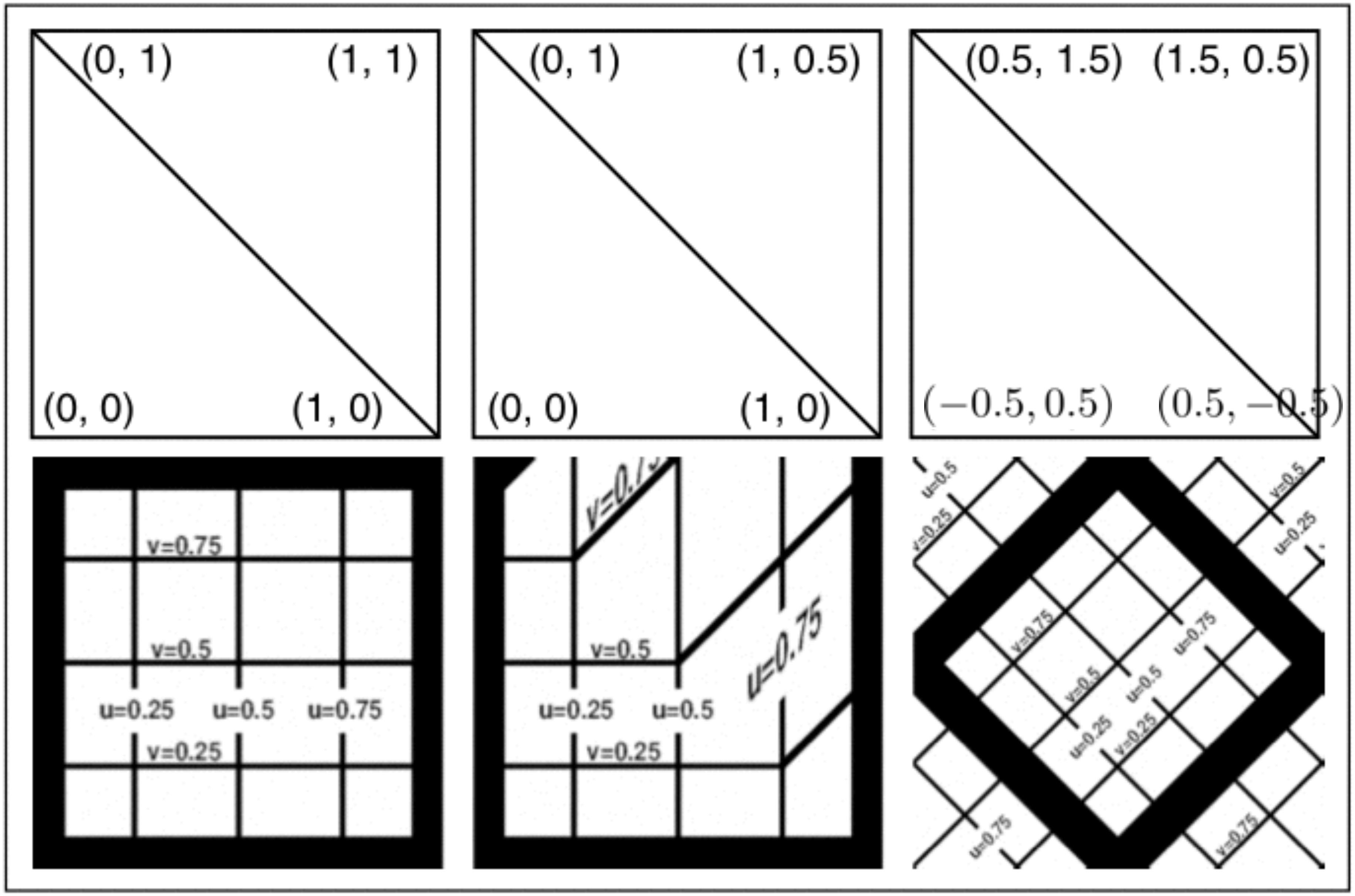




Texture Calibration

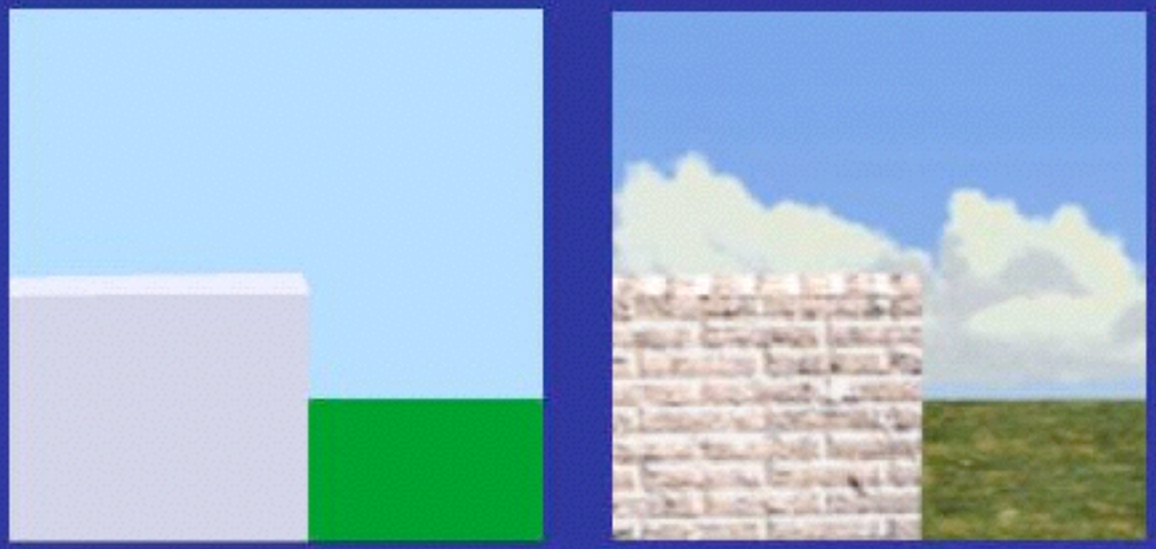




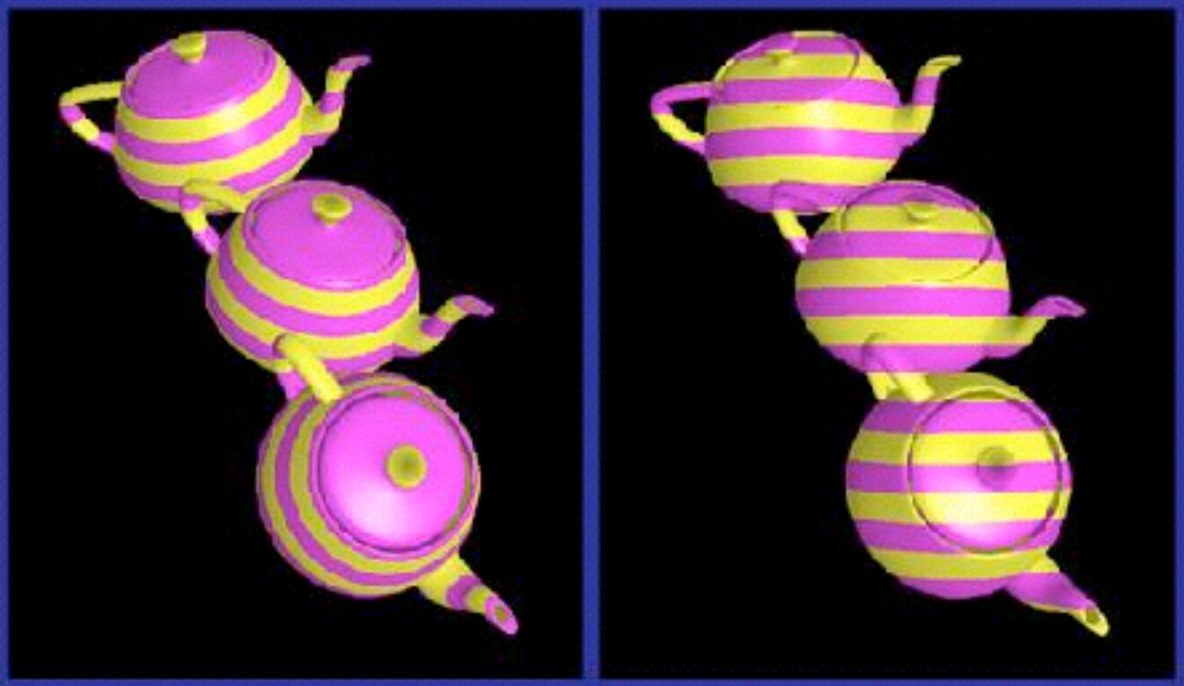


The major issues in texture mapping...

- What should the actual mapping be?



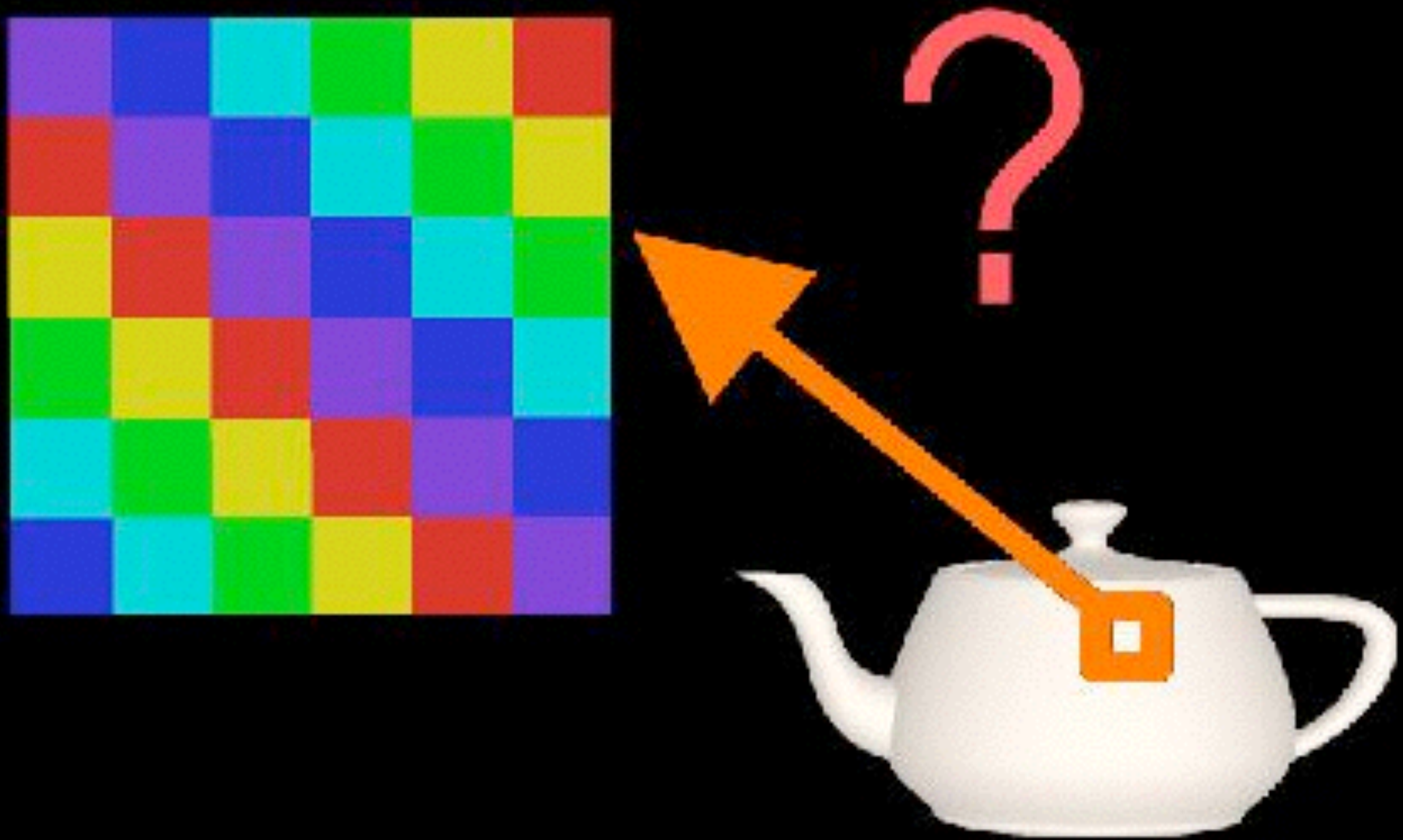
easy: rectangular surface



[Rosalee Wolfe]

harder: parametric surface

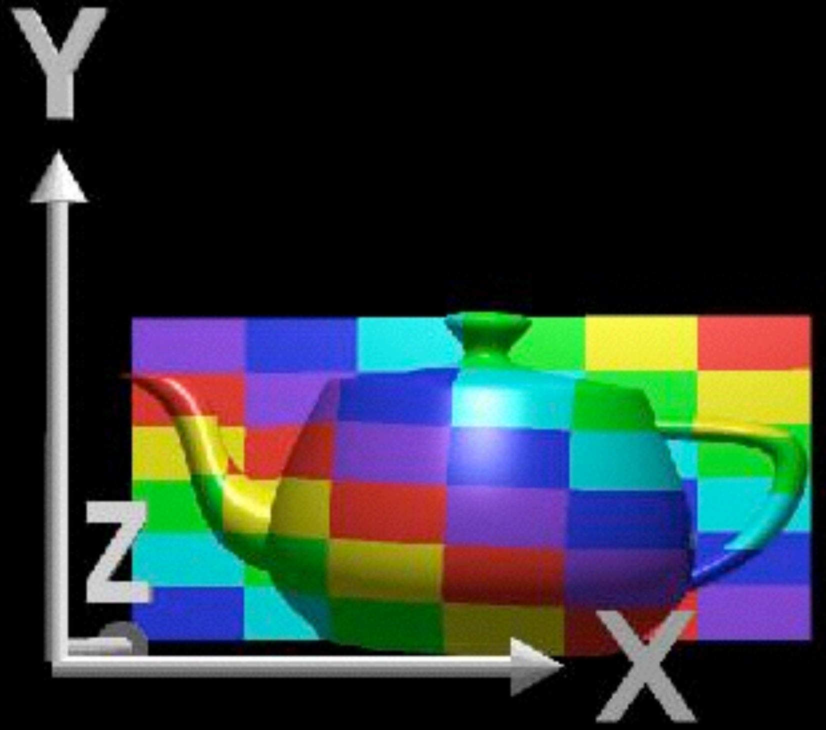
Given a point on the object **(x,y,z)**, what point **(u,v)** in the texture we use?

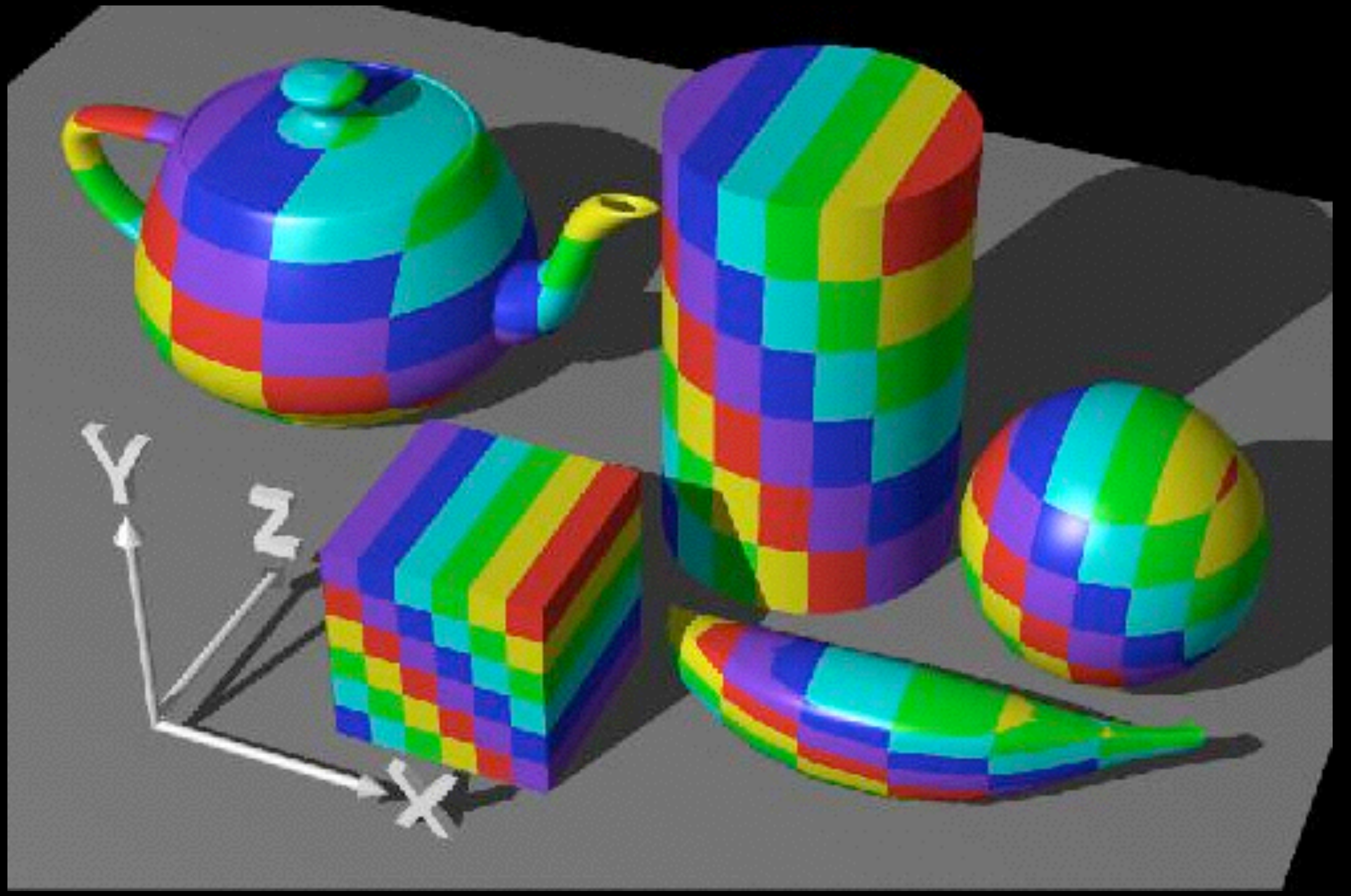


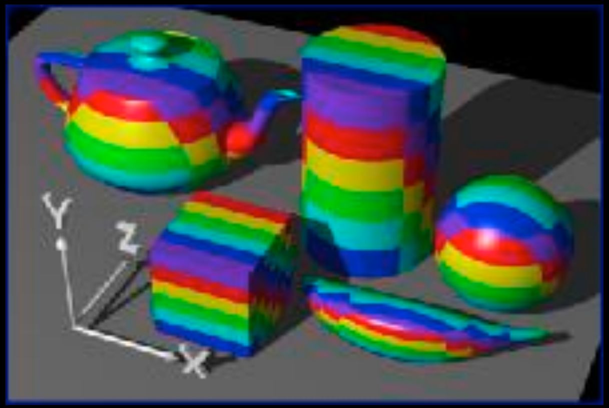
[Rosalee Wolfe]

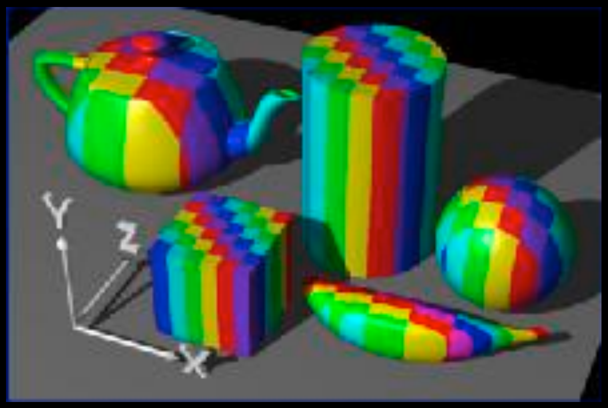
Example: planar mapping

[Rosalee Wolfe]



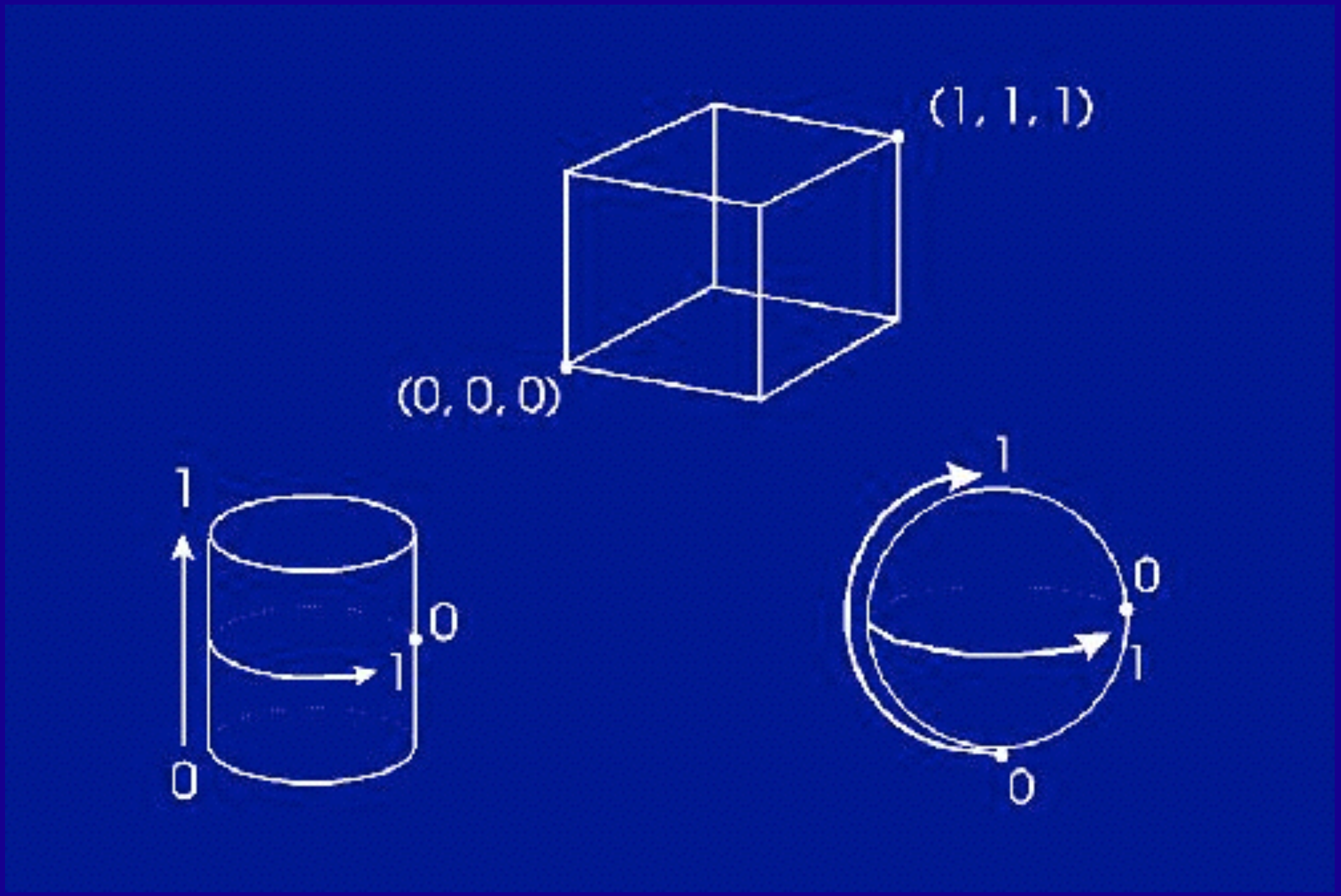






Intermediate surfaces

First map the texture to a simpler, intermediate surface

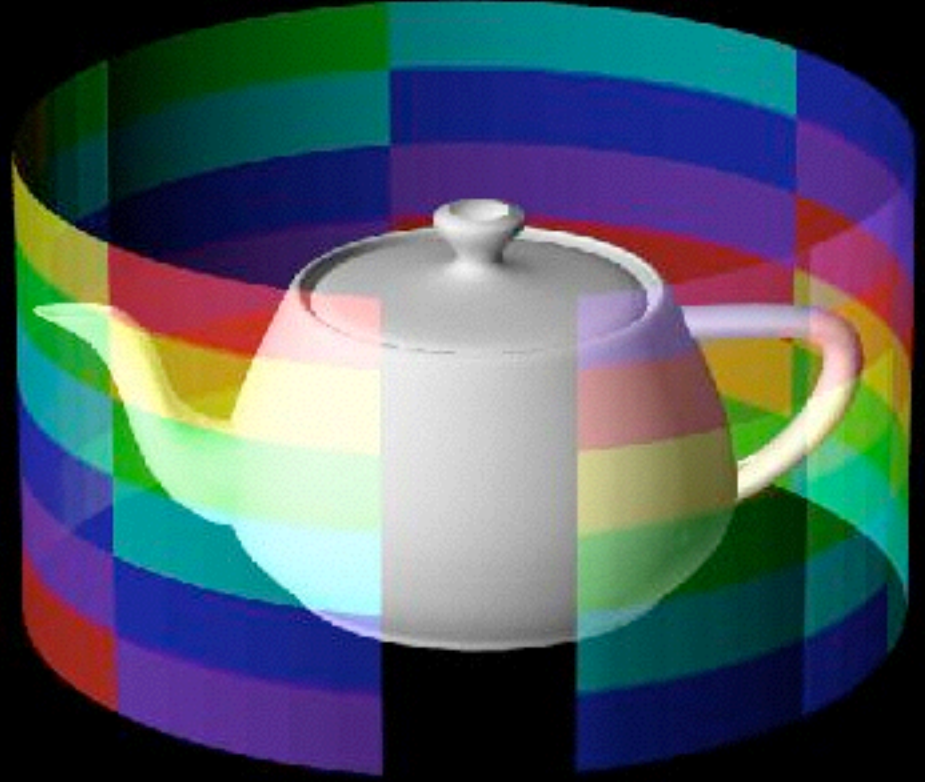


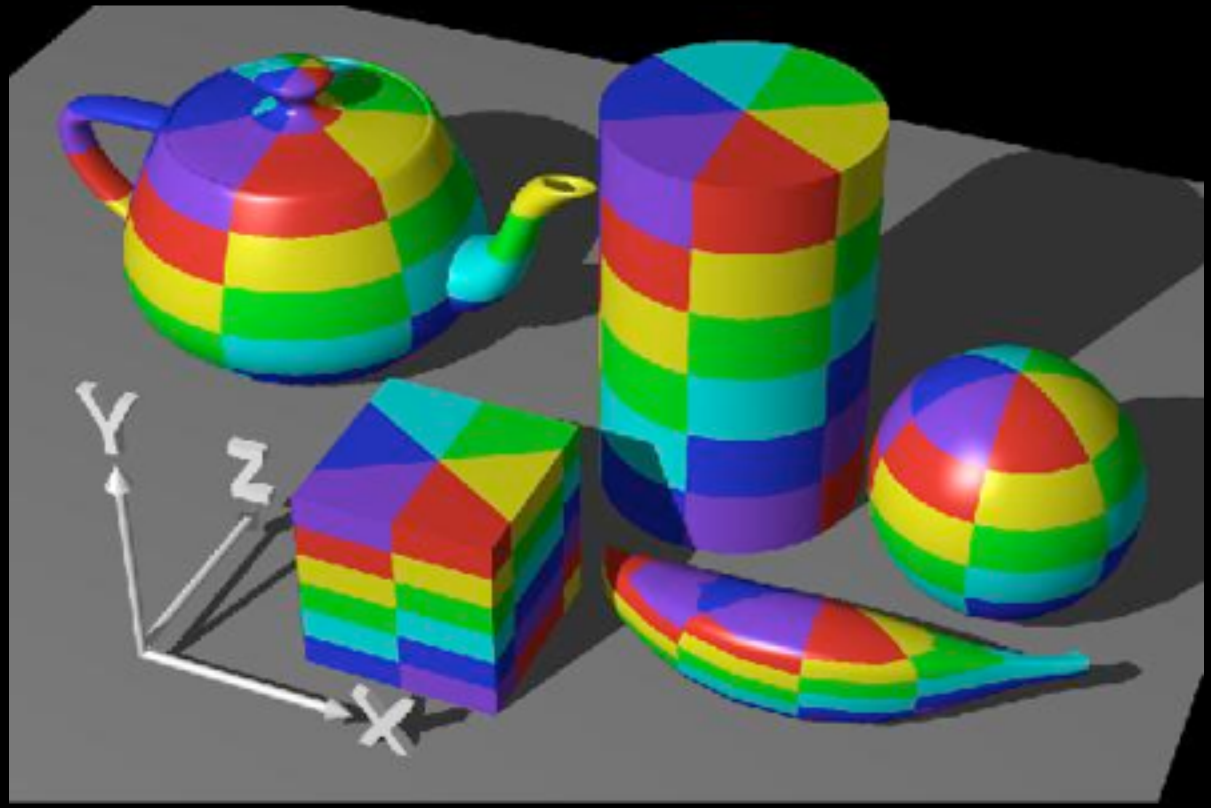
[Rosalee Wolfe]

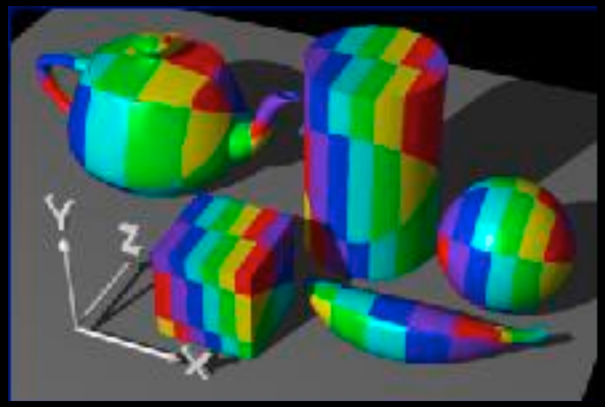
Cylindrical mapping

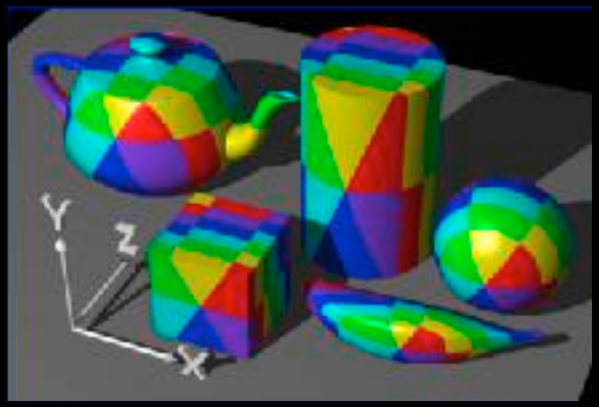
(x,y,z) -> (theta, h) -> (u,v)

[Rosalee Wolfe]





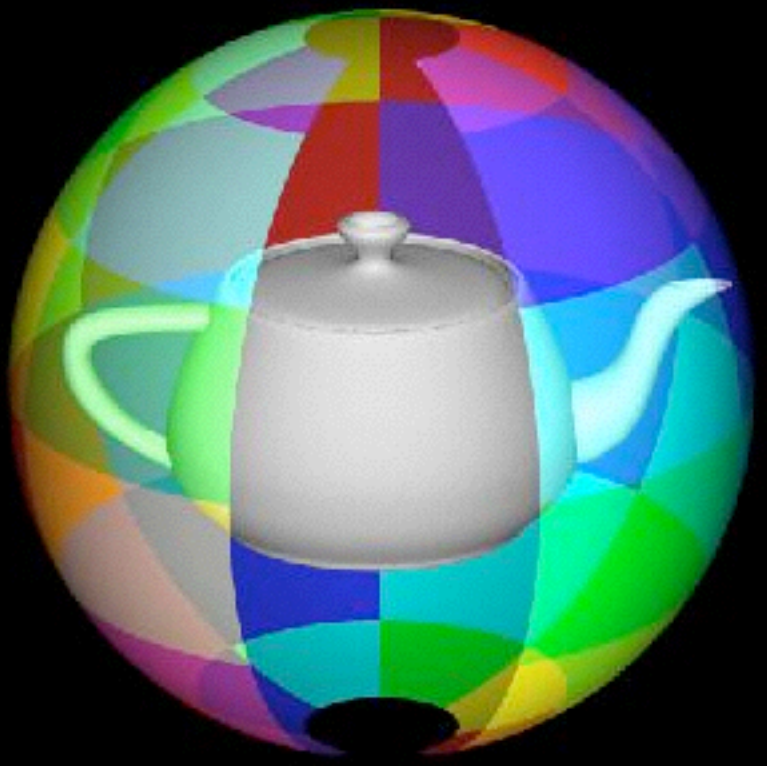


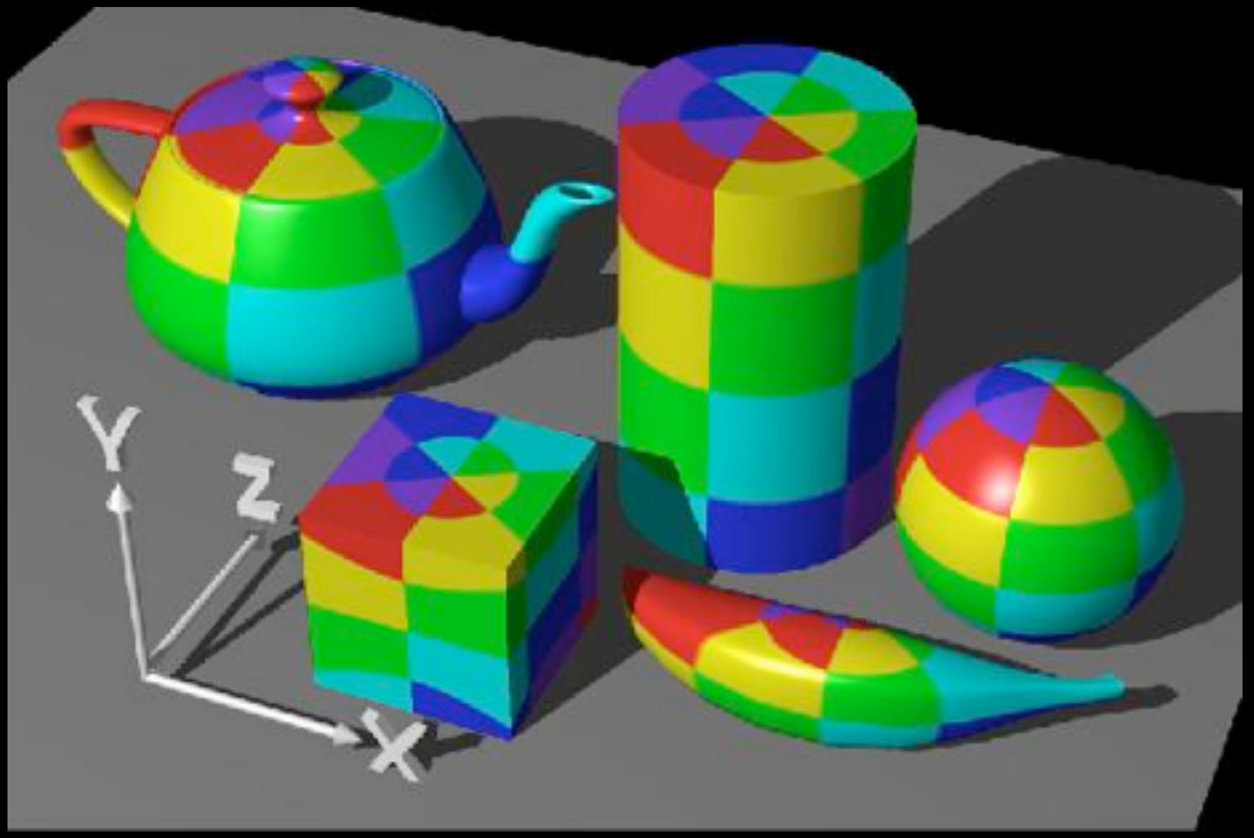


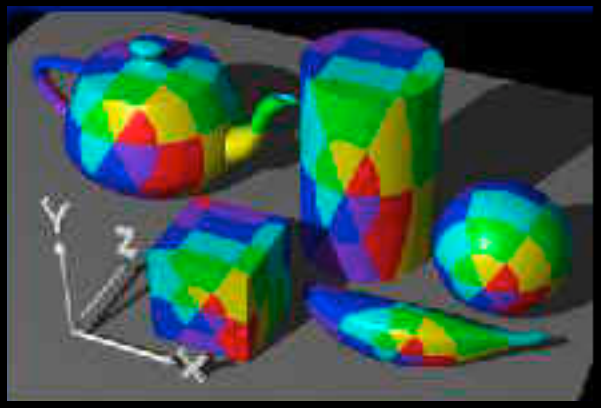
Spherical Mapping

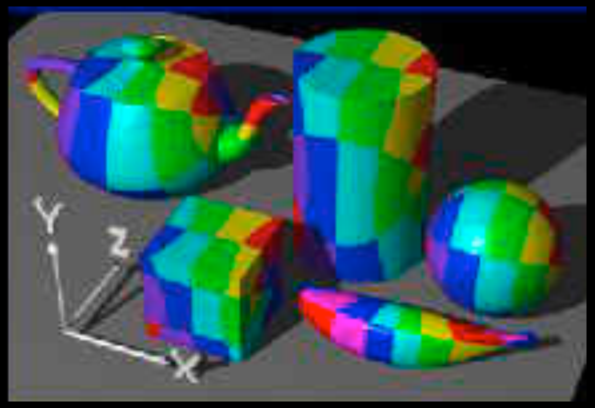
(x,y,z) -> (latitude,longitude)
-> (u,v)

[Rosalee Wolfe]

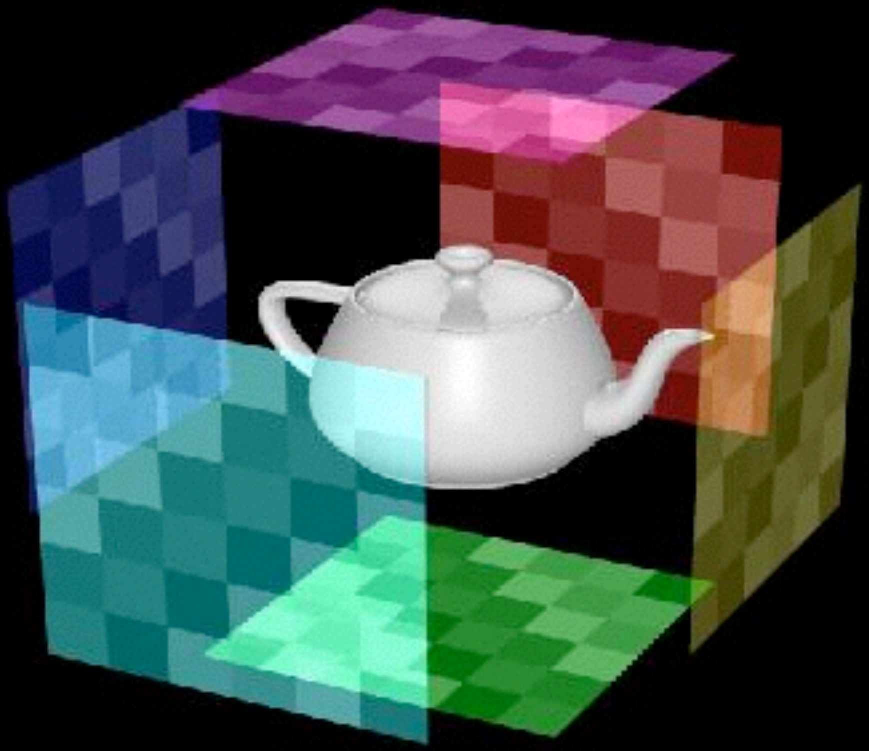


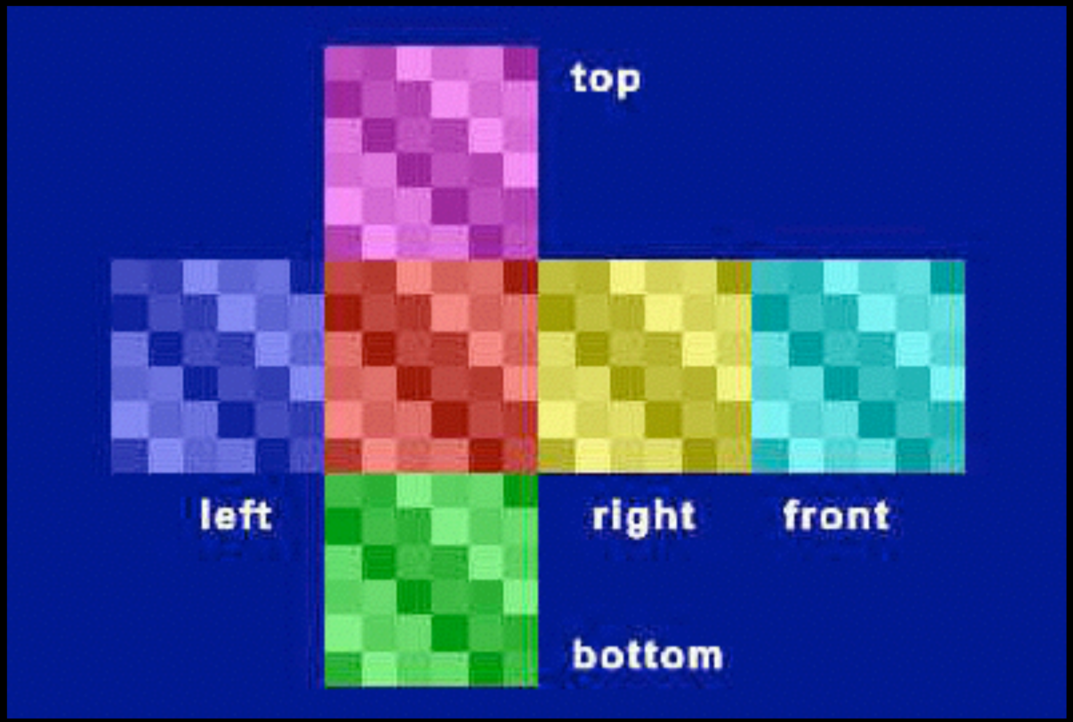


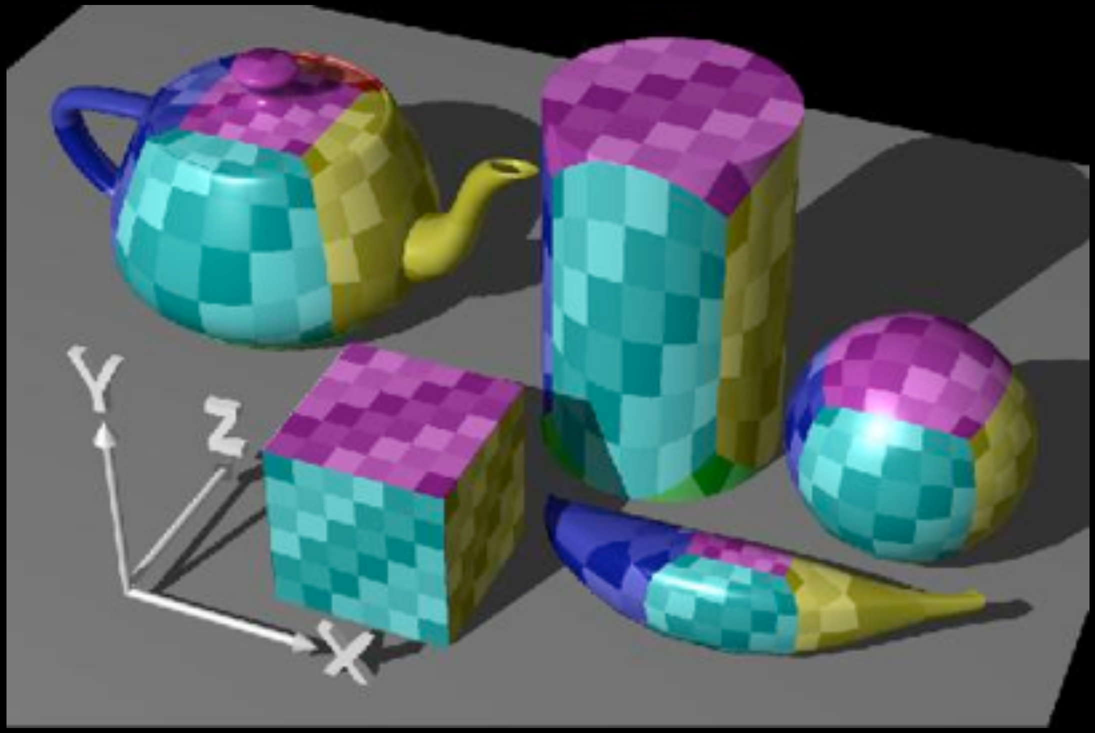




Box Mapping

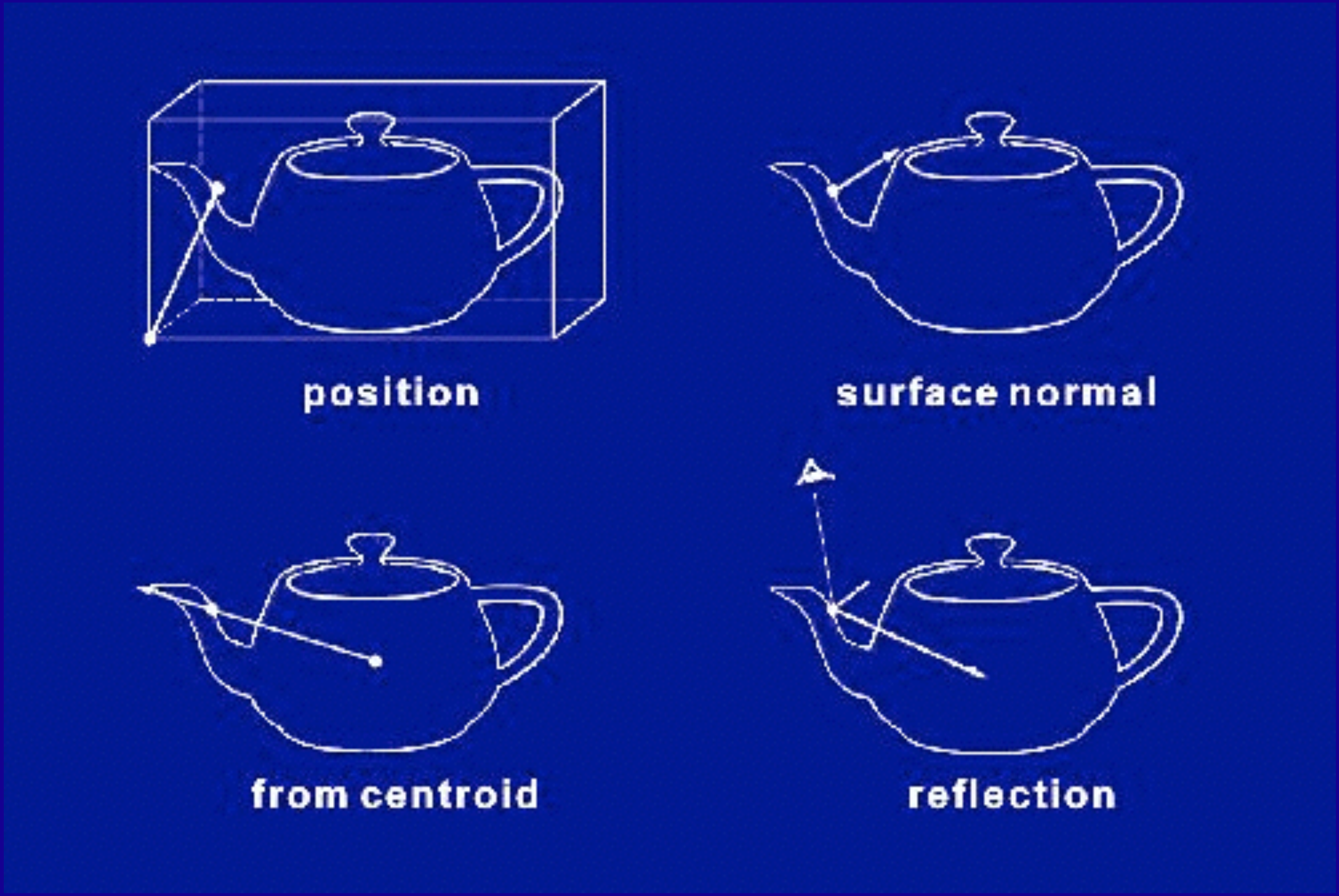






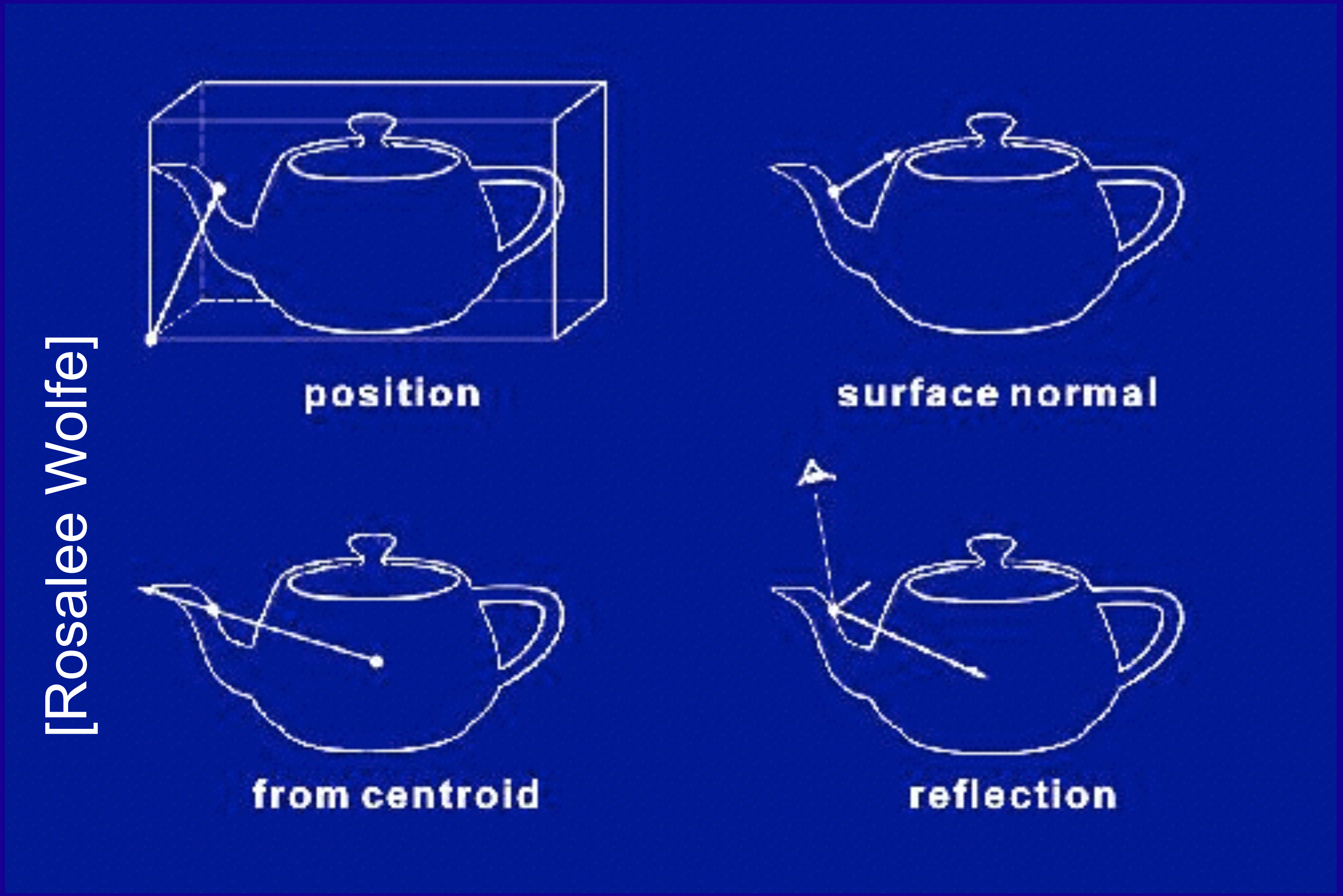
[Rosalee Wolfe]

How do we map between intermediate and actual objects?

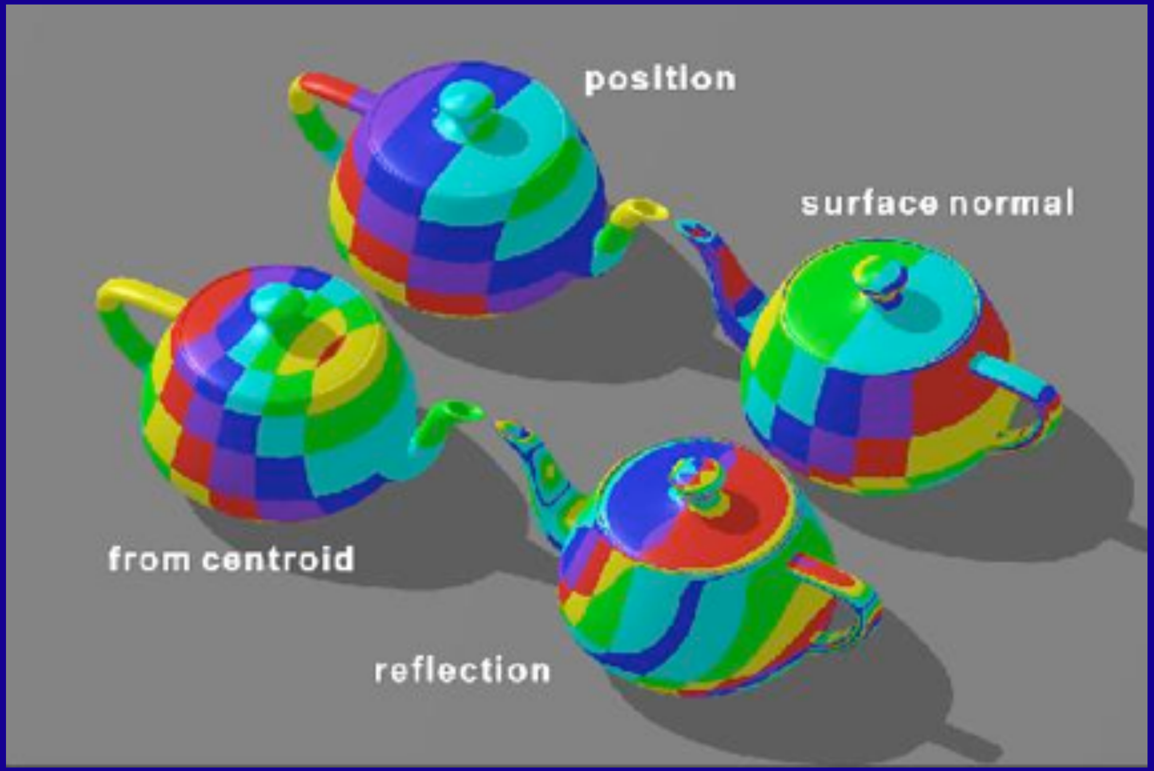


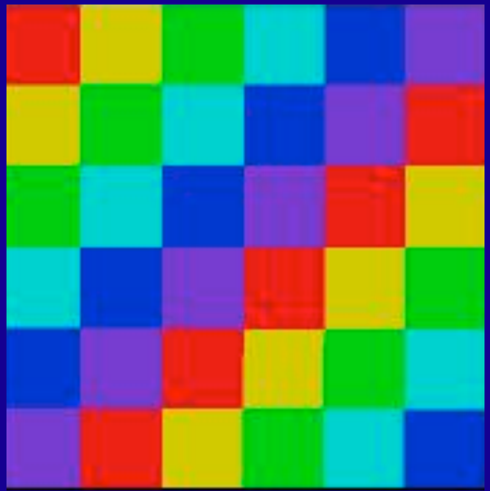
[Rosalee Wolfe]

How do we map between intermediate and actual objects?

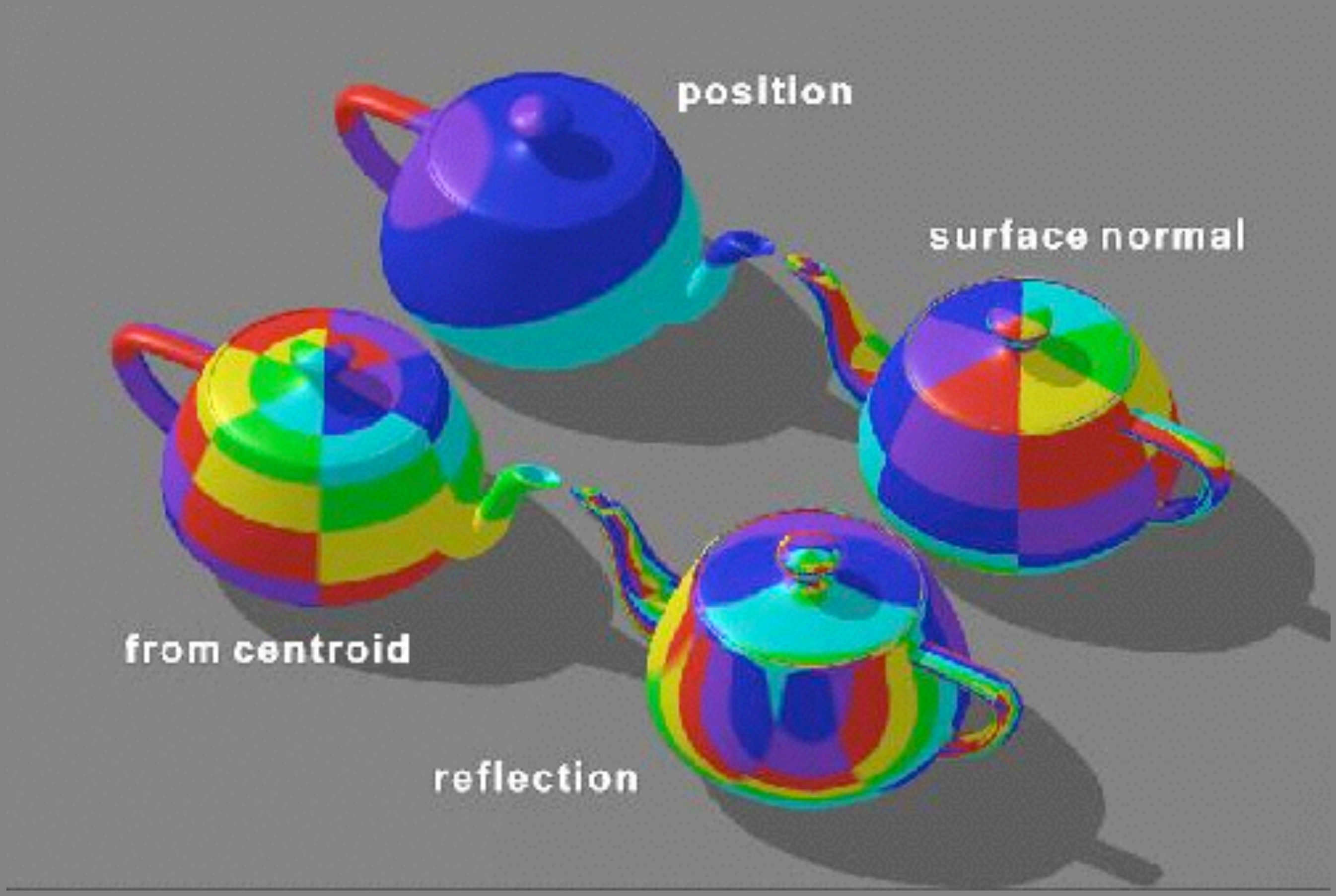


[Rosalee Wolfe]

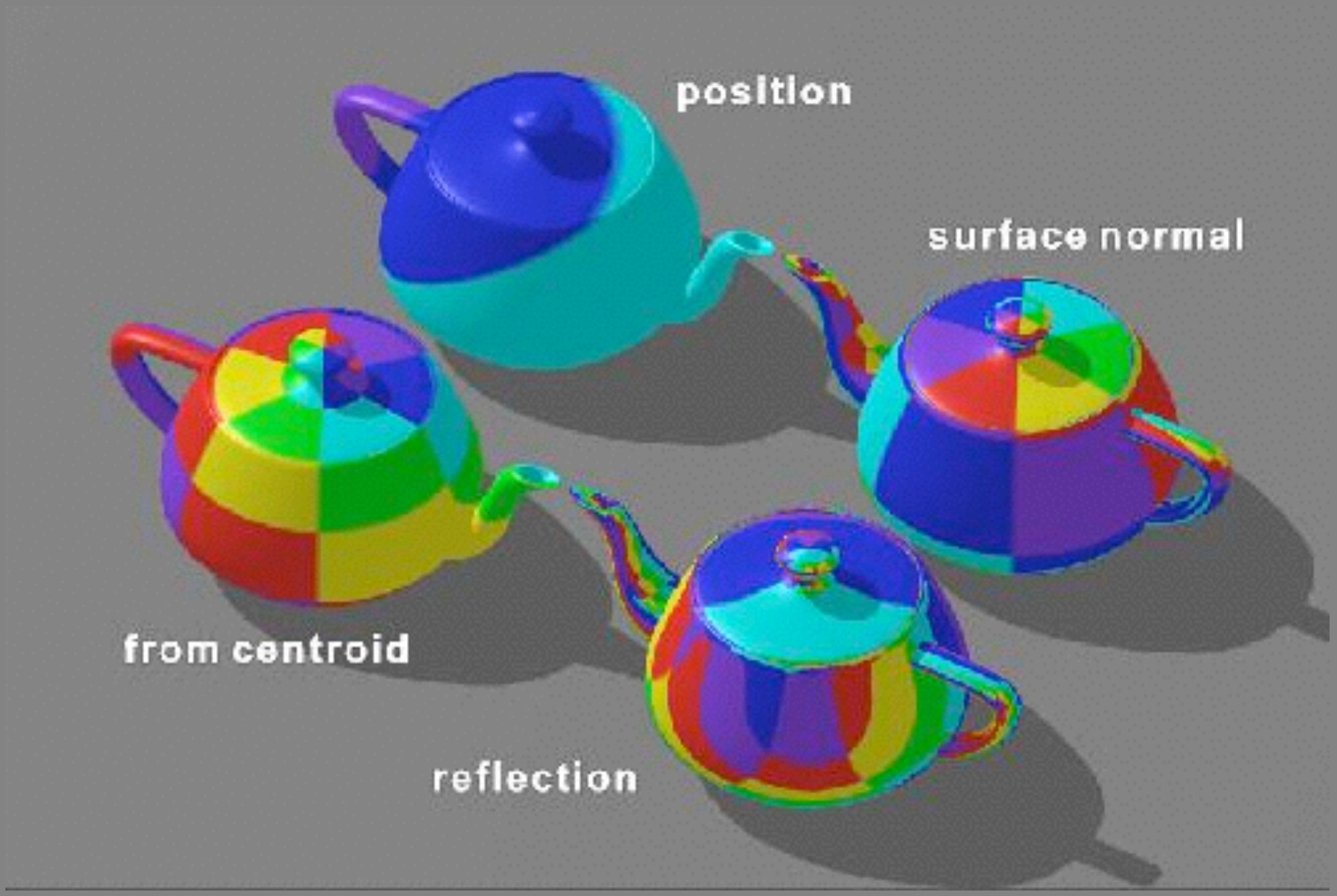




What intermediate shape was used here?

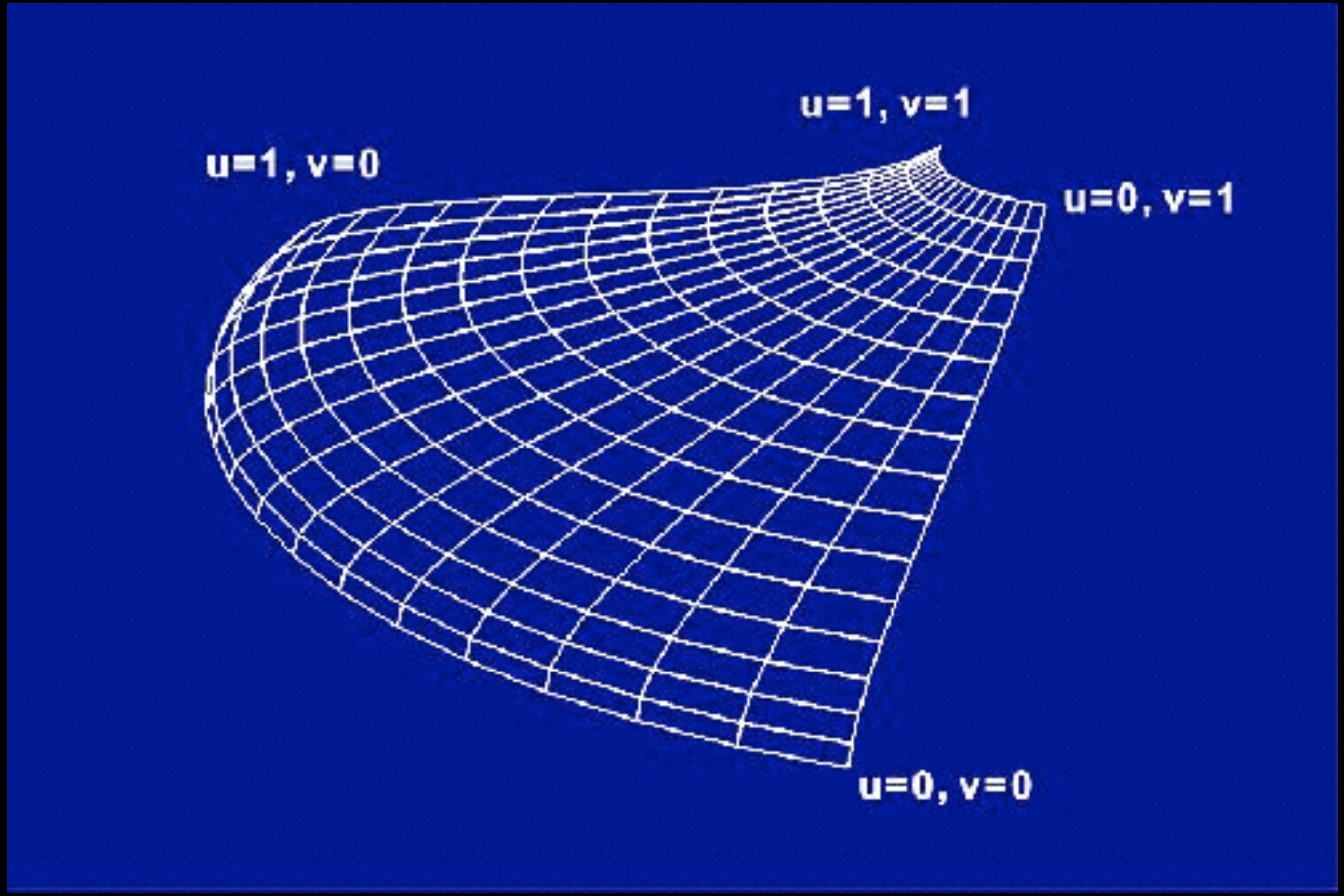


Cylindrical

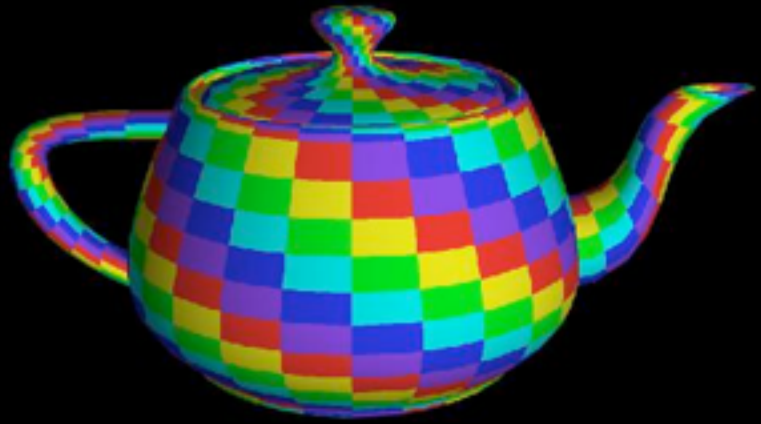


Spherical

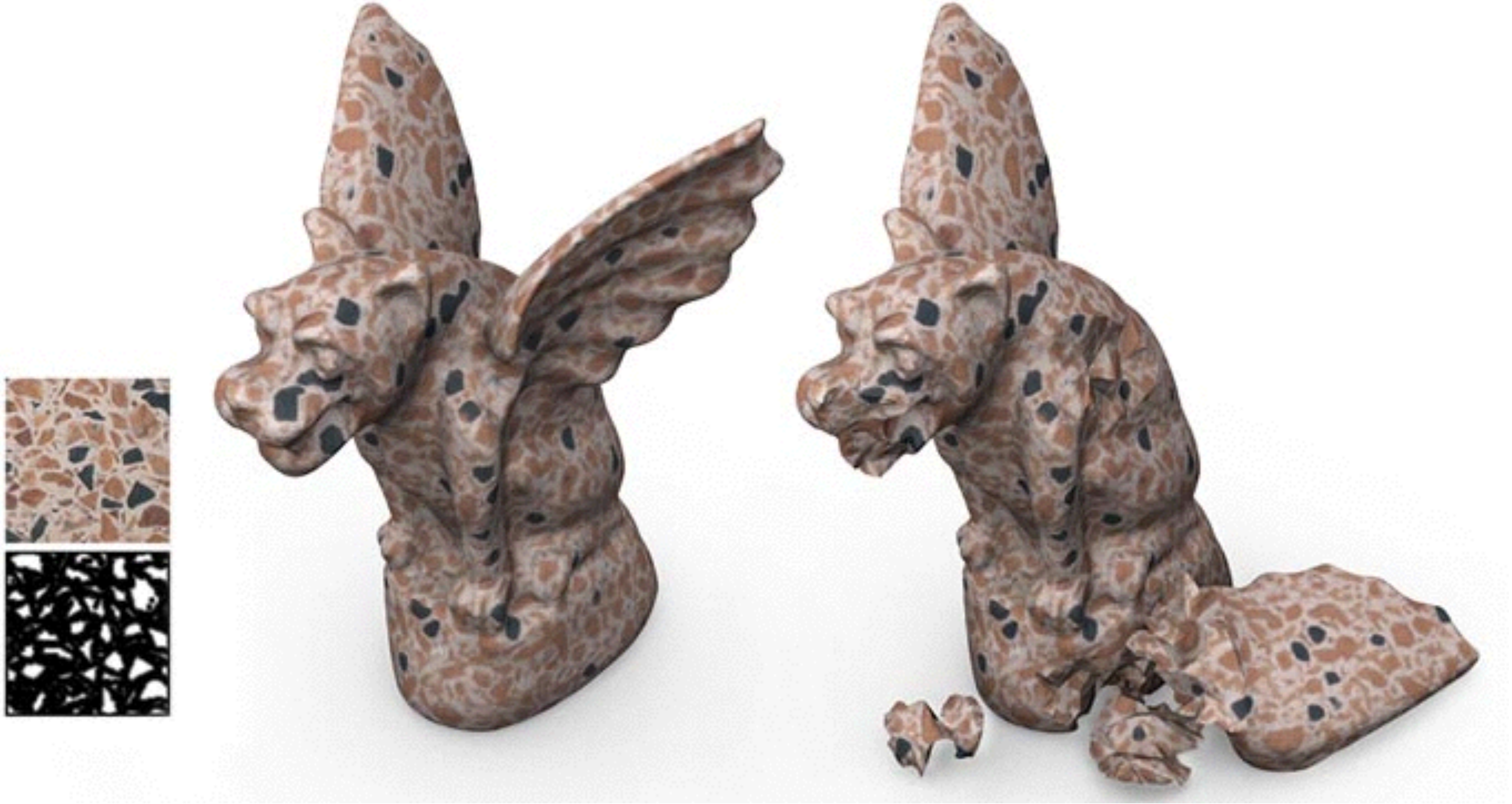
Parametric Surfaces



32 parametric patches



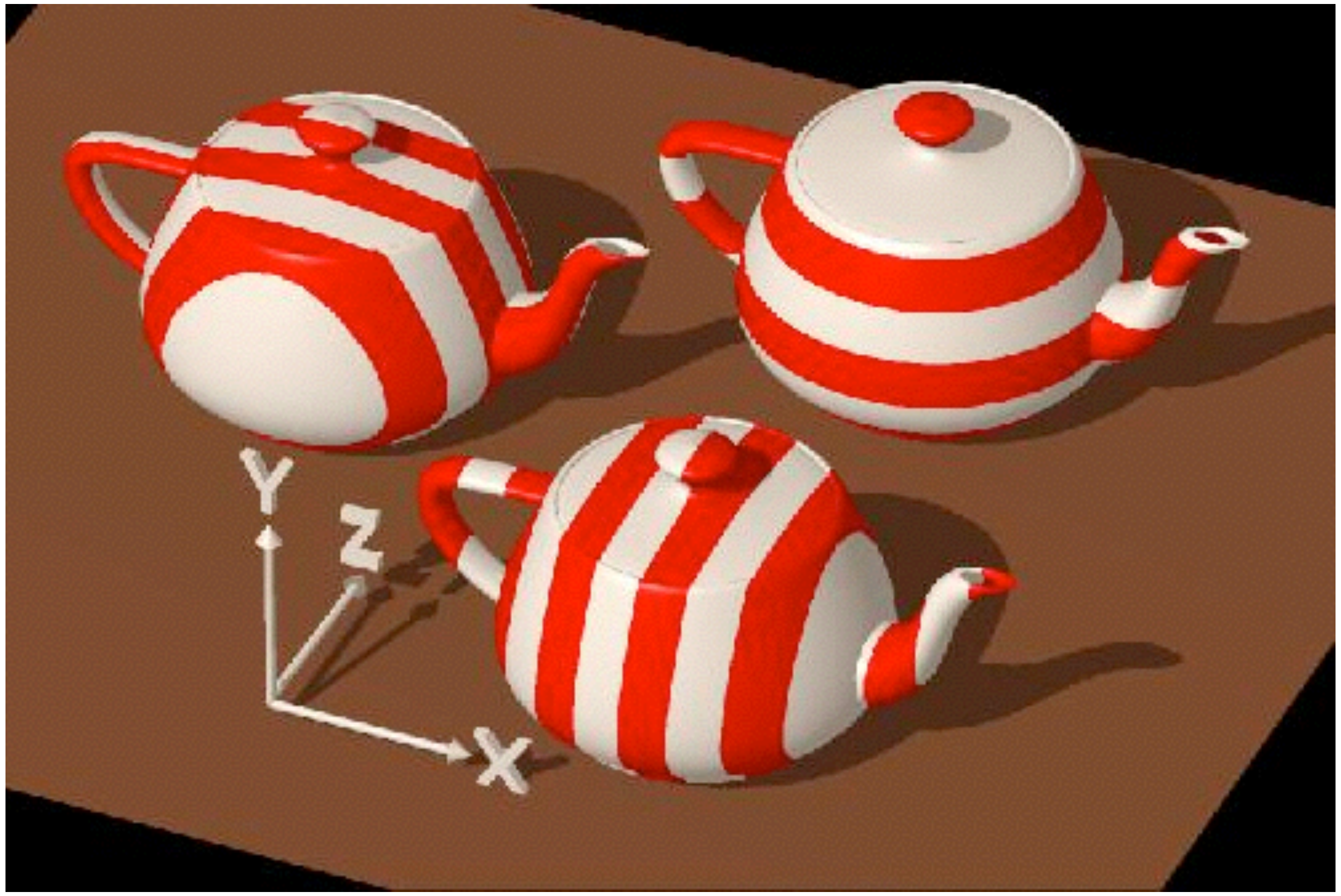
3D solid textures

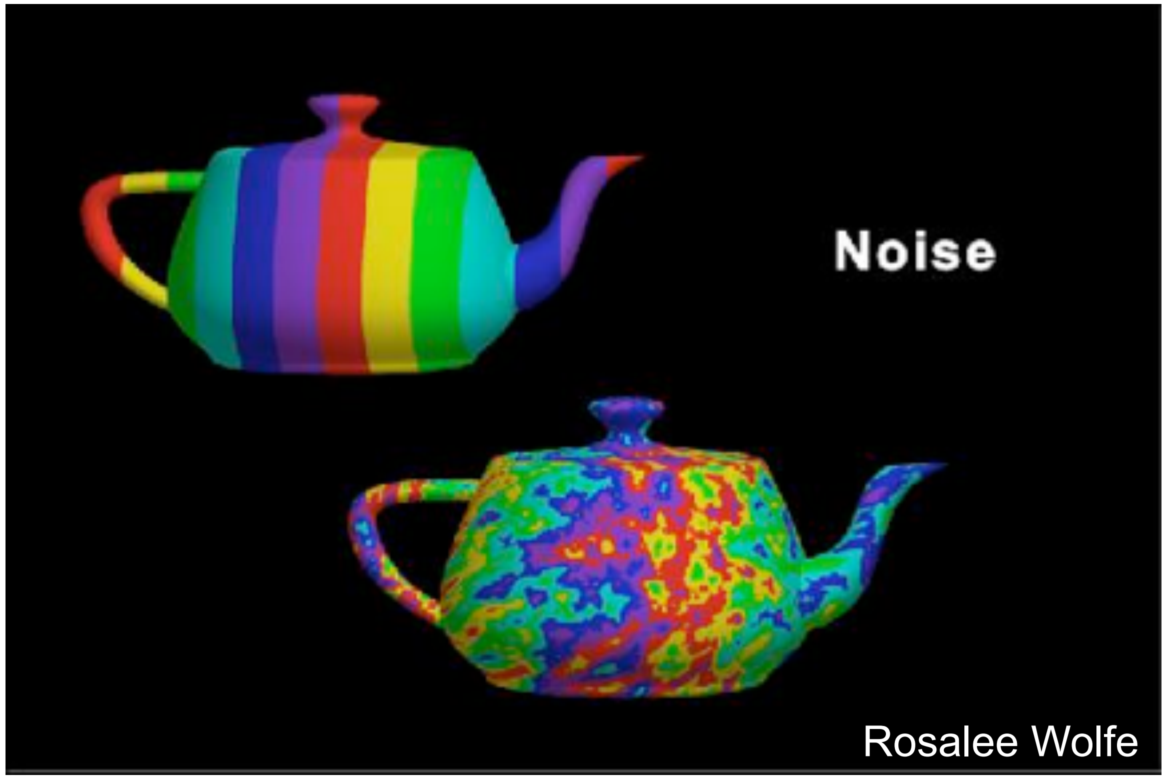


can map object (x,y,z) directly to texture (u,v,w)

[Dong et al., 2008]

Procedural textures





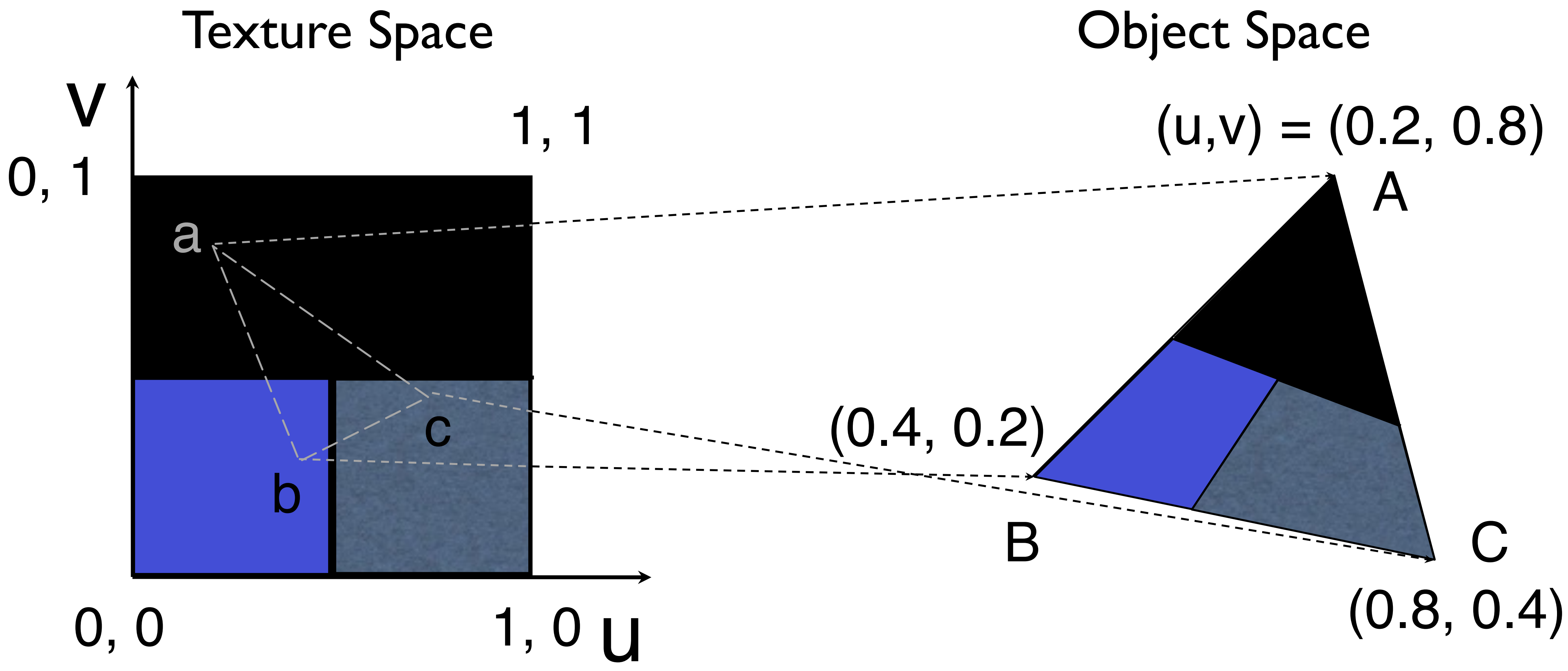
Rosalee Wolfe

e.g., Perlin noise

Triangles

Texturing triangles

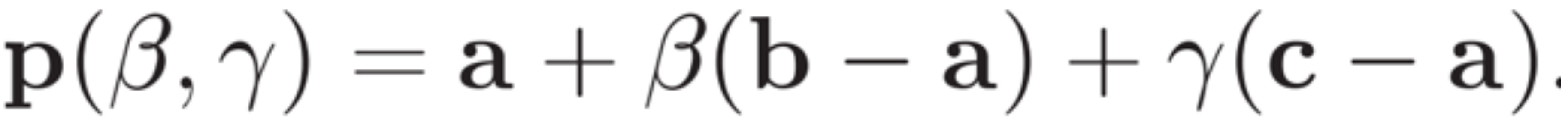
- Store (u,v) at each vertex
- interpolate inside triangles using barycentric coordinates



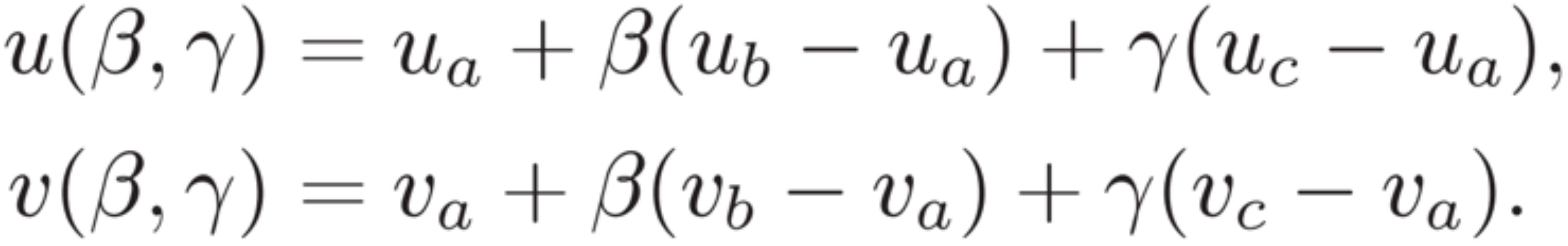
[Angel and Shreiner]

Texturing triangles

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



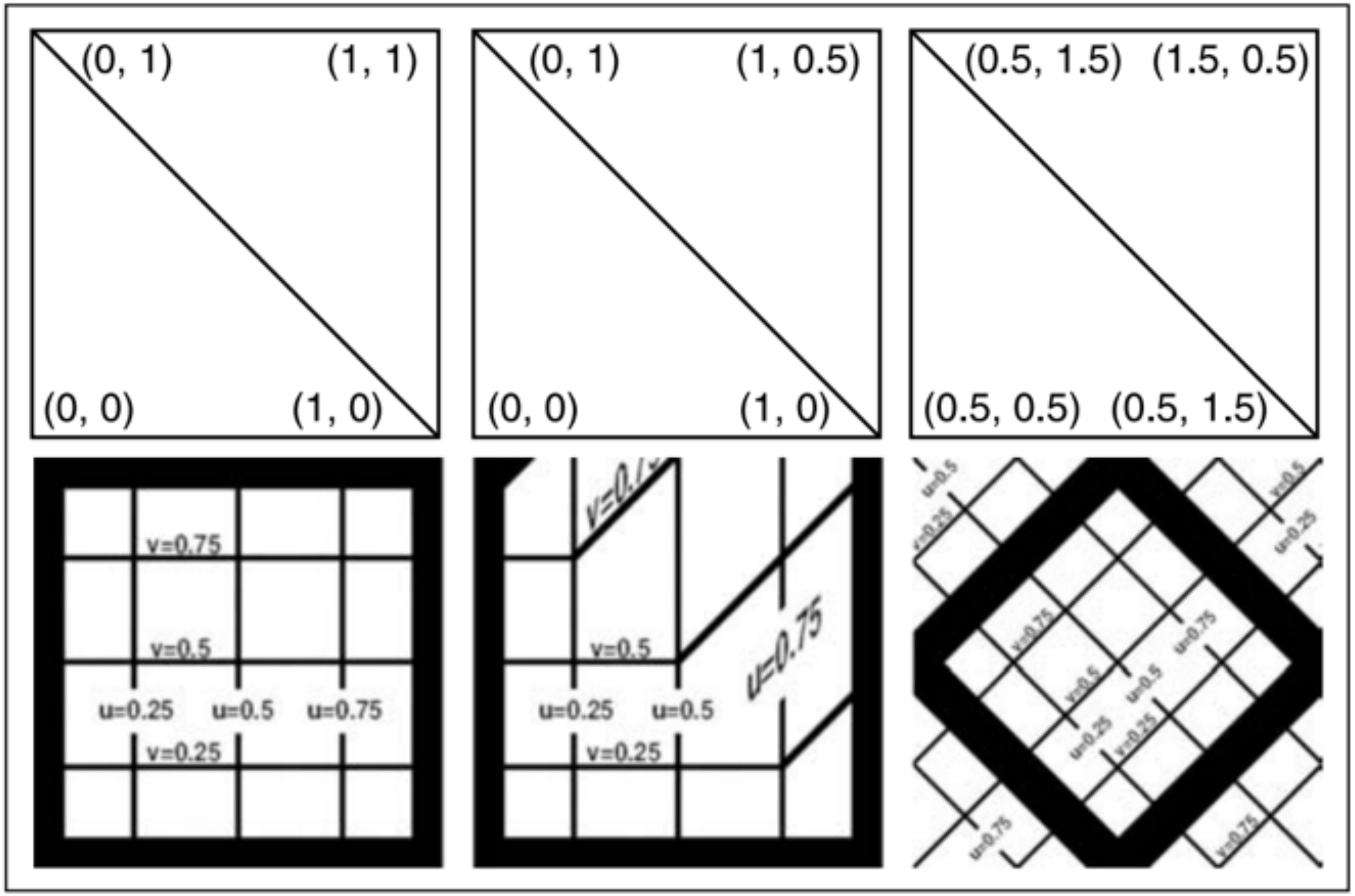
$$\mathbf{p}(\beta, \gamma) = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a}).$$



$$u(\beta, \gamma) = u_a + \beta(u_b - u_a) + \gamma(u_c - u_a),$$
$$v(\beta, \gamma) = v_a + \beta(v_b - v_a) + \gamma(v_c - v_a).$$

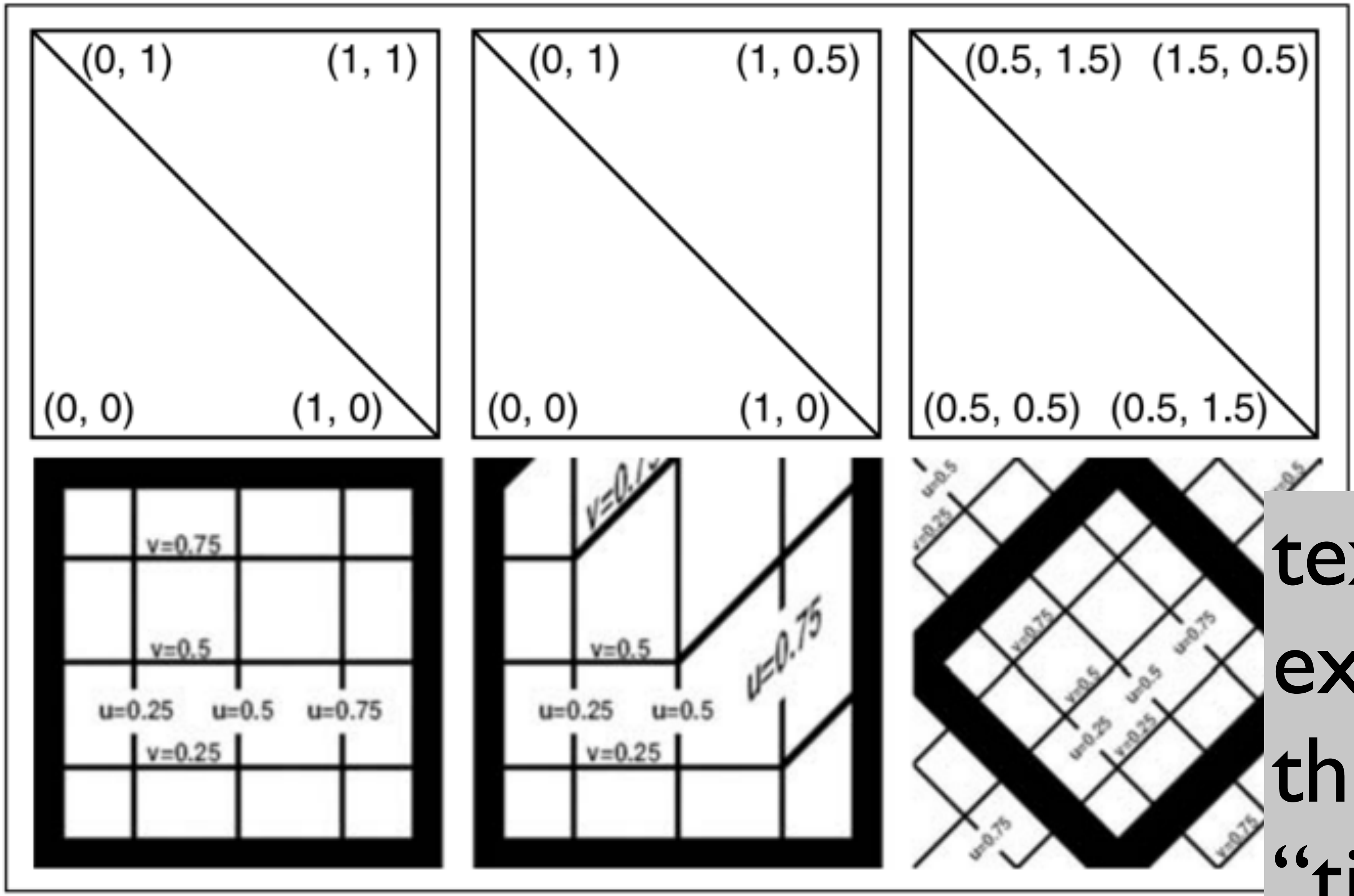
Texturing triangles

Choice of (u,v) makes big difference



Texturing triangles

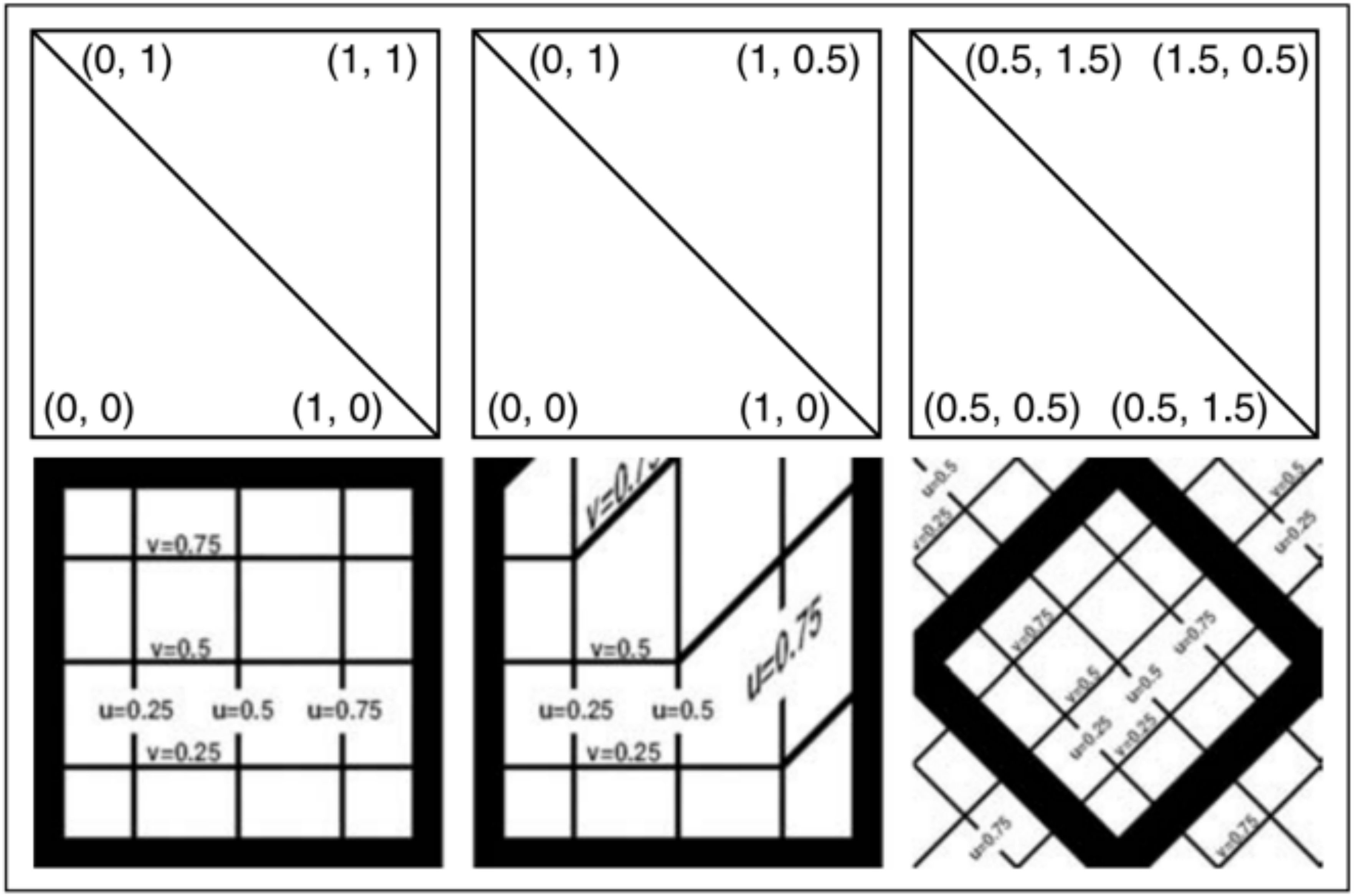
Choice of (u,v) makes big difference



texture extended through "tiling"

Texturing triangles

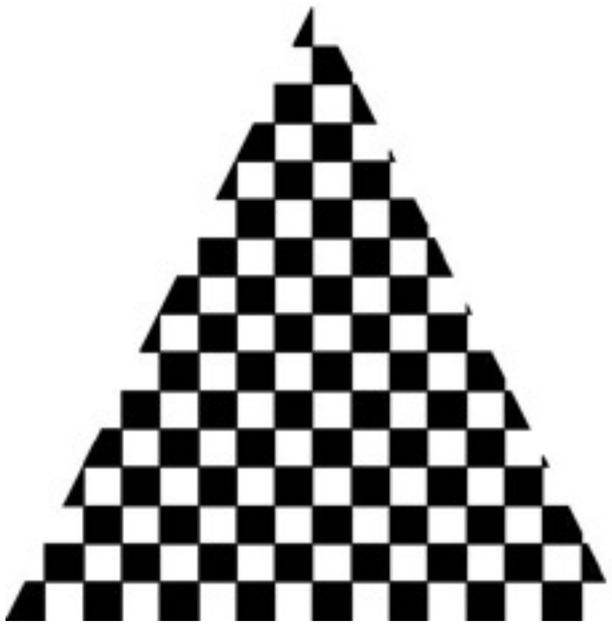
Choice of (u,v) makes big difference



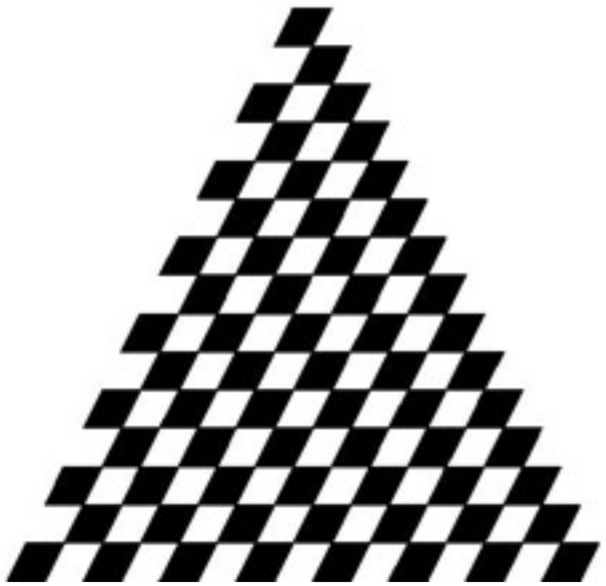
Textures in OpenGL

```
glTexCoord*()
```

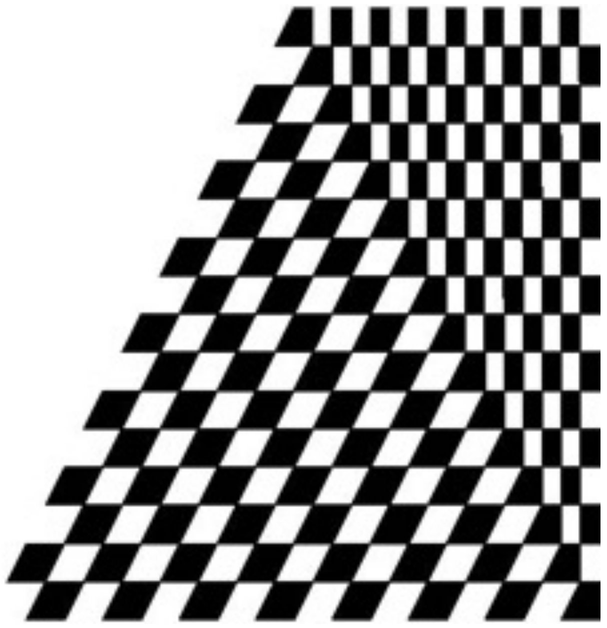
- Assign (u,v) to vertices
- OpenGL then uses interpolation for triangle interior



good selection
of tex coordinates

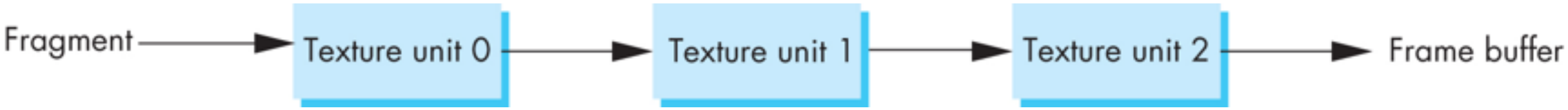


poor selection
of tex coordinates

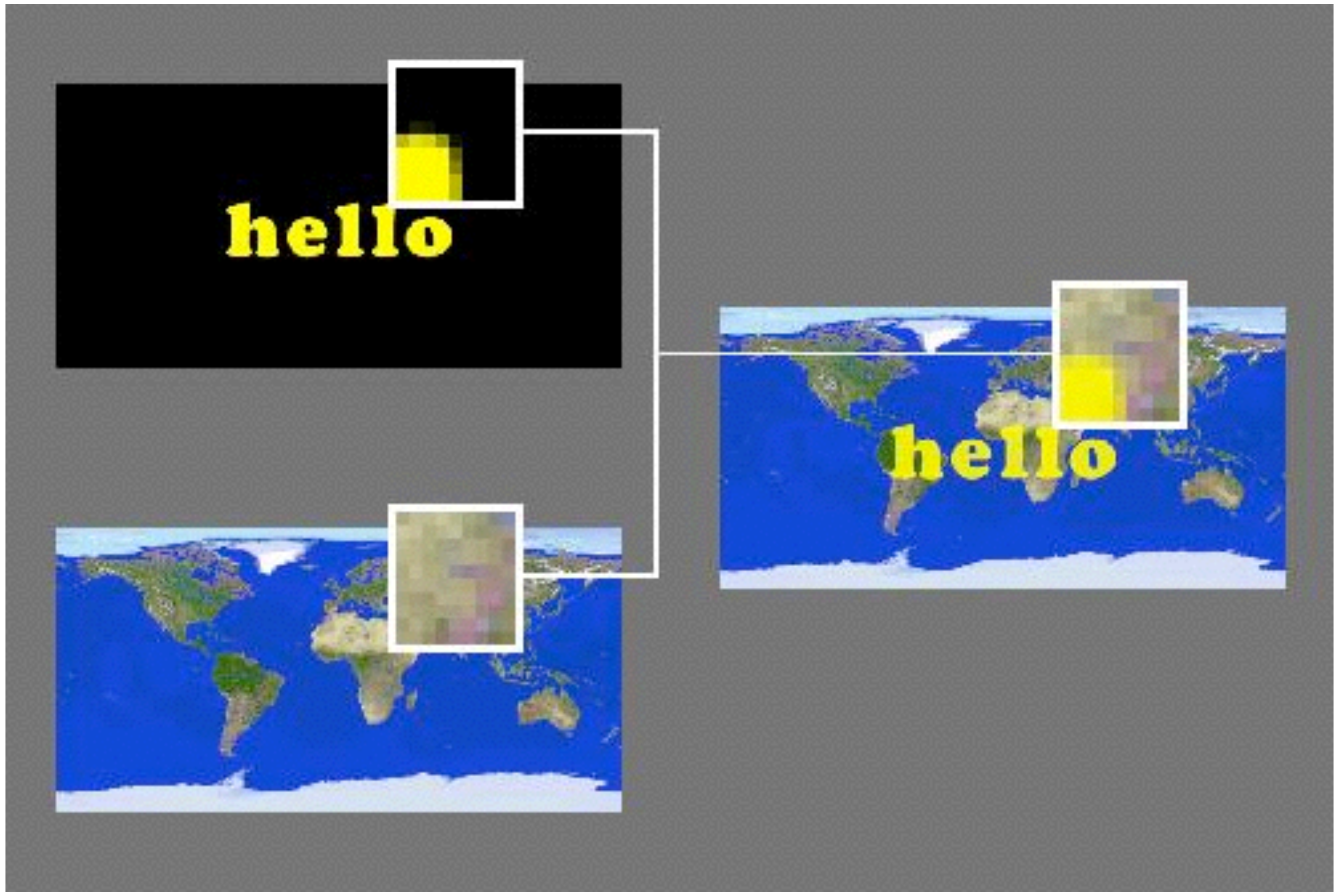


texture stretched
over trapezoid
showing effects of
bilinear interpolation

Multitexturing



Fragment → Texture unit 0 → Texture unit 1 → Texture unit 2 → Frame buffer

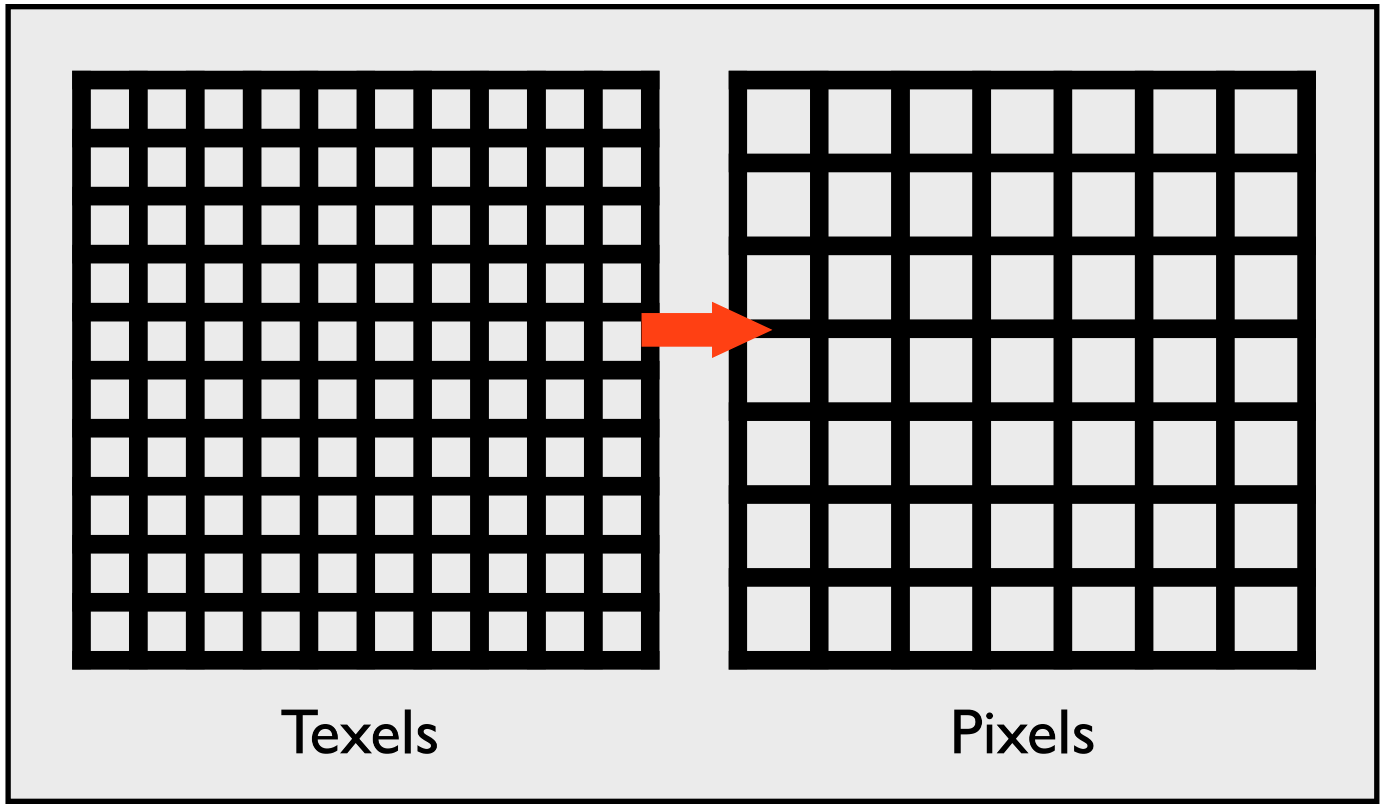


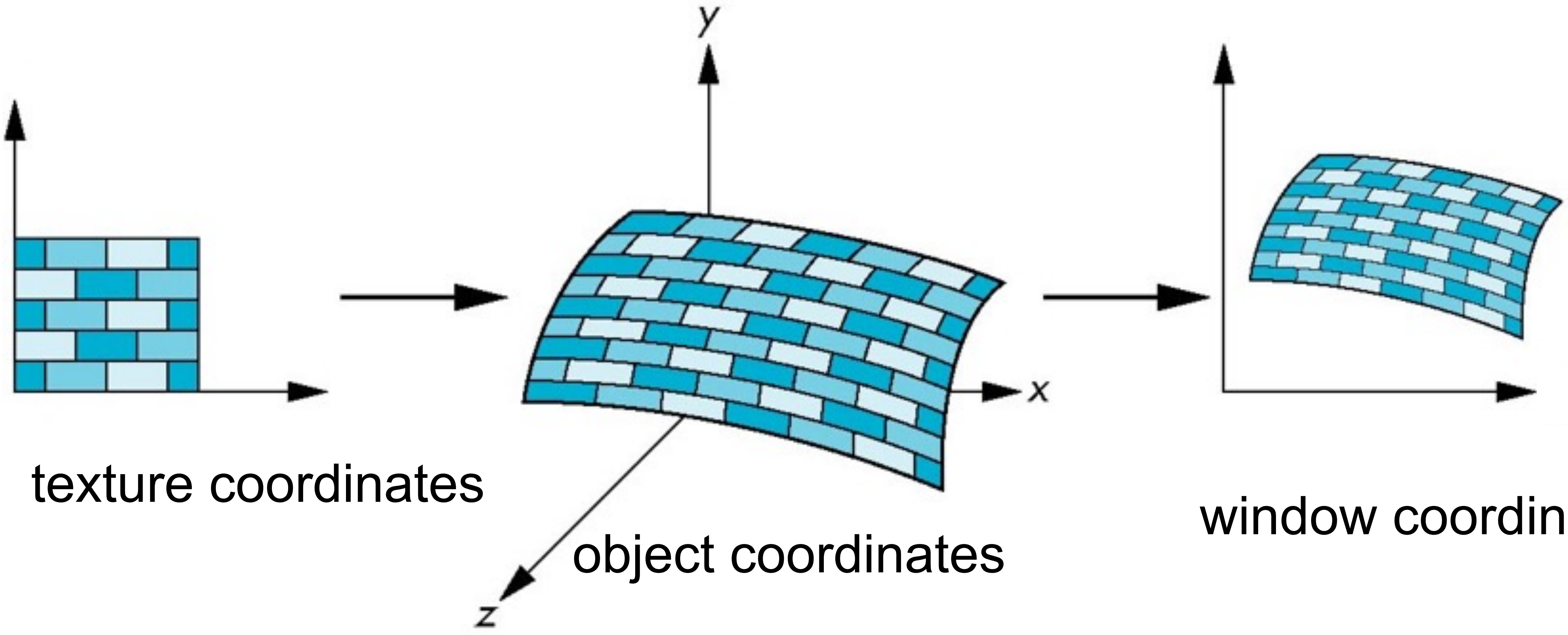
hello
hello
hello
hello

Rosalee Wolfe

Texture Sampling

Texture Mapping





texture coordinates

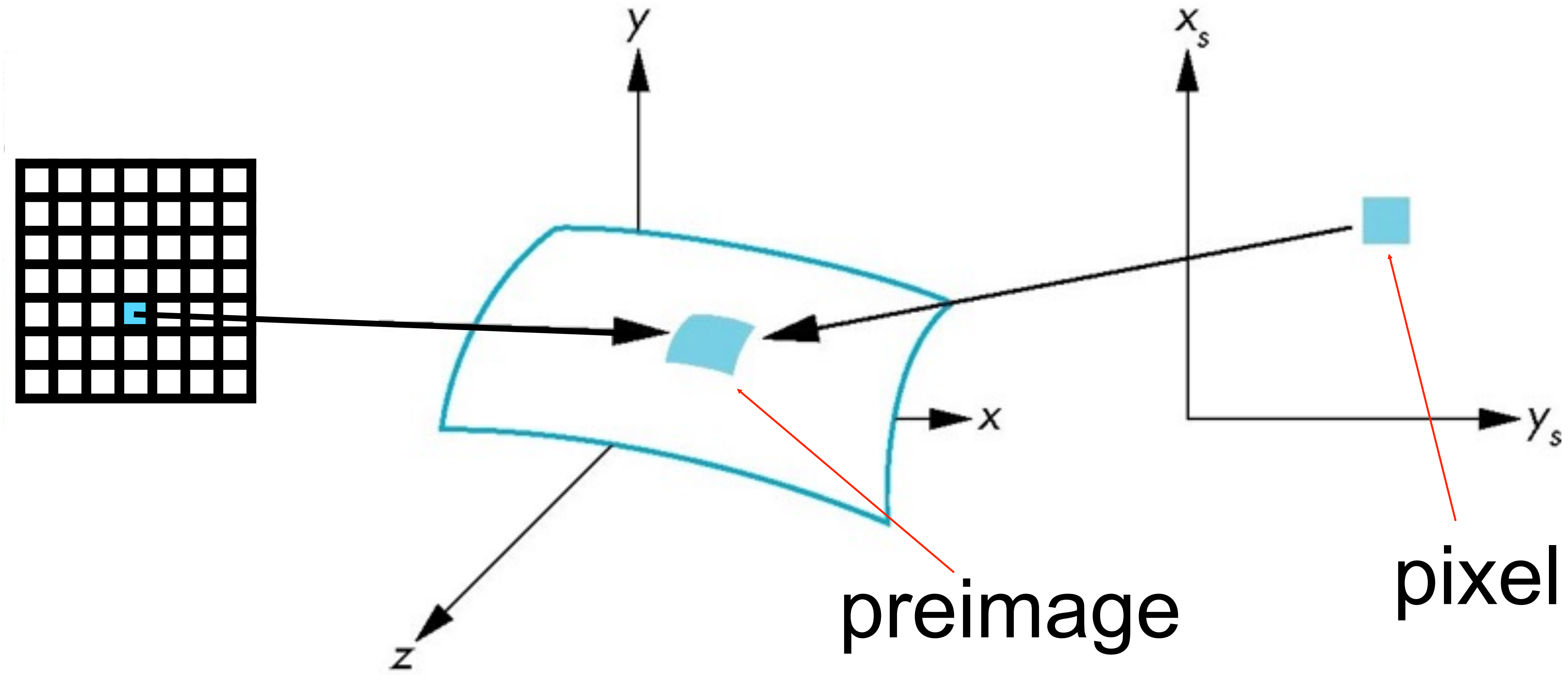
object coordinates

window coordinates

[Angel and Shreiner]

Point Sampling

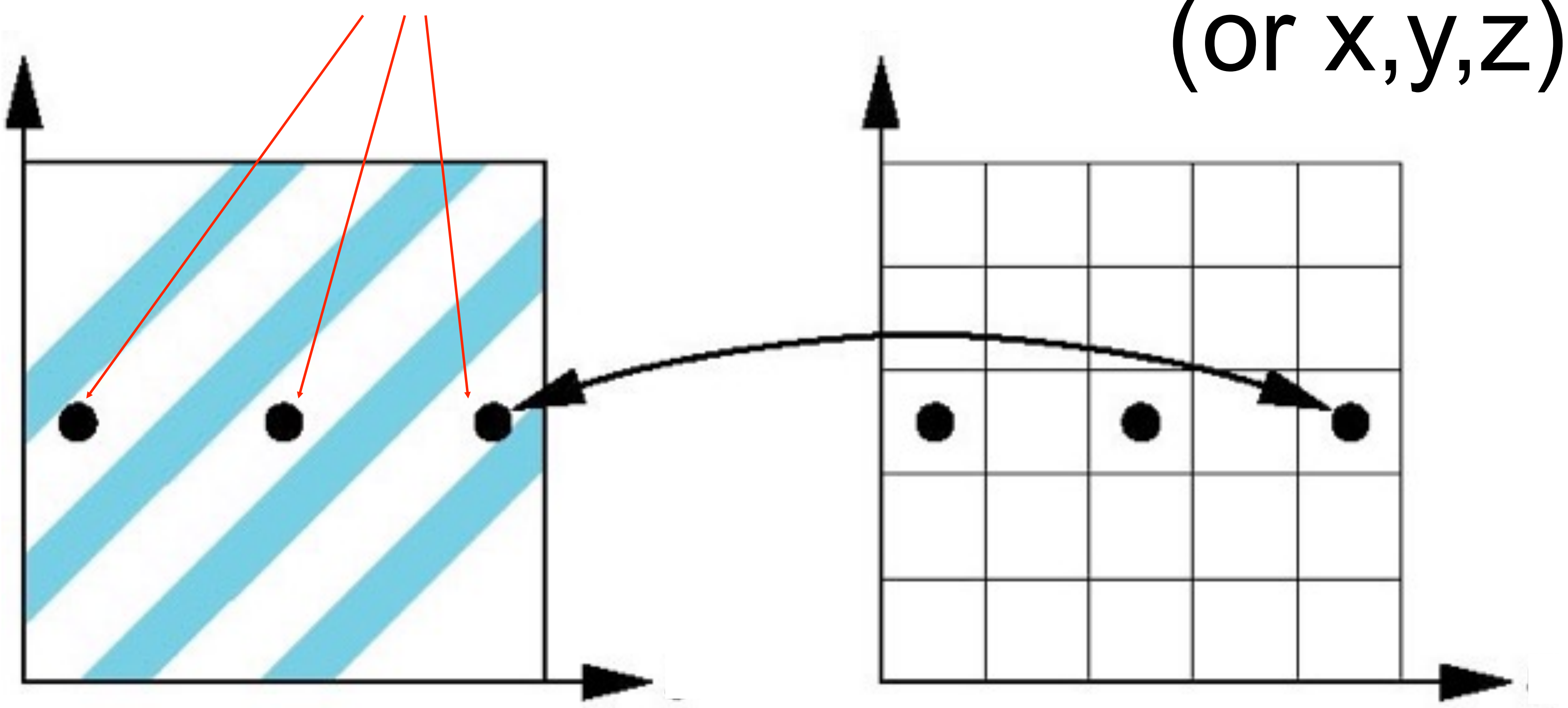
Map back to texture image and use the **nearest texel**



[Angel and Shreiner]

Aliasing

Point sampling of the texture can lead to aliasing artifacts



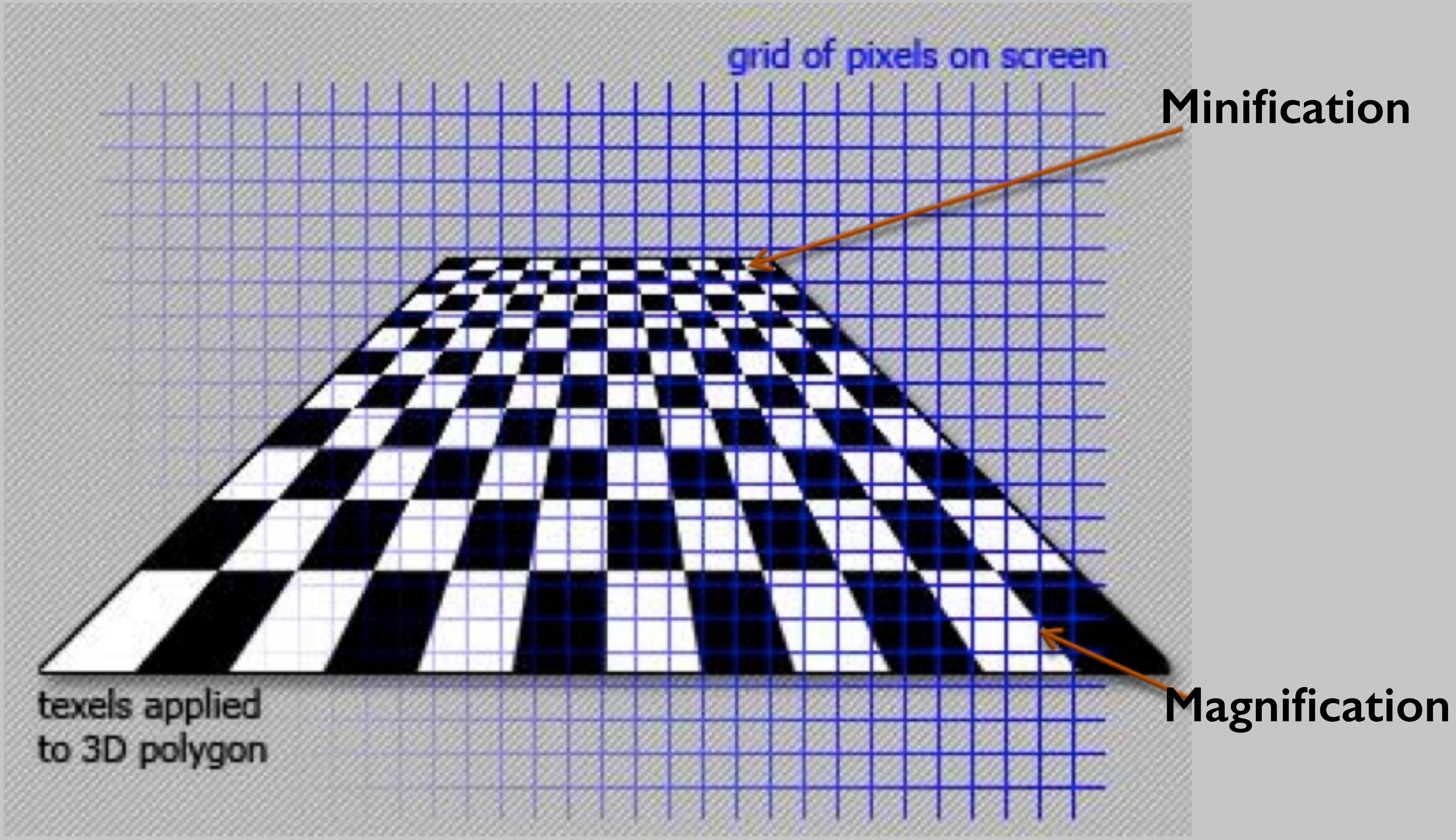
miss blue stripes

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

point samples in texture space

[Angel and Shreiner]

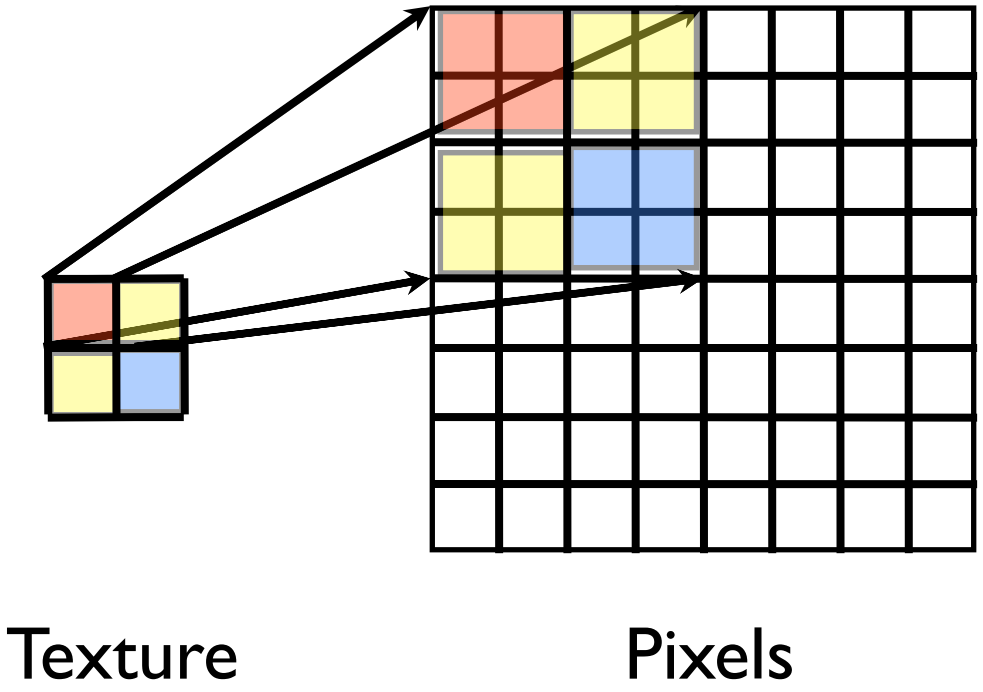
Magnification and Minification



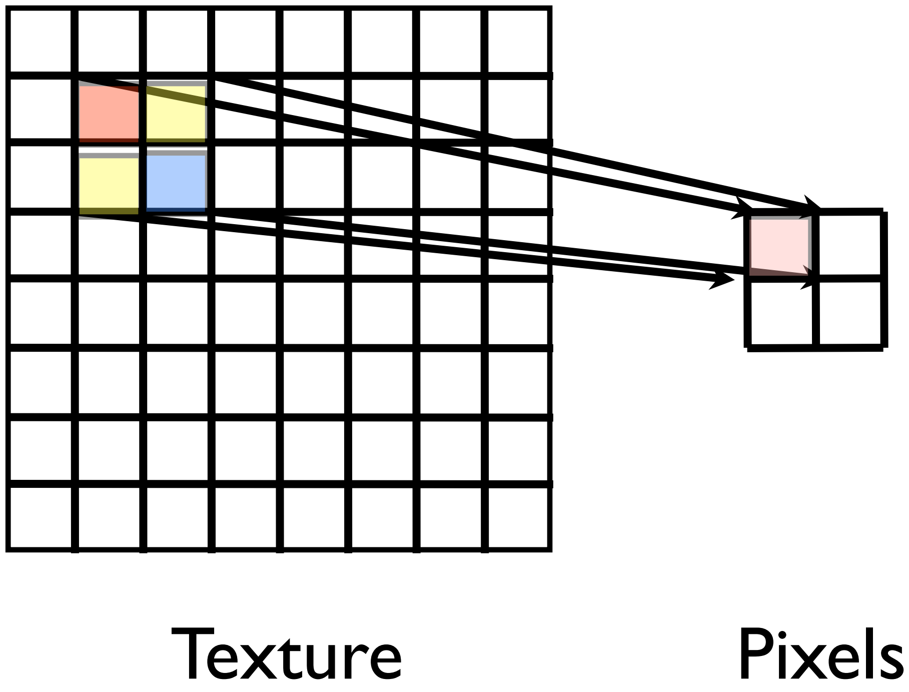
Magnification and Minification

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

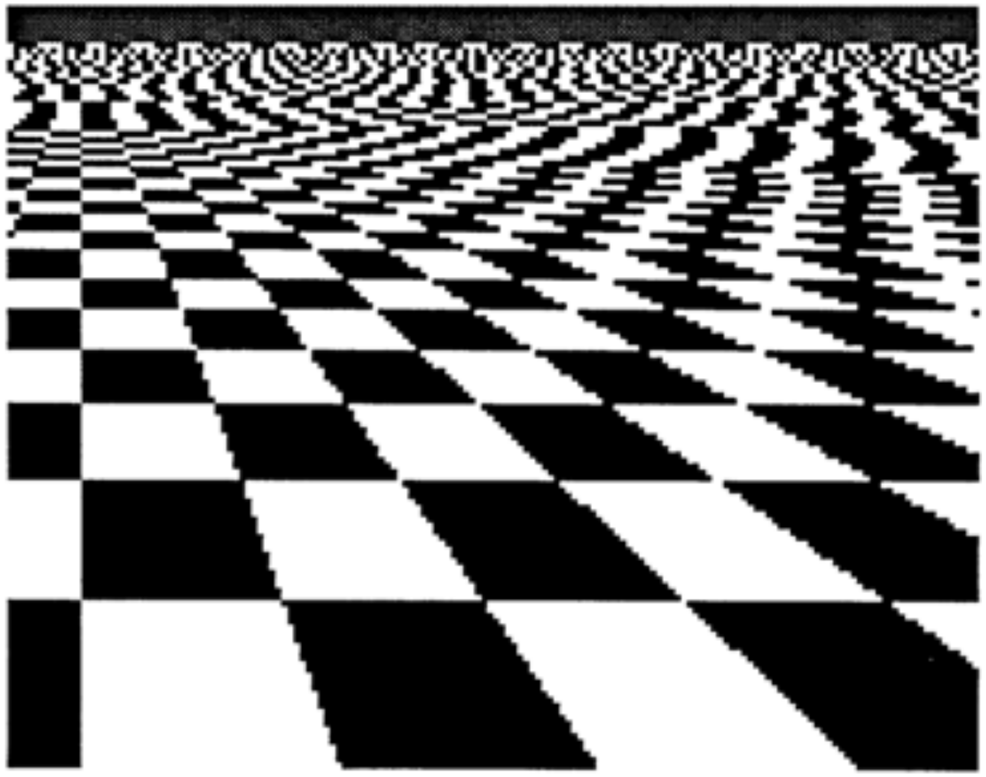


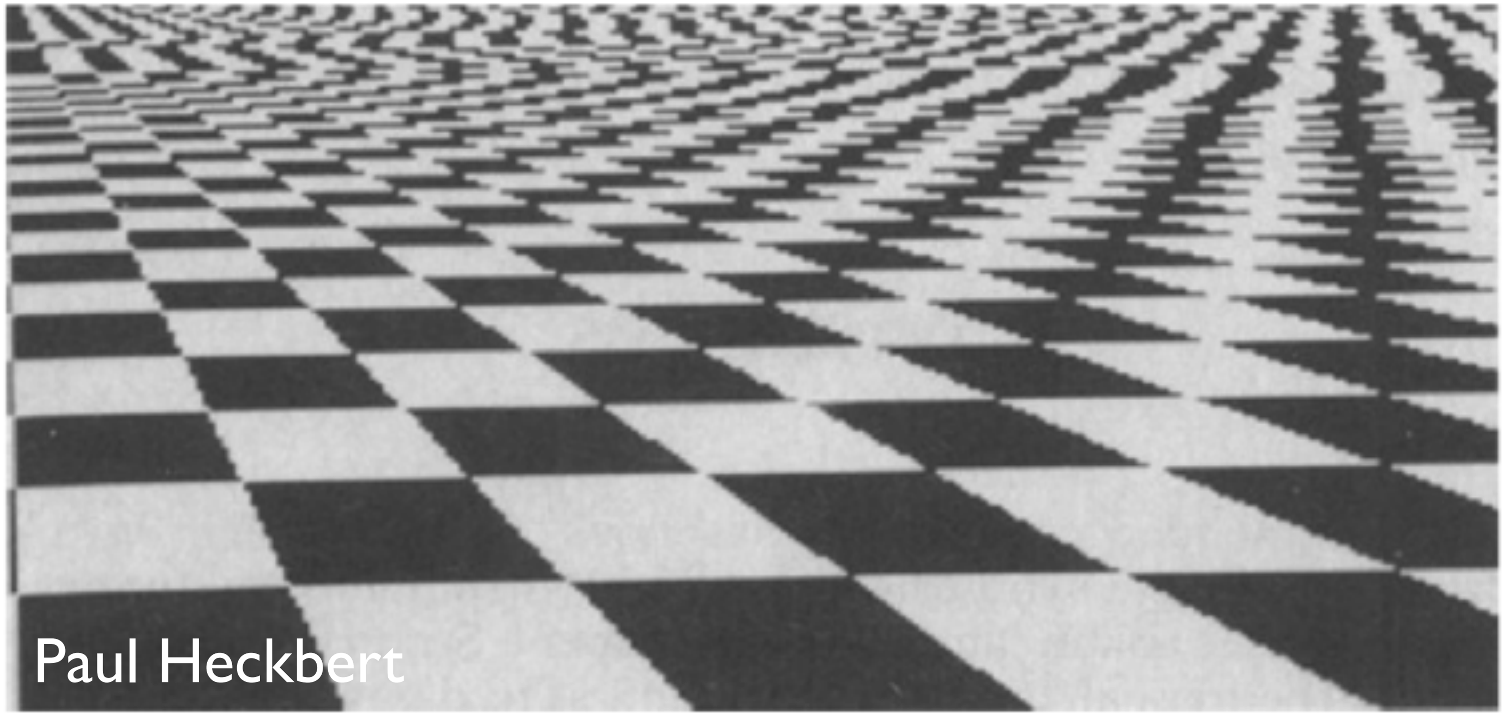
Magnification



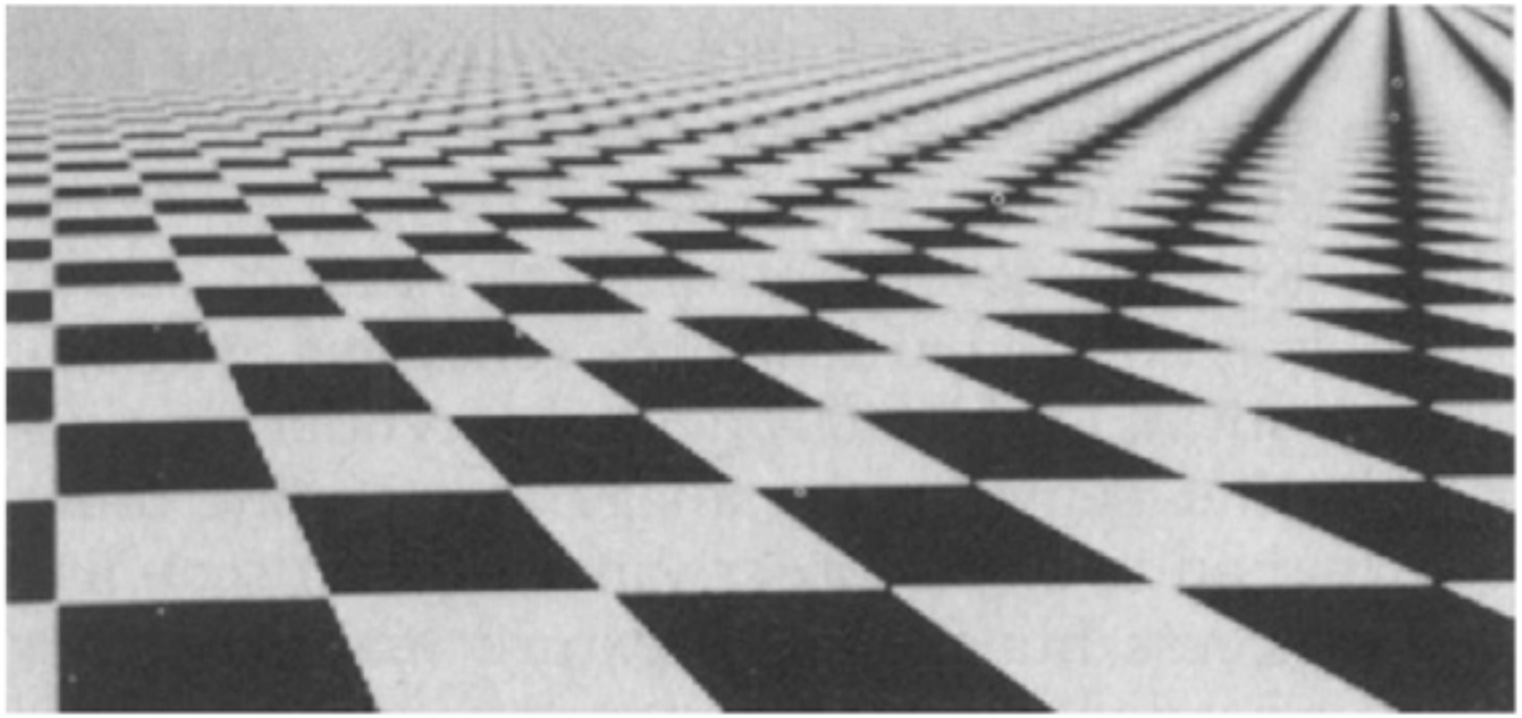
Minification

Aliasing artifacts





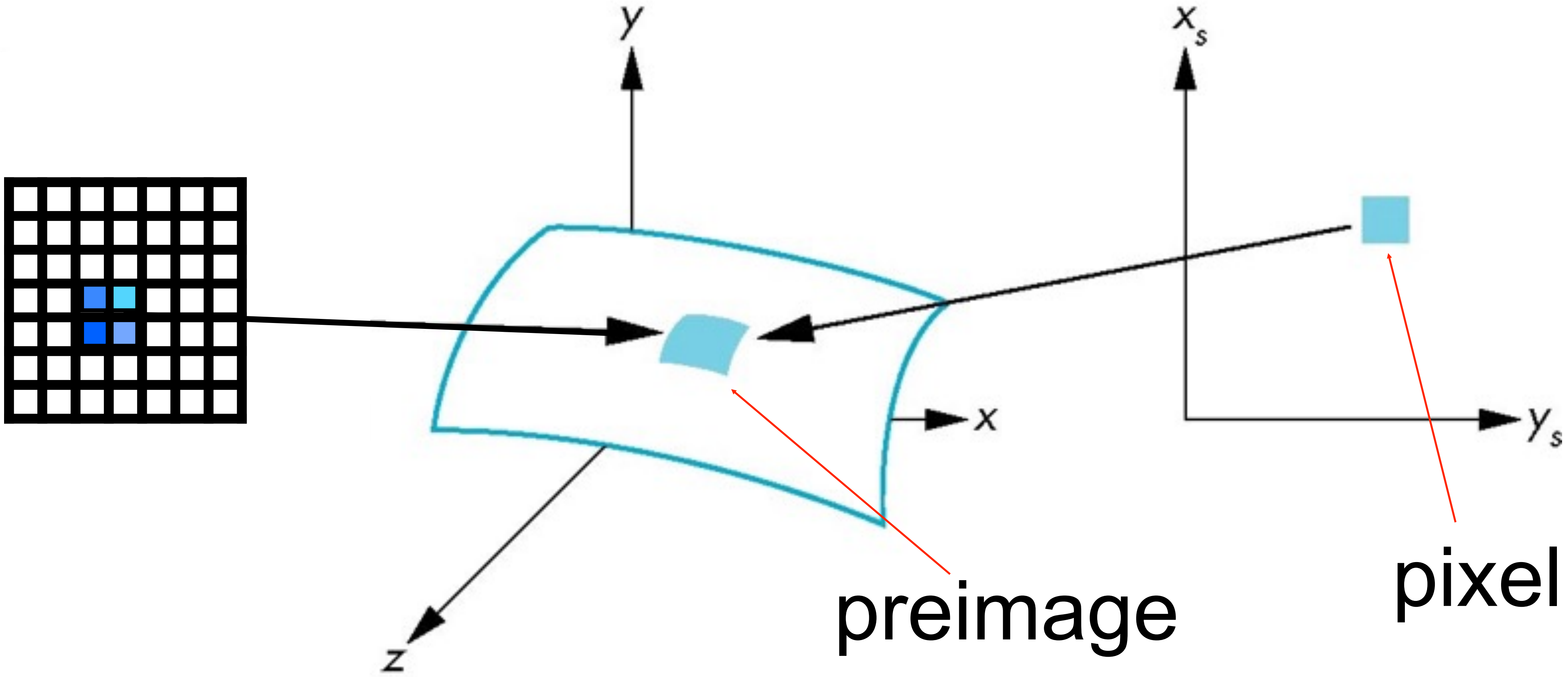
Paul Heckbert



We apply **filtering** to reduce aliasing artifacts

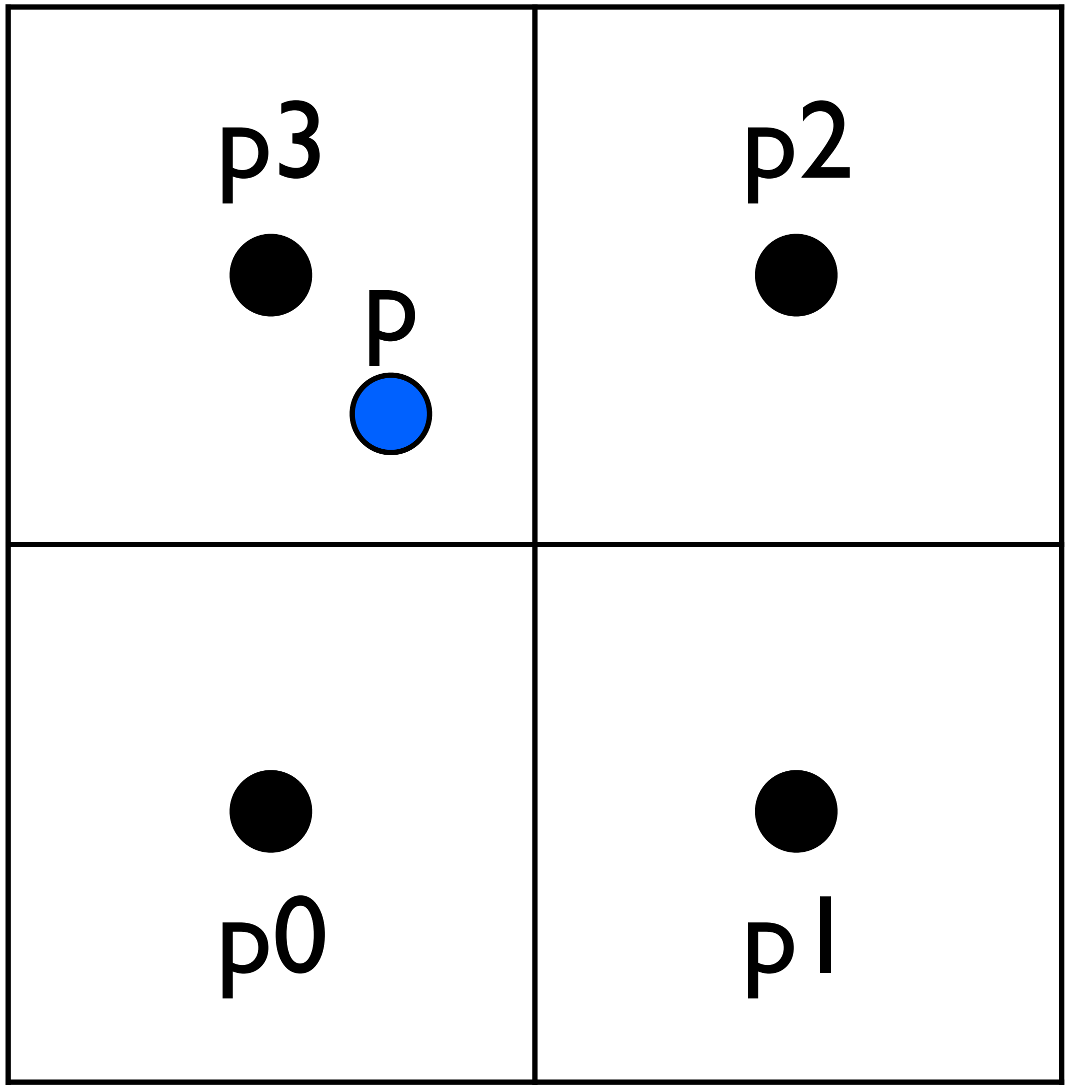
Area Averaging

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

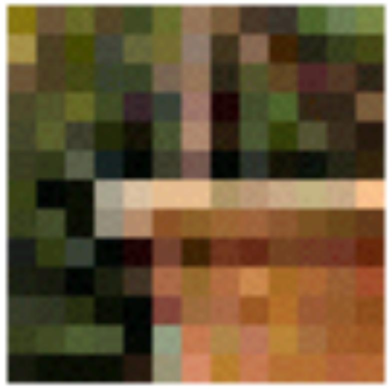


[Angel and Shreiner]

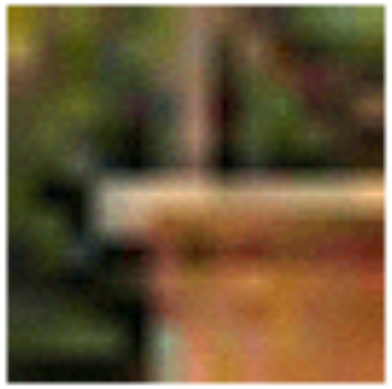
Use bilinear filtering



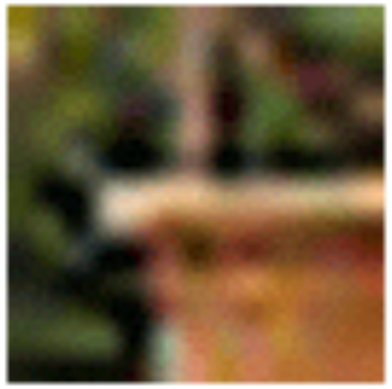
$p = ?$



nearest neighbor



bilinear

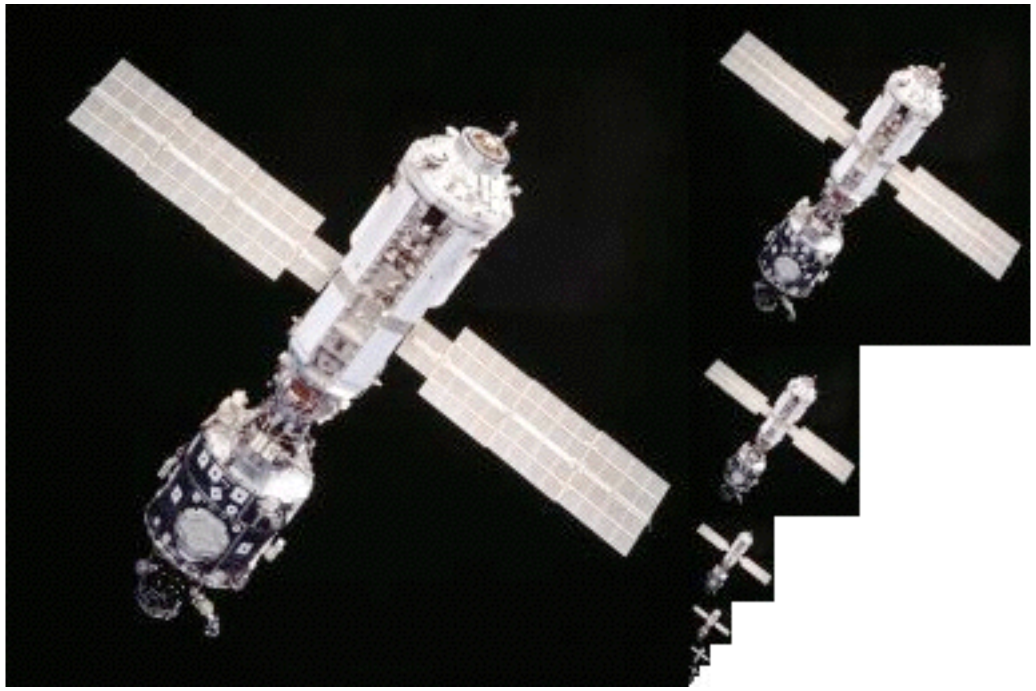


Wikipedia

bicubic

mitigate magnification artifacts

Mipmapping



Togikun, Wikimedia Commons

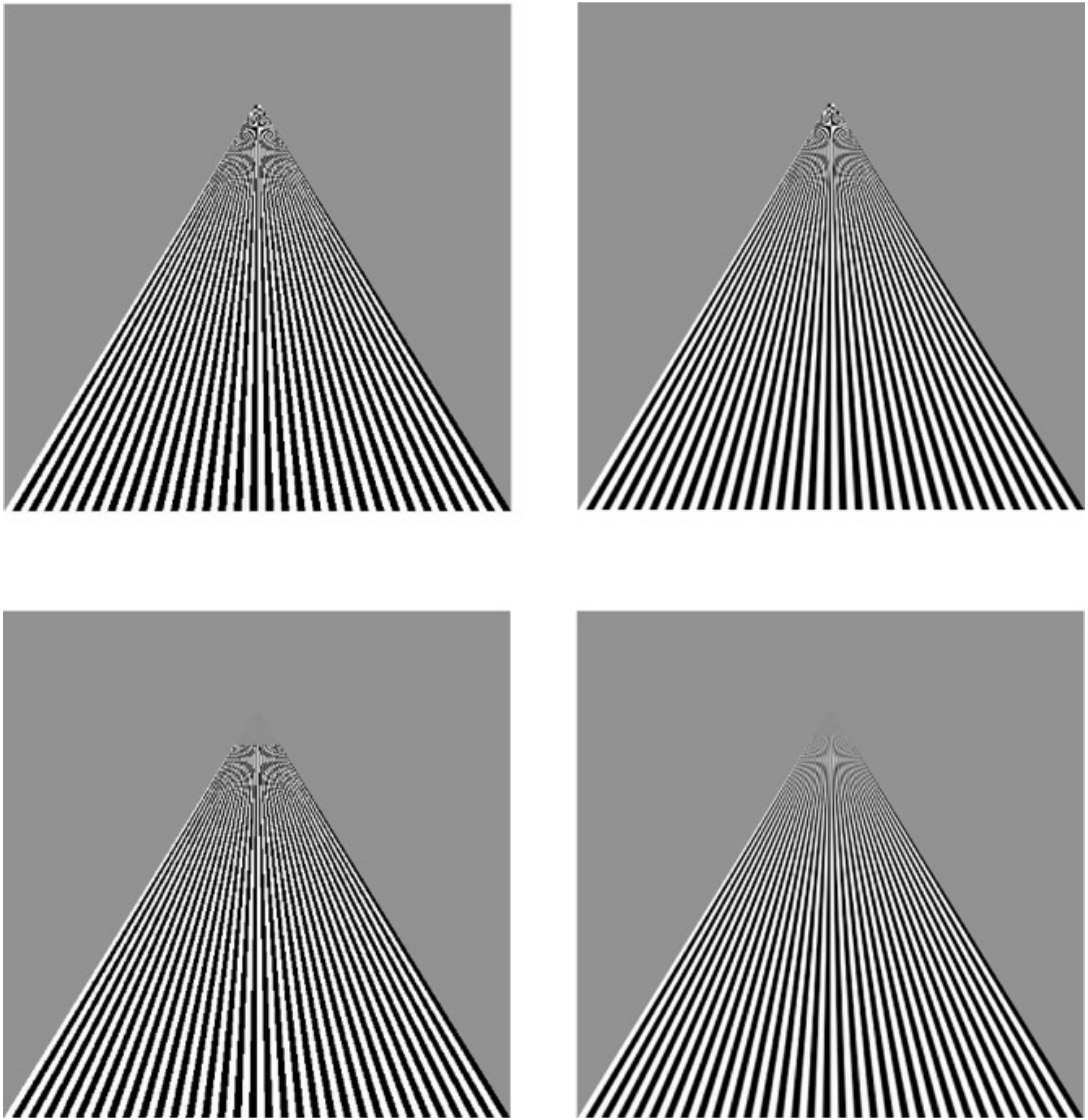
128×128, 64×64, 32×32, 16×16, 8×8, 4×4, 2×2, 1×1

Reduce minification artifacts

Prefilter the texture to obtain reduced resolutions

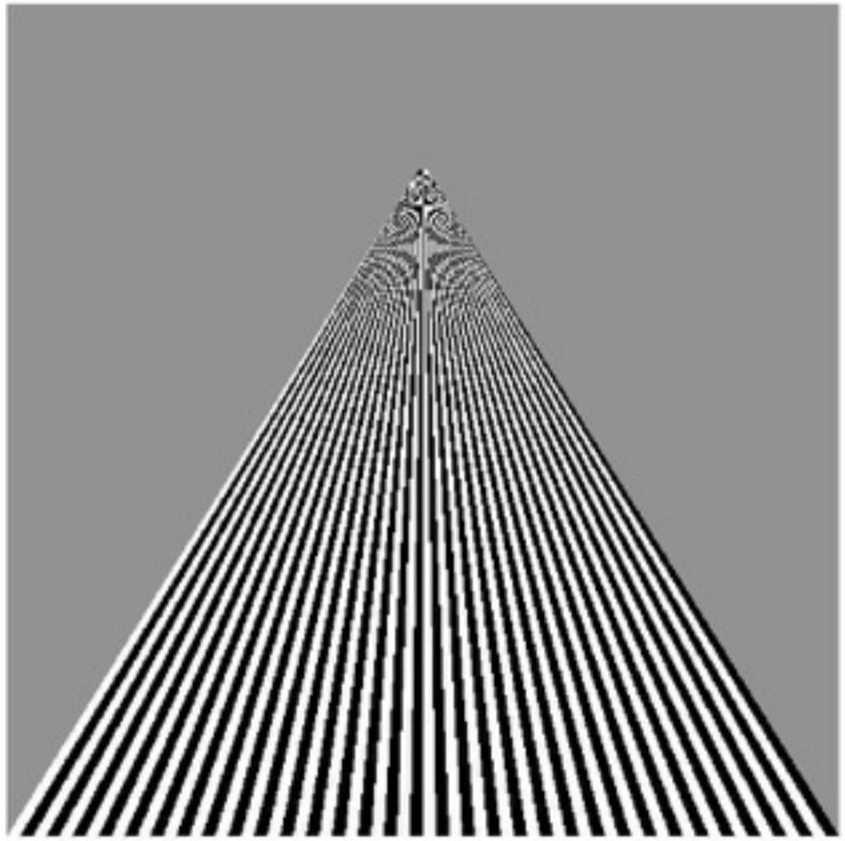
Requires 1/3 more space

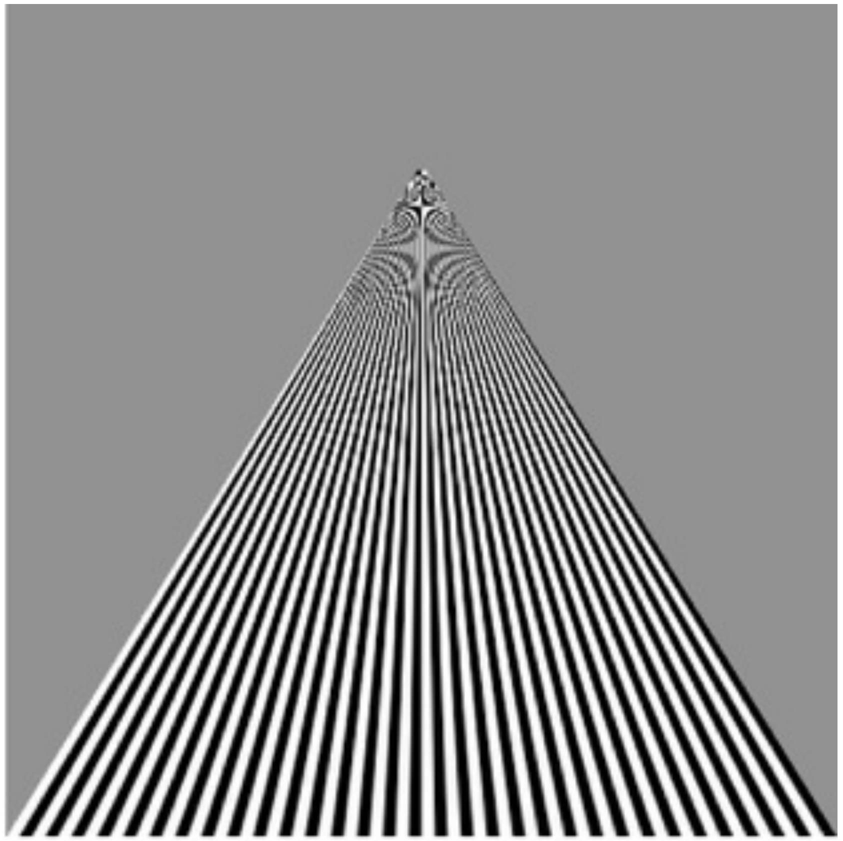
Get a texture hierarchy indexed by level



[Angel and Shreiner]

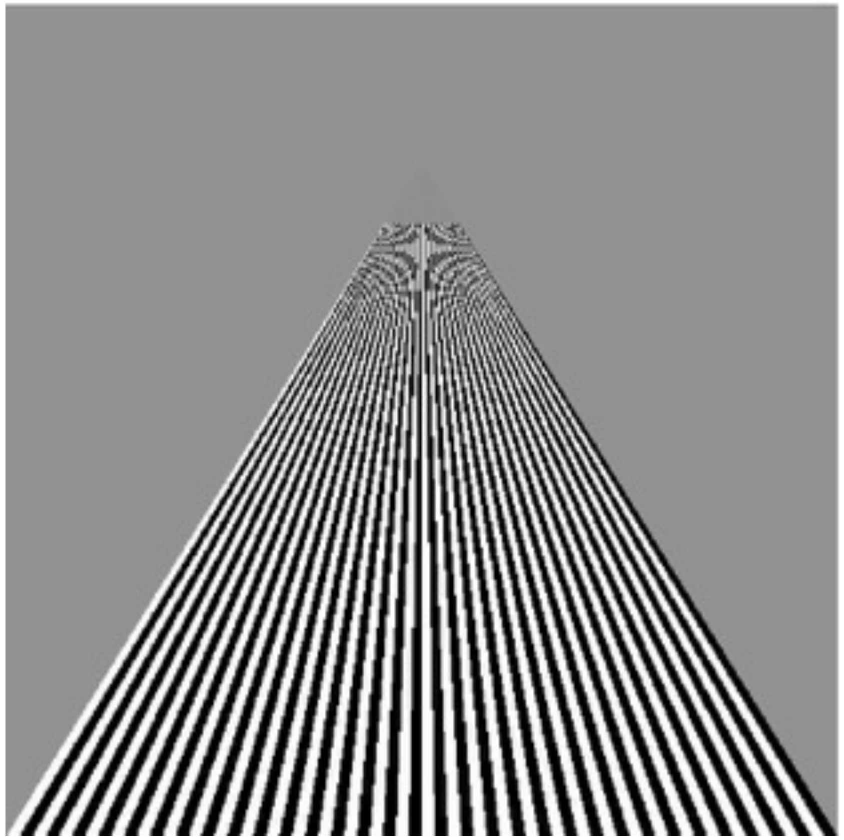
point
sampling

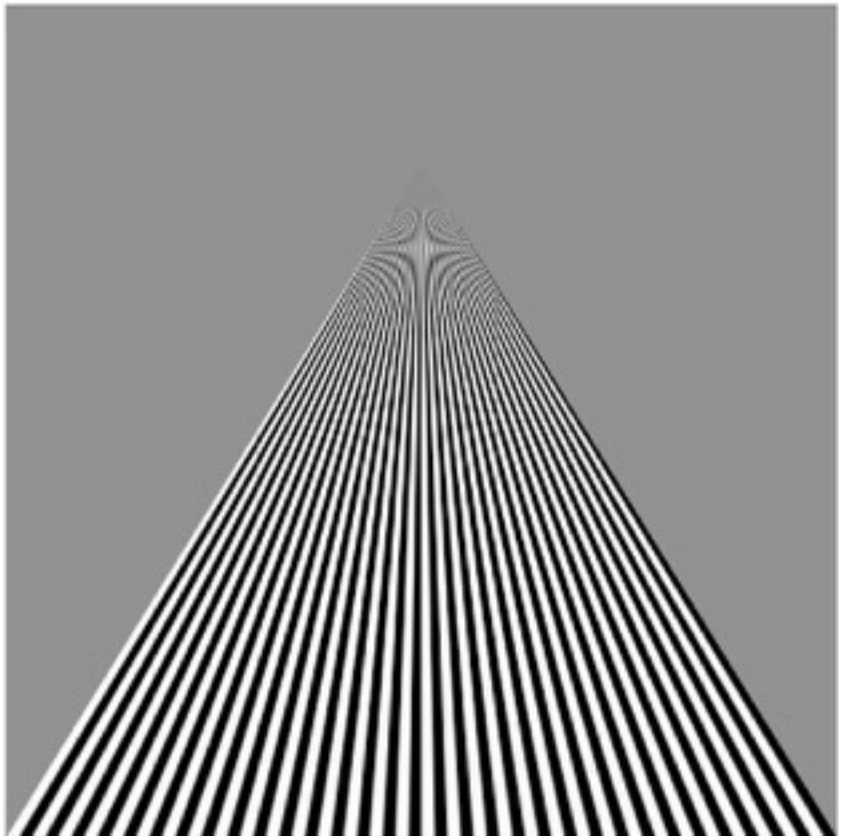




linear
filtering

mipmapped
point
sampling





mipmapped
linear
filtering

[Angel and Shreiner]