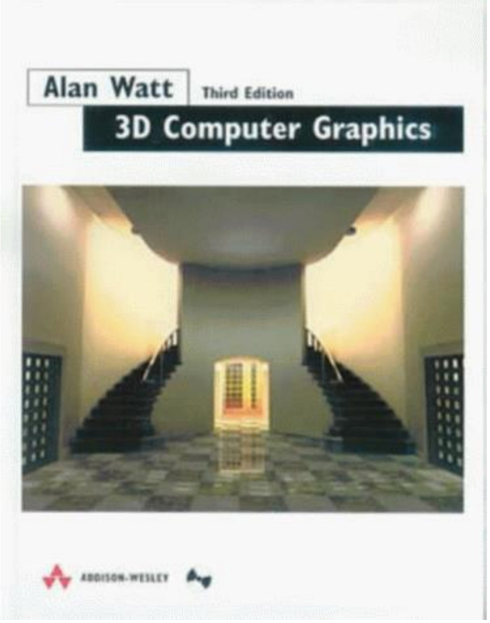
COM3503/4503/6503: 3D Computer Graphics

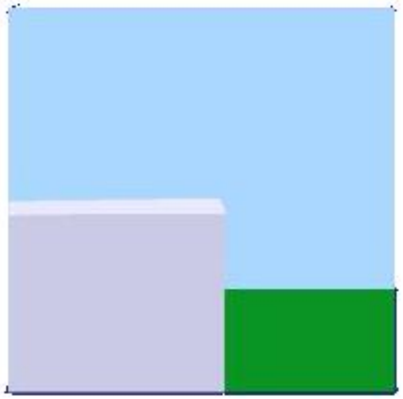
Lecture 10: Texture: part 1

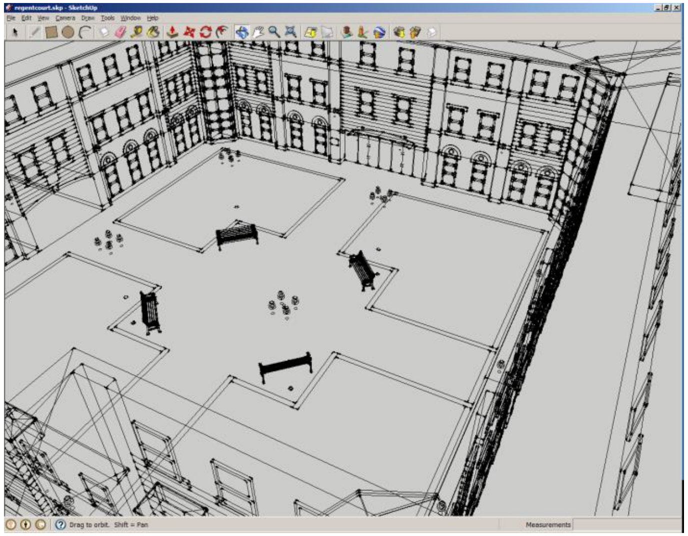


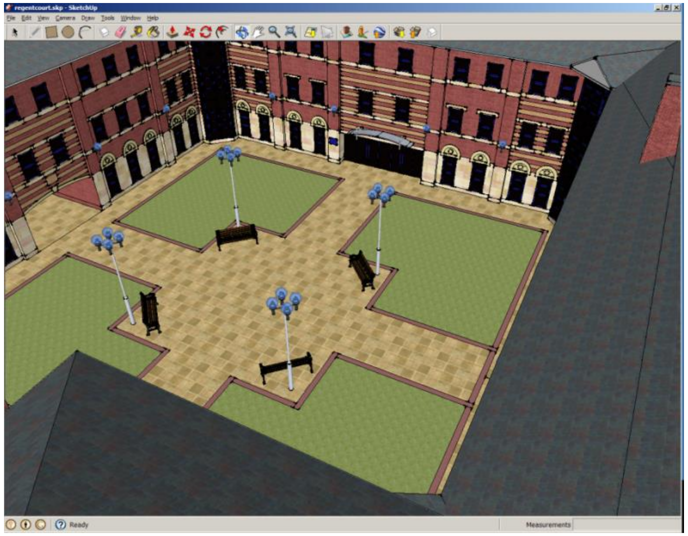
Dr. Steve Maddock
Room G011, Regent Court
s.maddock@sheffield.ac.uk

1. Introduction

- Texture mapping applies a pattern of colour to an object – texture is not ‘texture’!!
- 2D texture mapping pastes a bitmap picture (texture) onto the surface of an object
- Endless supply of digital photographs

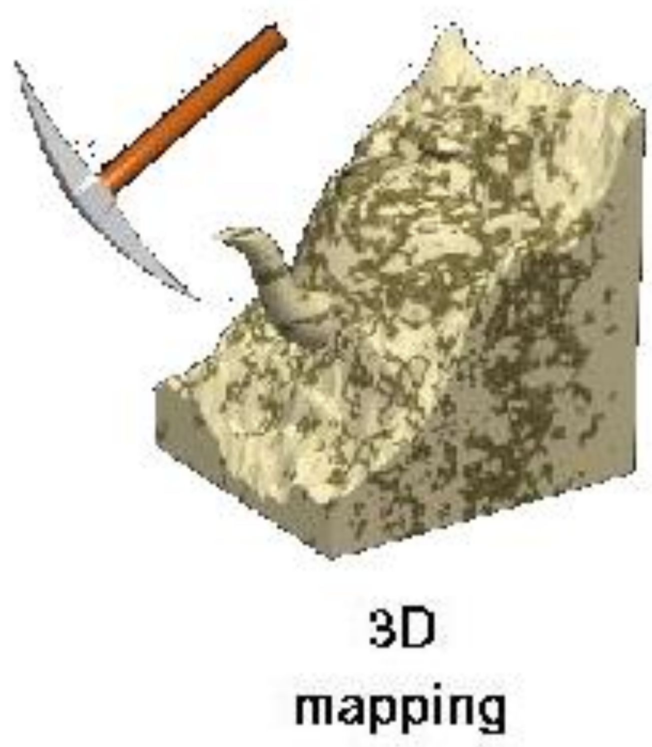






1.1 2D and 3D texturing

- 2D texture (colour) mapping pastes a bitmap picture (texture) onto the surface of an object – today's lecture
- 3D texture mapping places an object inside a 3D pattern (often represented as a procedure) – see later lecture.

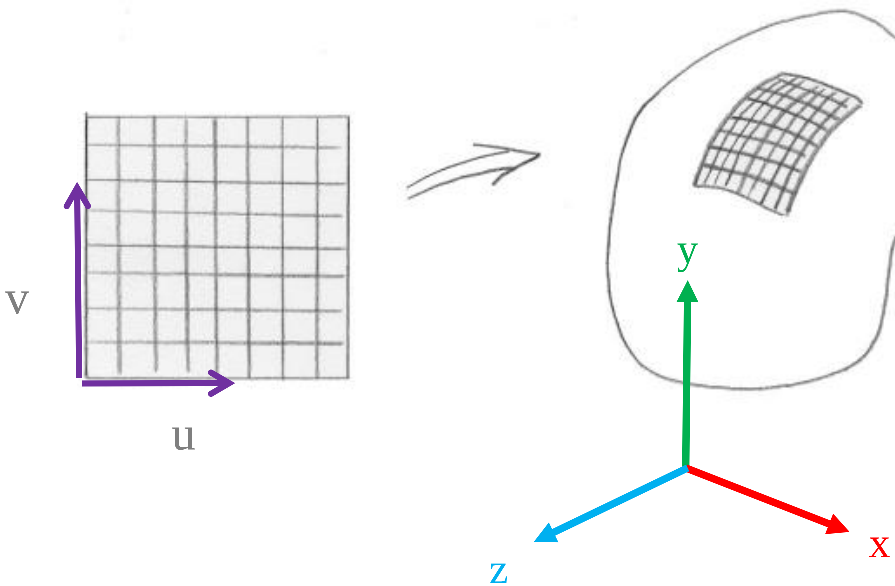


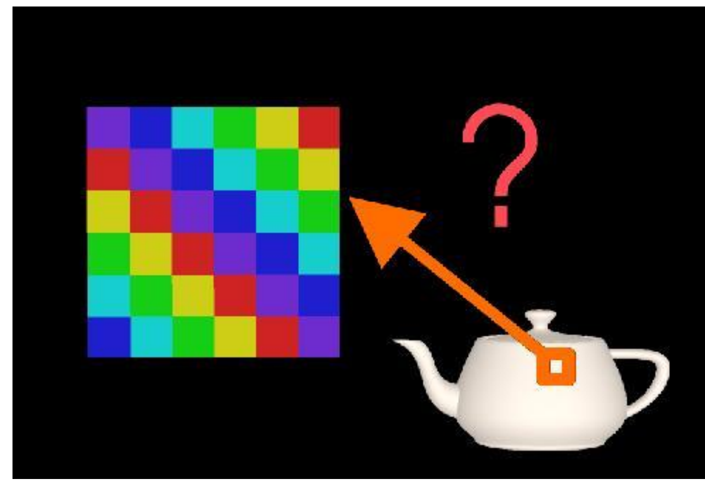
1.2 Texture as detail

- For object detail, it is cheaper to use textures rather than geometry
  - E.g. brick texture rather than model individual bricks
- Can also make it appear as if the surface has bumps that respond to the lighting by using a texture map
  - see later lecture (bump mapping)

1.3 Texture mapping considerations

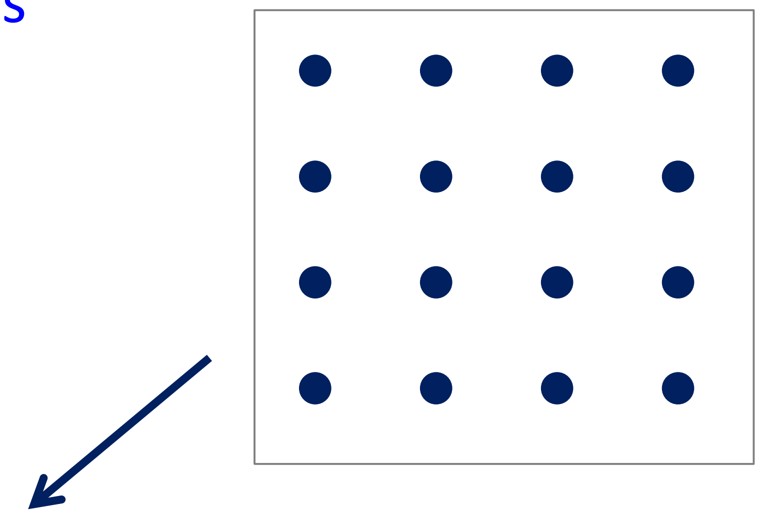
- How to map a 2D texture space (u,v) to a 3D object (x,y,z), especially one defined by polygons

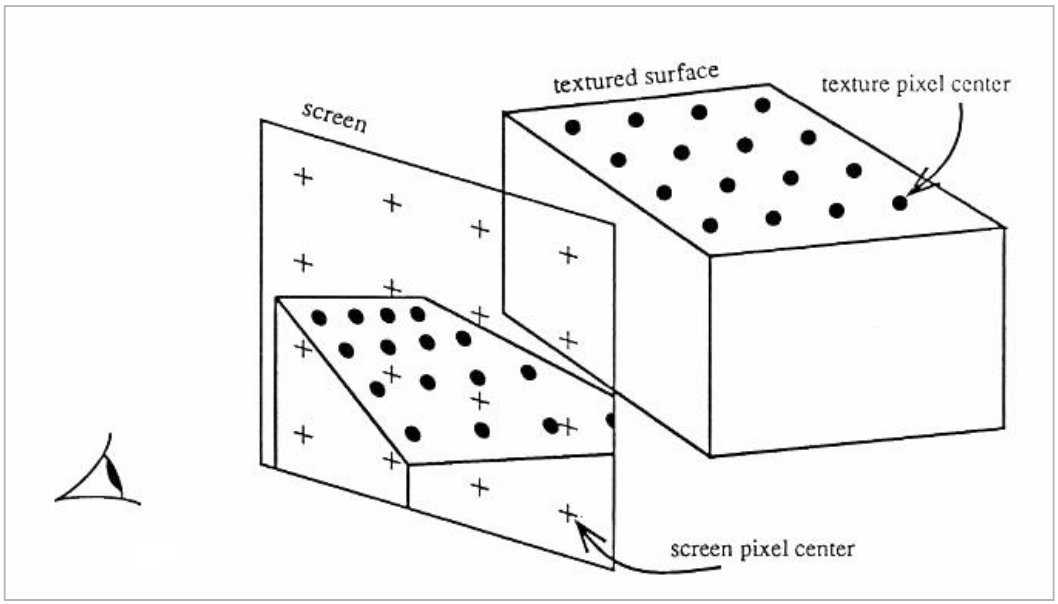


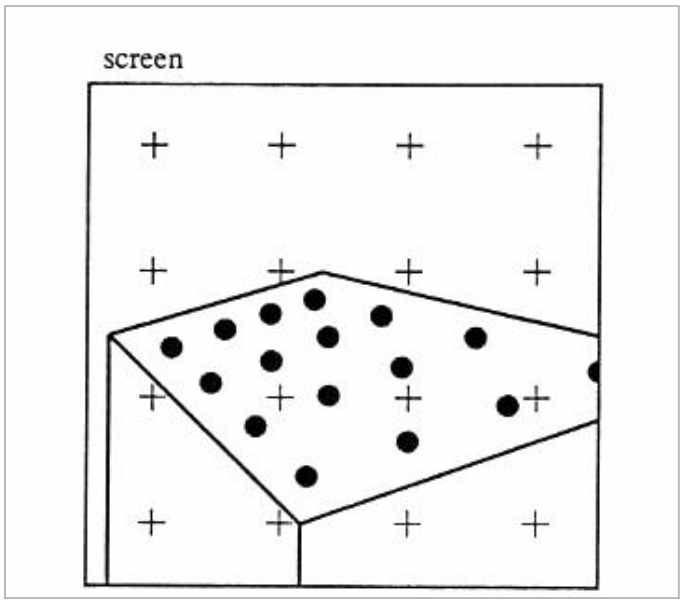


1.3 Texture mapping considerations

- 2D texture mapped onto object
- Object projected onto 2D screen
- How to sample?

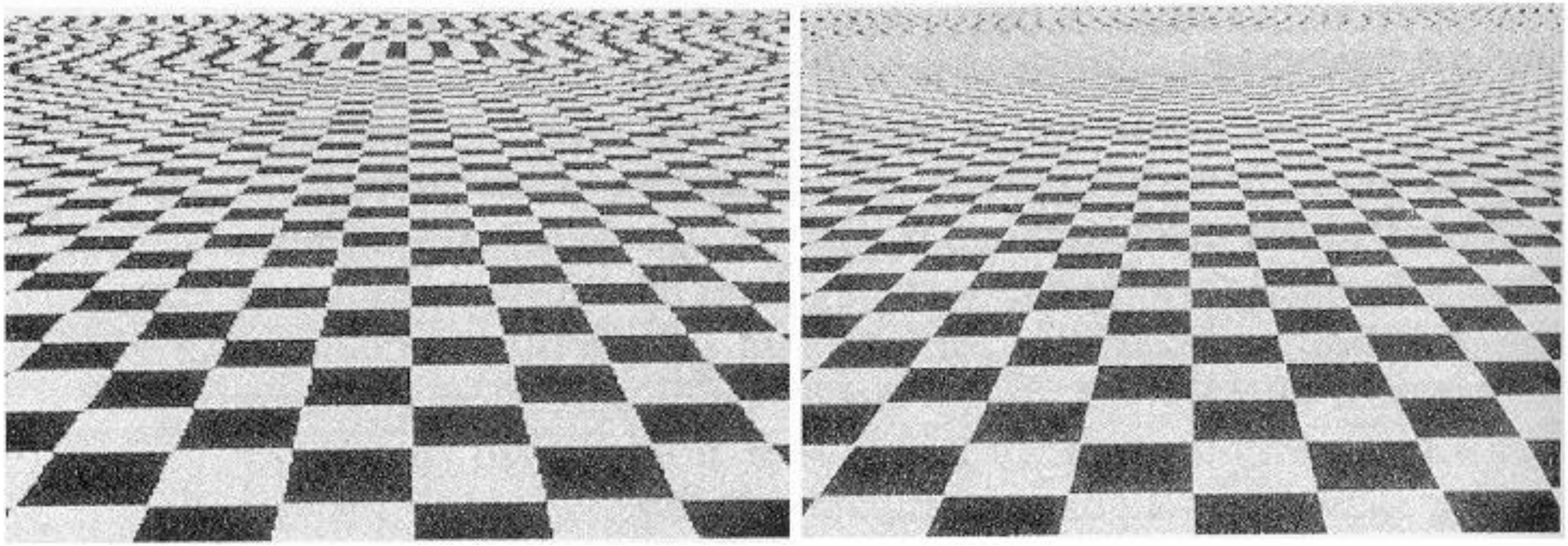






1.3 Texture mapping considerations

- Anti-aliasing problems: coherence and compression.
  - E.g. Chequerboard pattern.



1.4 Aspects to modulate

Replace or modulate k_d

- Surface colour (diffuse reflection coefficient):

$$I_{r,g,b} = k_a I_a + I_i \left(k_d \, T_{r,g,b}(u,v) \, (\mathbf{L.N}) + k_s \, (\mathbf{N.H})^n\right)$$

- Specular 'colour':
    - Environment mapping (view dependent) makes it looks as if an object is reflecting its environment.
- Normal vector perturbation:
    - Bump mapping fools the eye into thinking the geometry has altered.
- Displacement mapping:
    - Use a height field to perturb geometry (thus changing the normal)
- Transparency:
    - Control the opacity of an object, e.g. etched glass.

1.4 Aspects to modulate

Replace or modulate k_d

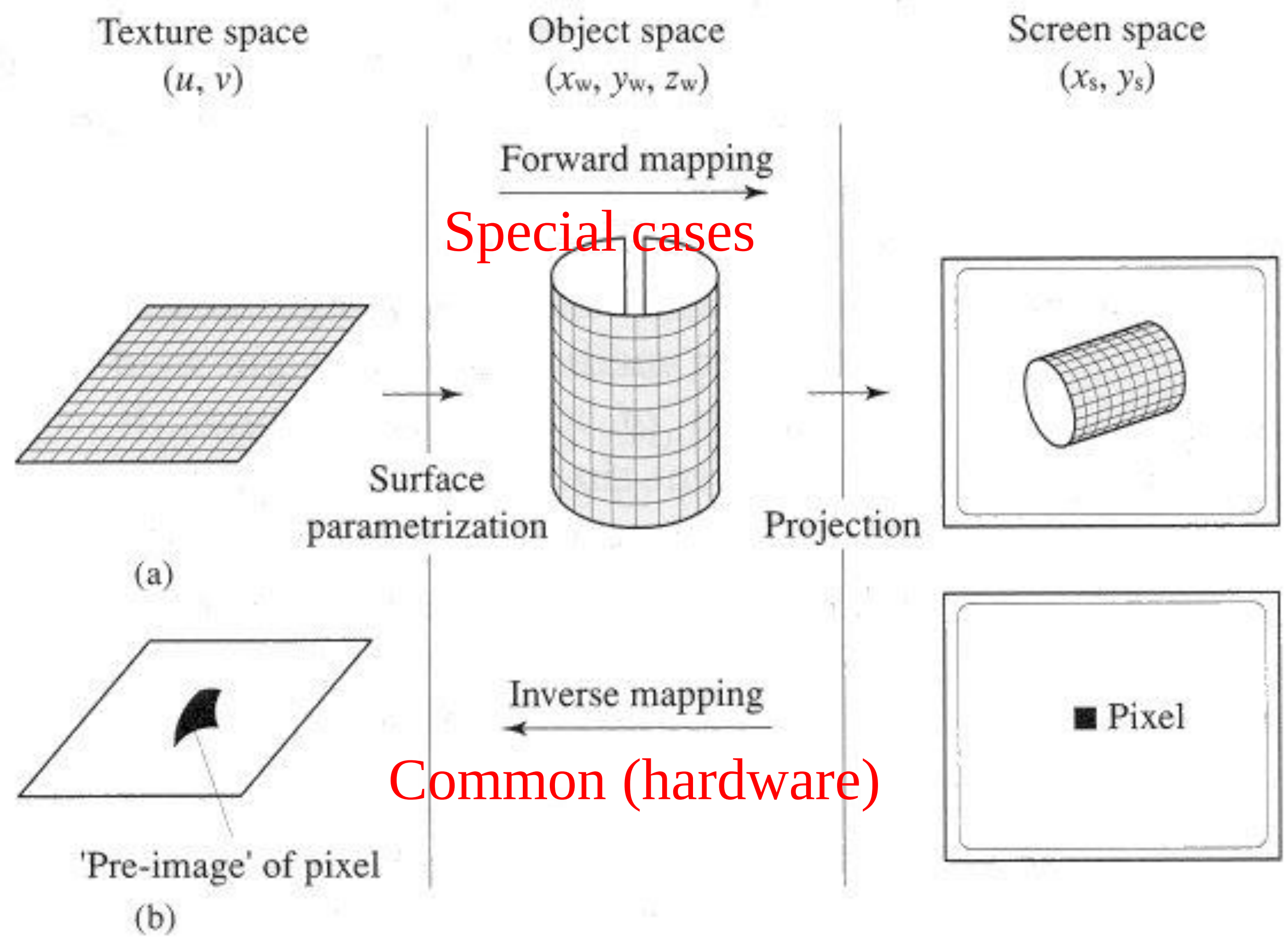
- Surface colour (focusing on the diffuse reflection coefficient):

$$I_{r,g,b} = k_a I_a + I_i \left(k_d\, T_{r,g,b}(u,v)\, (\mathbf{L.N}) + k_s\, (\mathbf{N.H})^n\right)$$

  - Alternative names: texture colour mapping, colour mapping, diffuse colour mapping, uv mapping
- Texture mapping, confusingly, is often the generic name for all the different approaches
- Specular 'colour':
  - Environment mapping (view dependent) makes it looks as if an object is reflecting its environment.
- Normal vector perturbation:
  - Bump mapping fools the eye into thinking the geometry has altered.
- Displacement mapping:
  - Use a height field to perturb geometry (thus changing the normal)
- Transparency:
  - Control the opacity of an object, e.g. etched glass.

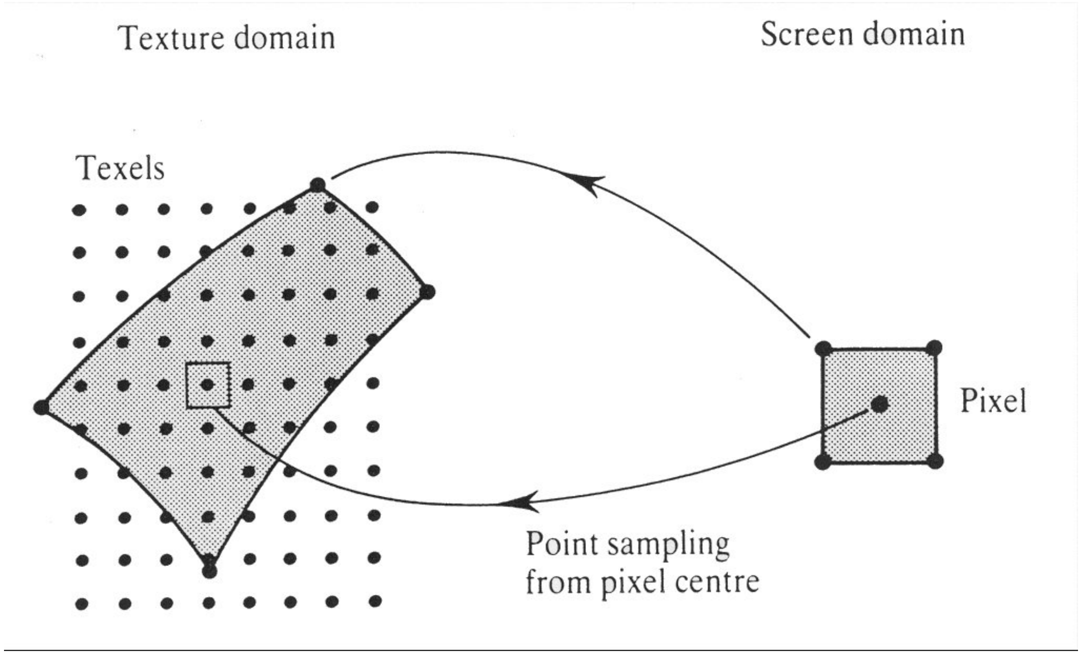
2. 2D texture (colour) mapping

- An image warp



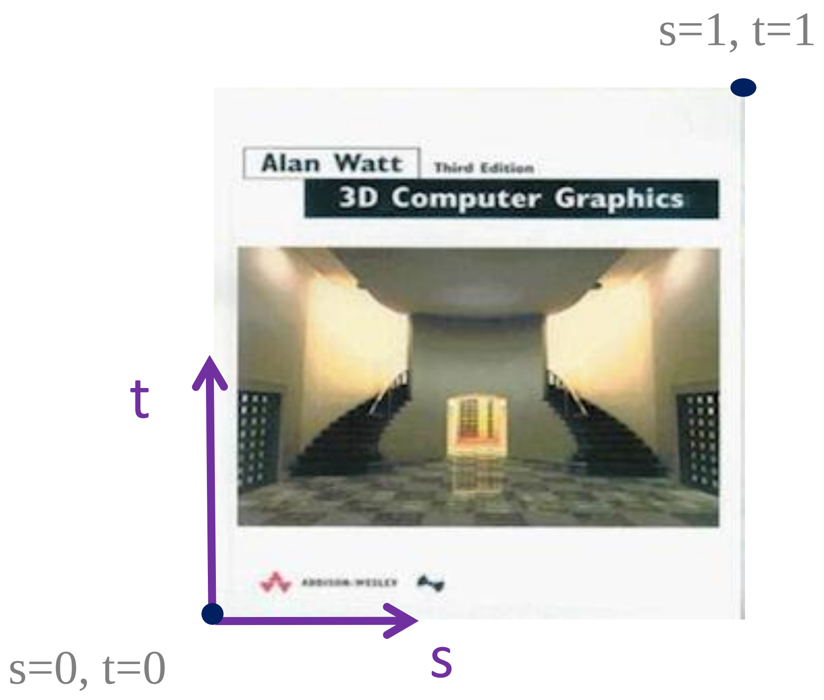
3. Inverse mapping by bilinear interpolation

- Most common approach is inverse mapping
  - Common in hardware
- For a pixel in screen space, find its pre-image in texture space and filter the information



3.1 A texture coordinate space is defined

- The texture (e.g. a digital image) is parameterised
- 0<=u<=1, 0<=v<=1

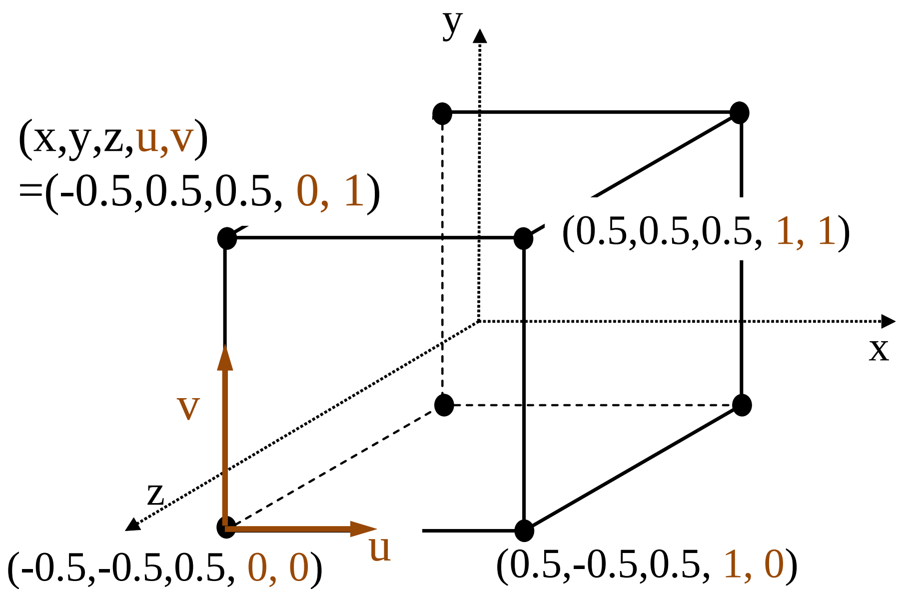


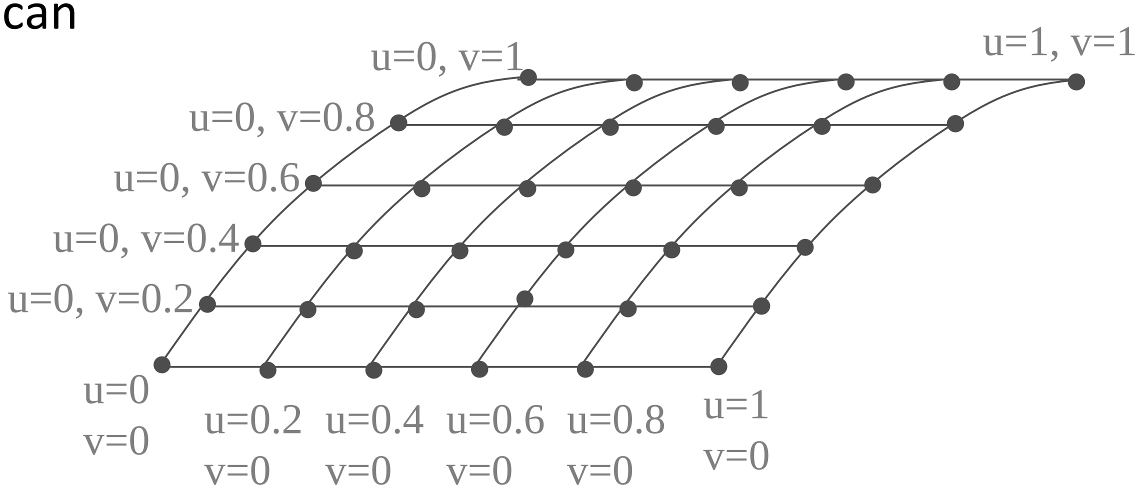
3.2 Vertices are labelled with (u,v) coordinates

- During modelling, each polygon vertex is given texture coordinates (u,v)

Examples:

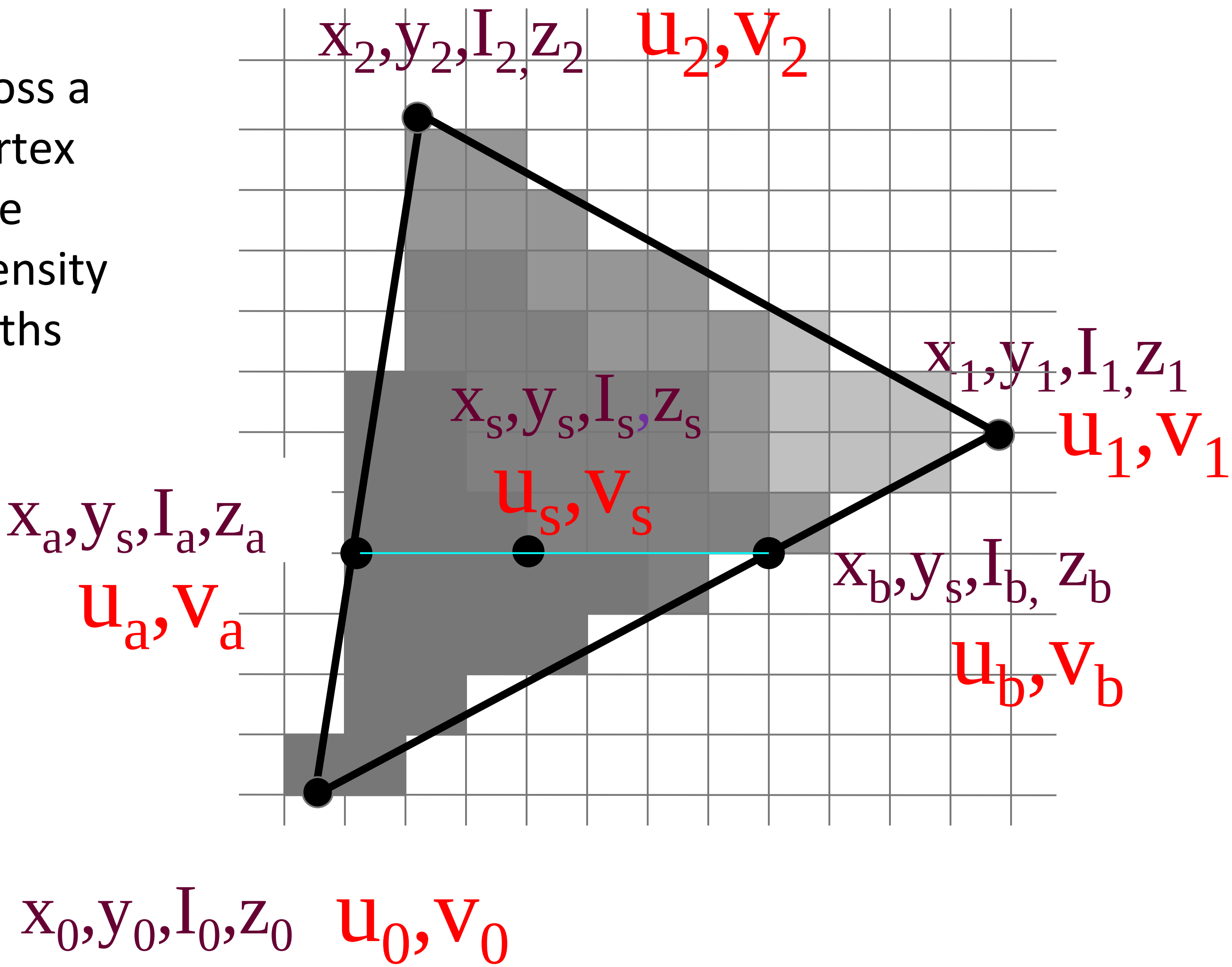
- Each face of a cube is given separate (u,v) parameterisations
- A curved surface made up of polygon facets can also be given a (u,v) parameterisation



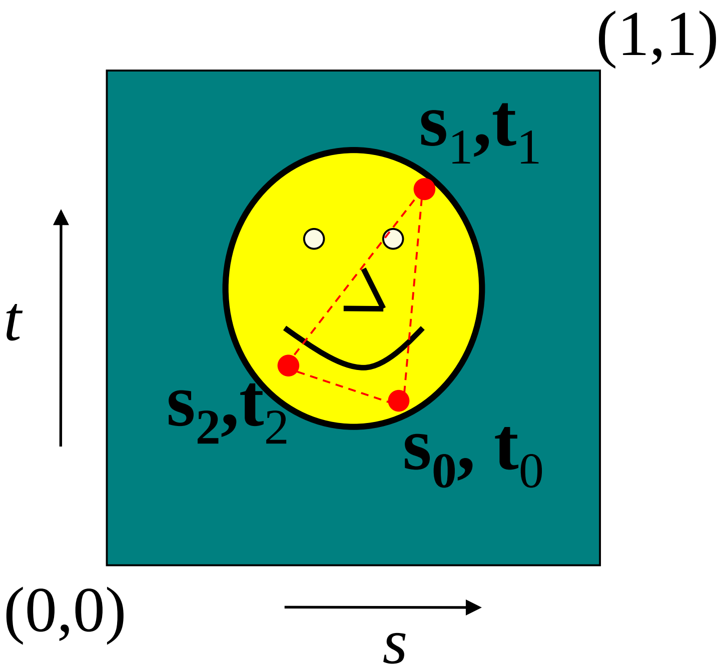


3.3 Interpolation

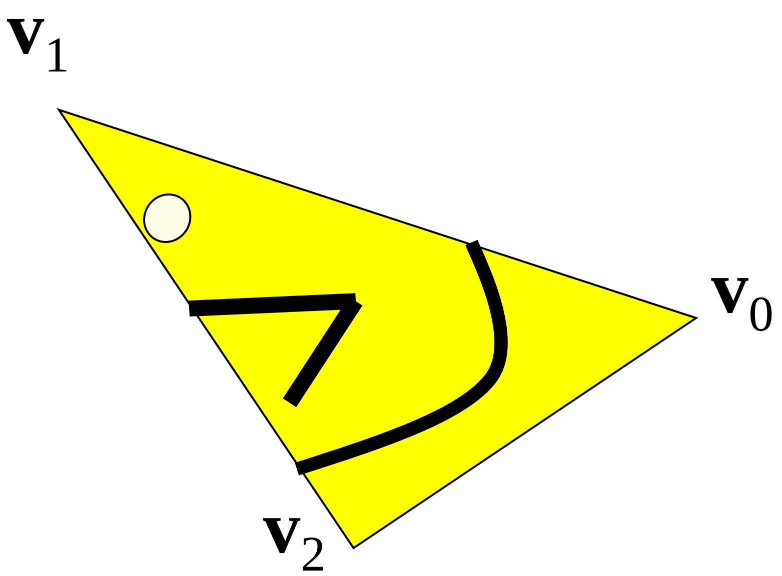
- u and v are interpolated across a polygon from vertex (u,v) values in the same way as intensity values and z depths



- Here (s,t) is used instead of (u,v)



Texture Space

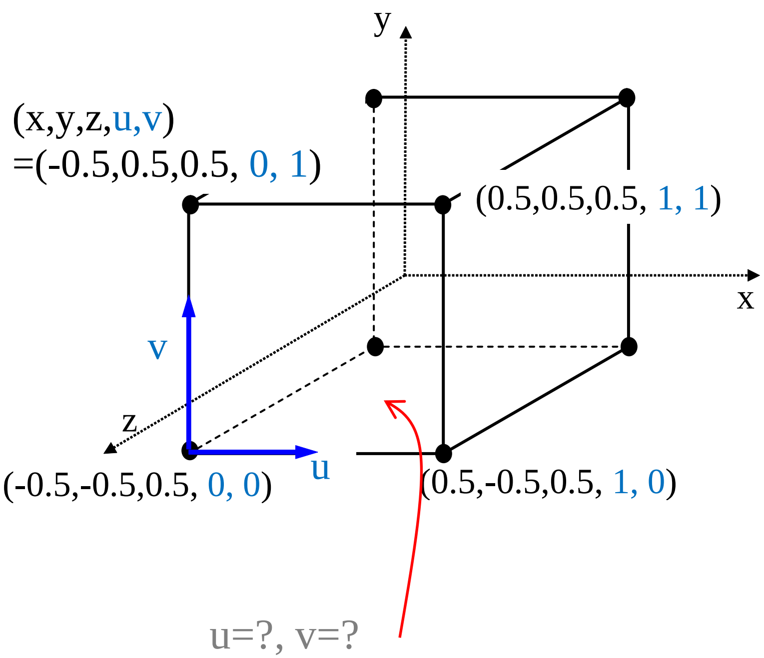


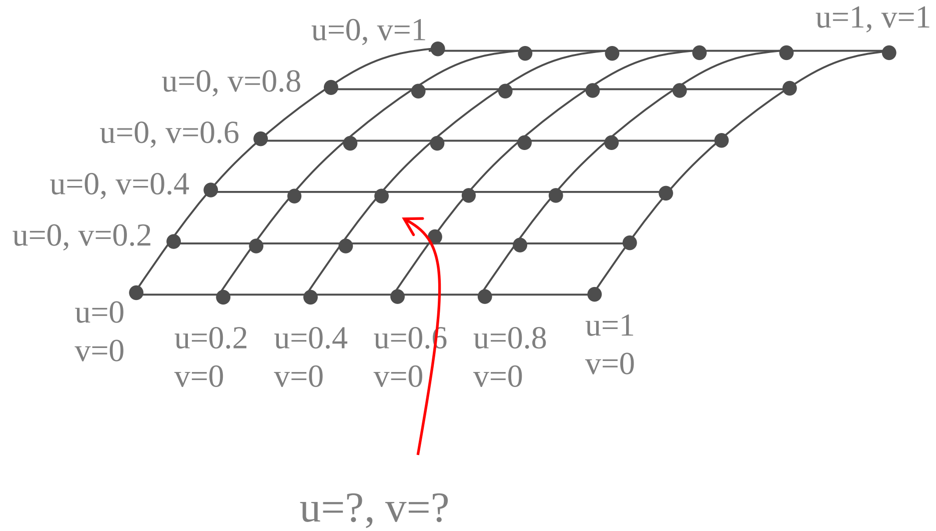
Triangle, with vertices v0,v1,v2

After (Rotenberg, 05, graphics.ucsd.edu/courses/cse167_f05/

3.3 Interpolation

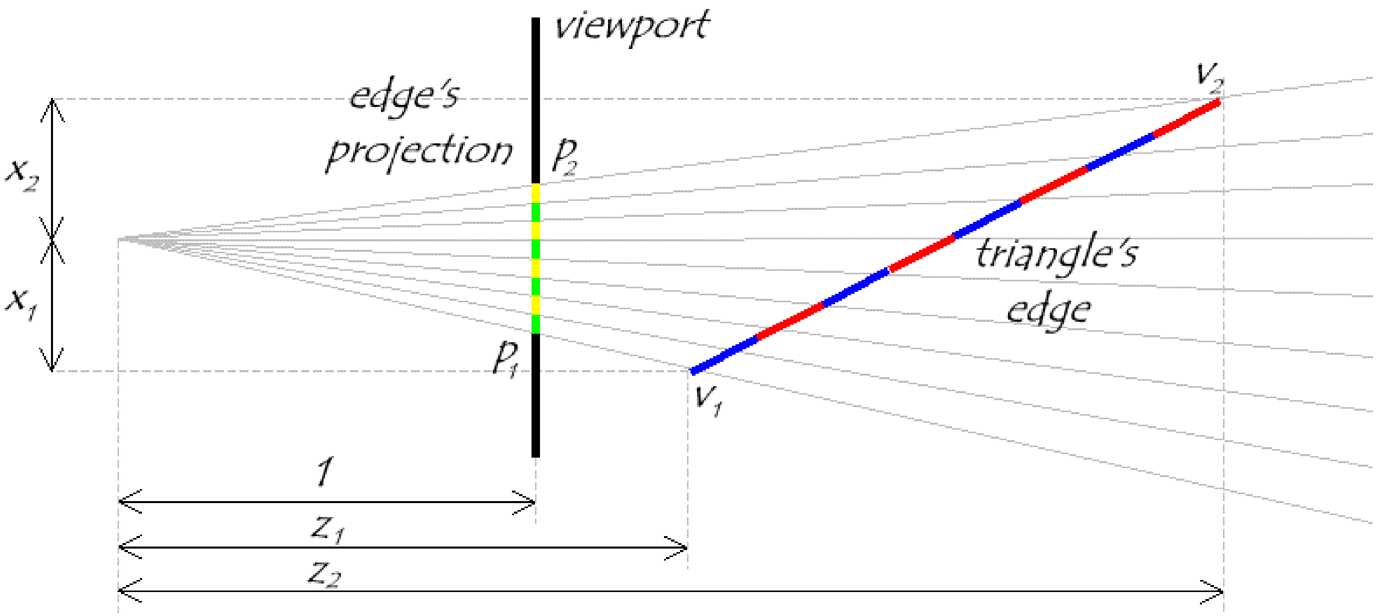
- Interpolate homogeneous coordinates (u',v',q): u'=uq, v'=vq, q=1/z
- (u,v) values are found for a pixel using u=u'/q, v=v'/q



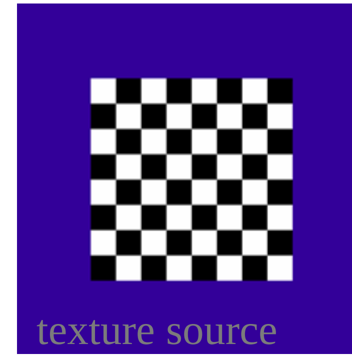


3.4 Why we need perspective correct interpolation

- Linear interpolation in screen space gives incorrect results
- Uniform steps on the image plane do not correspond to uniform steps along the edge

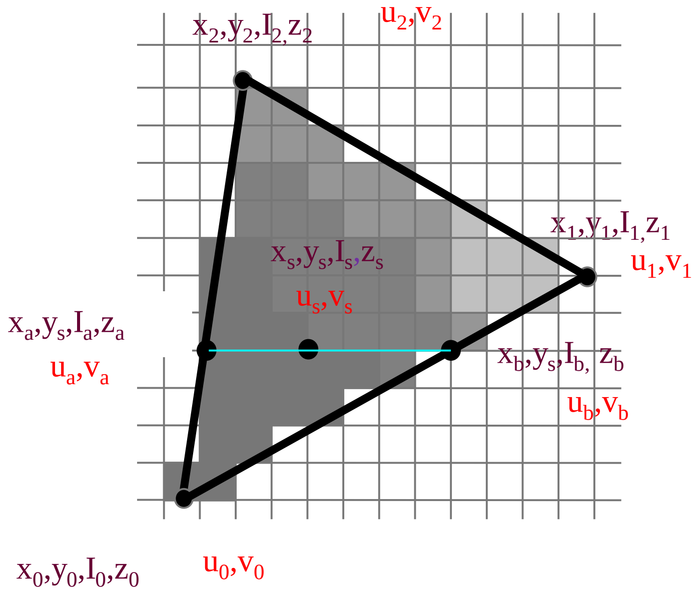


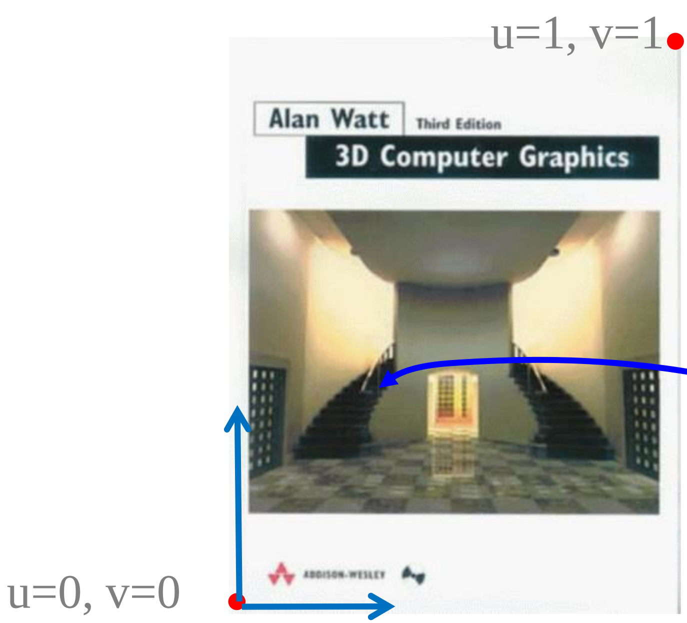
Durand, 06

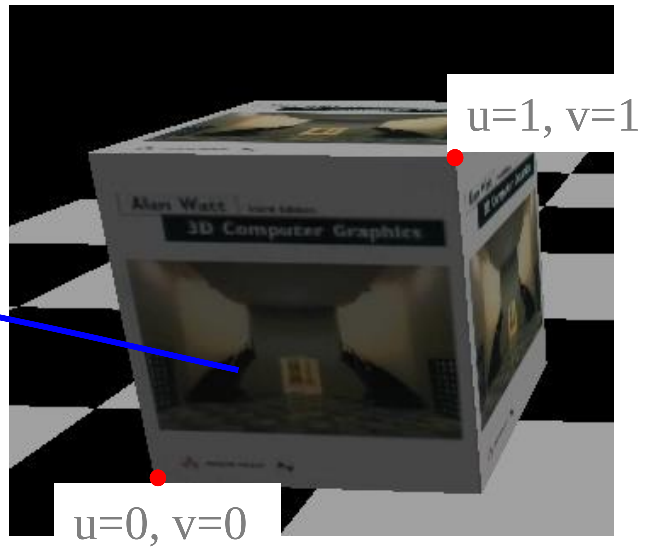


3.5 Lookup

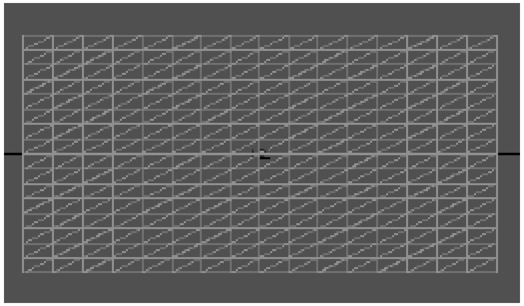
- Interpolated (u, v) is used to lookup value from texture map
- Easy to incorporate this interpolation into a z buffer

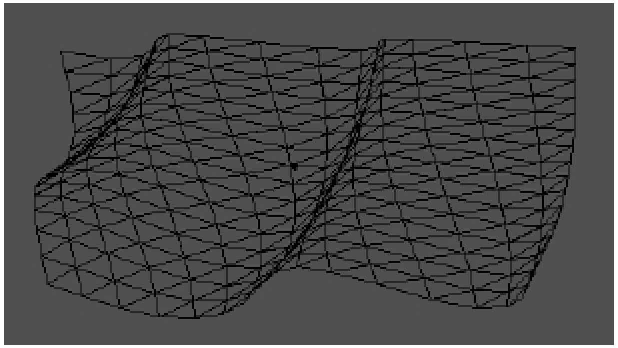






3.6 Animation example

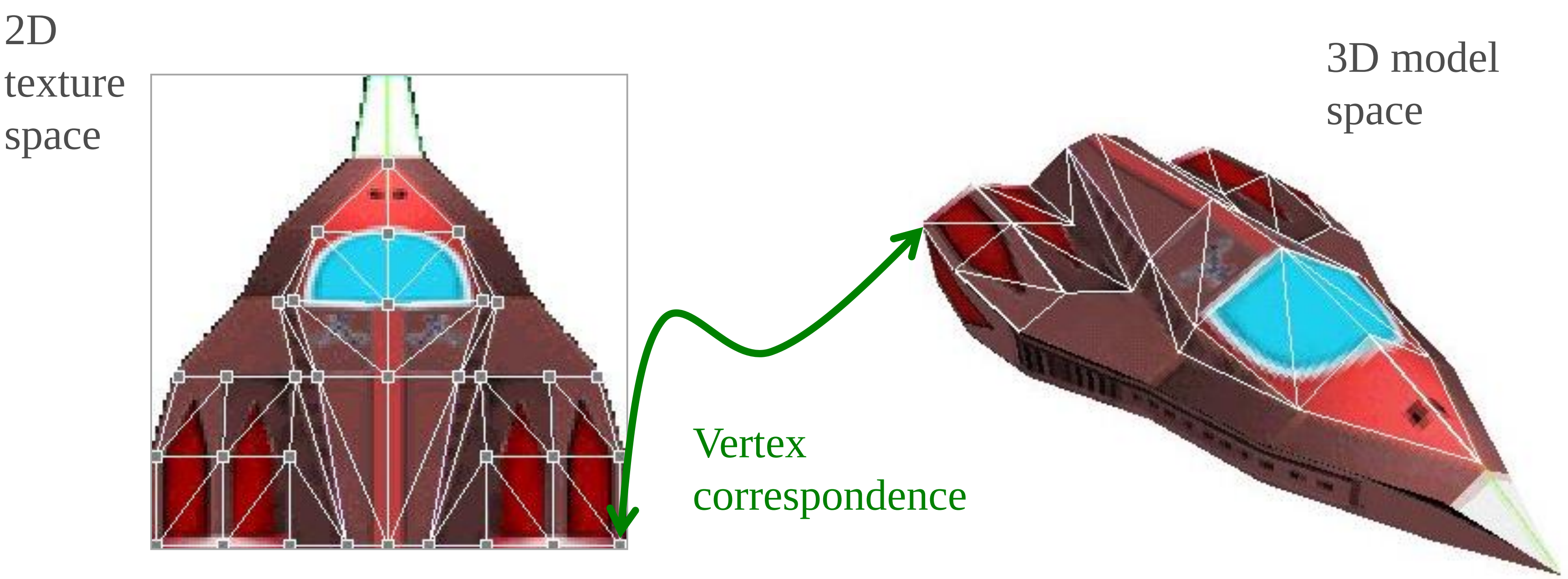




intuitionbase.com/waveguide/tut6.html

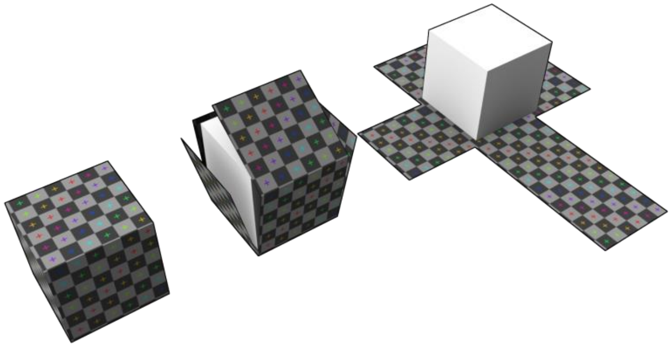
3.7 Practical system

- Artist moves 2D texture vertex and sees result in 3D model space
- Can be difficult to get this right without stretching the texture, which can produce visual artefacts

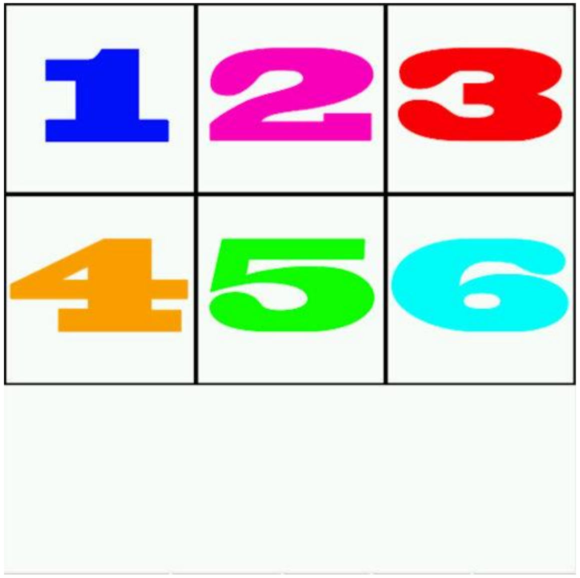


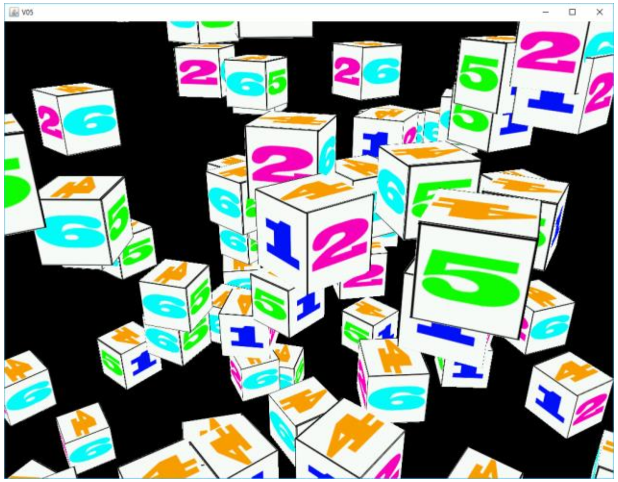
4. Complex objects

- To help the artist specify the uv mapping:
  - can unwrap the object mesh
  - artist can paint the individual areas



Zephyris at en.wikipedia [CC-BY-SA-3.0 (http://creativecommons.org/licenses/by-sa/3.0) or GFDL (http://www.gnu.org/copyleft/fdl.html)], from Wikimedia Commons





Levy et al, Proc. Siggraph 02

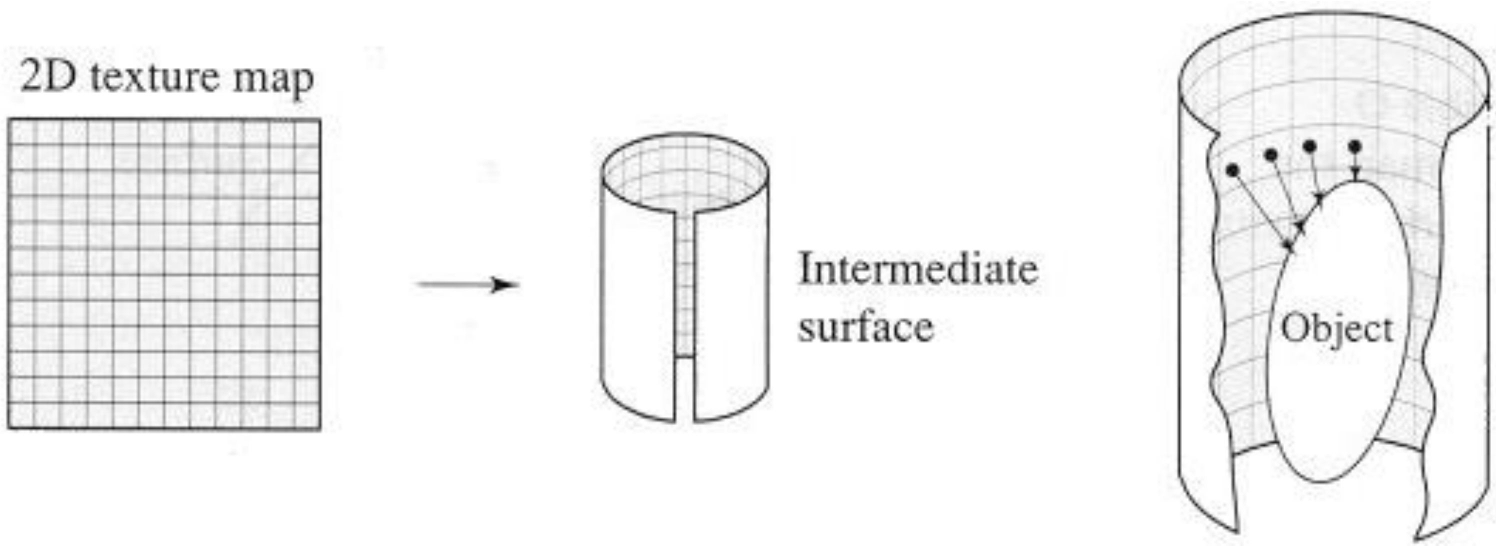
4. Complex objects

- To help the artist specify the uv mapping:
    - create separate pieces joined at seams (store pieces in texture atlas)
    - artist can paint the individual areas

Levy et al, Proc. Siggraph 02

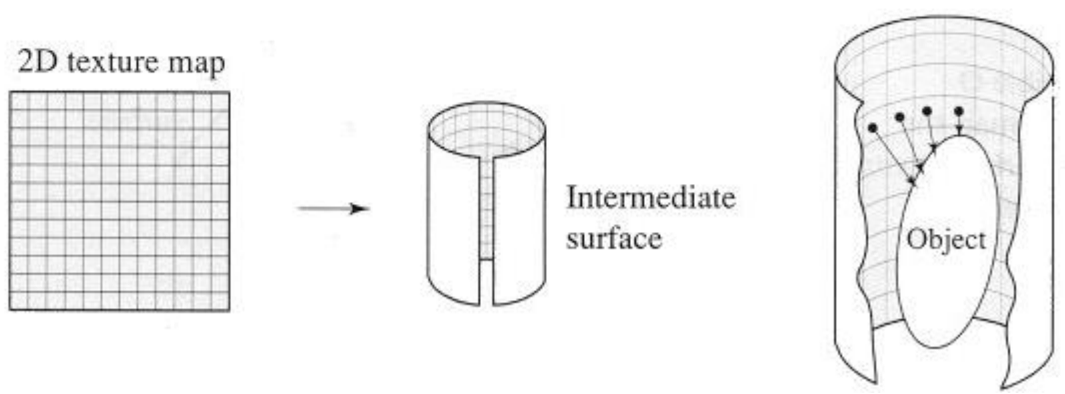
5. Two-stage mapping: Inverse mapping using an intermediate surface

- Some polygon surfaces are difficult to parameterise, i.e. very convoluted
- Can waste a lot of artist time manually defining u and v for vertices
- Instead, use a two-stage process (Bier and Sloan, 1986)
  - Stage 1 – map to an intermediate that is easy to parameterise
  - Stage 2 – map the intermediate to the object



5. Two-stage mapping: Inverse mapping using an intermediate surface

- *Stage 1*: Map T(u,v) onto an 'easy' intermediate surface, e.g. a cylinder
  - 'Easy' because it is an analytical function
  - Known as the S mapping
  - $T(u,v) \rightarrow T'(x_i,y_i,z_i)$
- *Stage 2*: Map the three dimensional texture onto the object: $R_3 \rightarrow R_3$
  - $T'(x_i,y_i,z_i) \rightarrow O(x_w,y_w,z_w)$
  - Known as the O mapping
  - There are a number of variations for the O mapping



5.1 Stage 1 surfaces

- A plane at any orientation
- The curved surface of a cylinder
- The faces of a cube
  - most common – see environment mapping
- The surface of a sphere

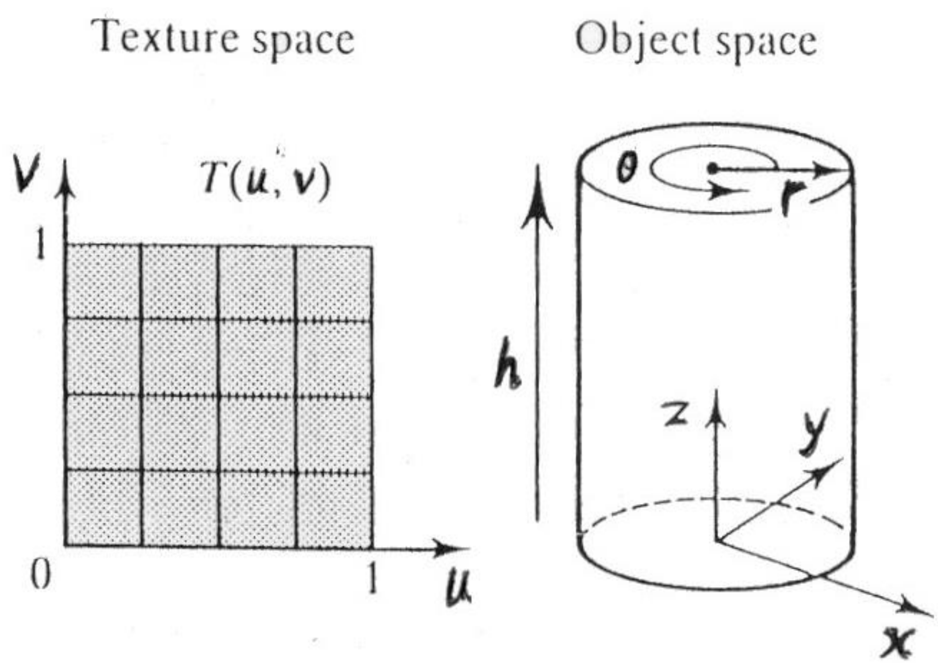
Example: A cylinder

$(x,y,z) = (r \cos\theta, r \sin\theta, hz)$

where

$0 \le \theta \le 2\pi, \; 0 \le z \le 1$

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



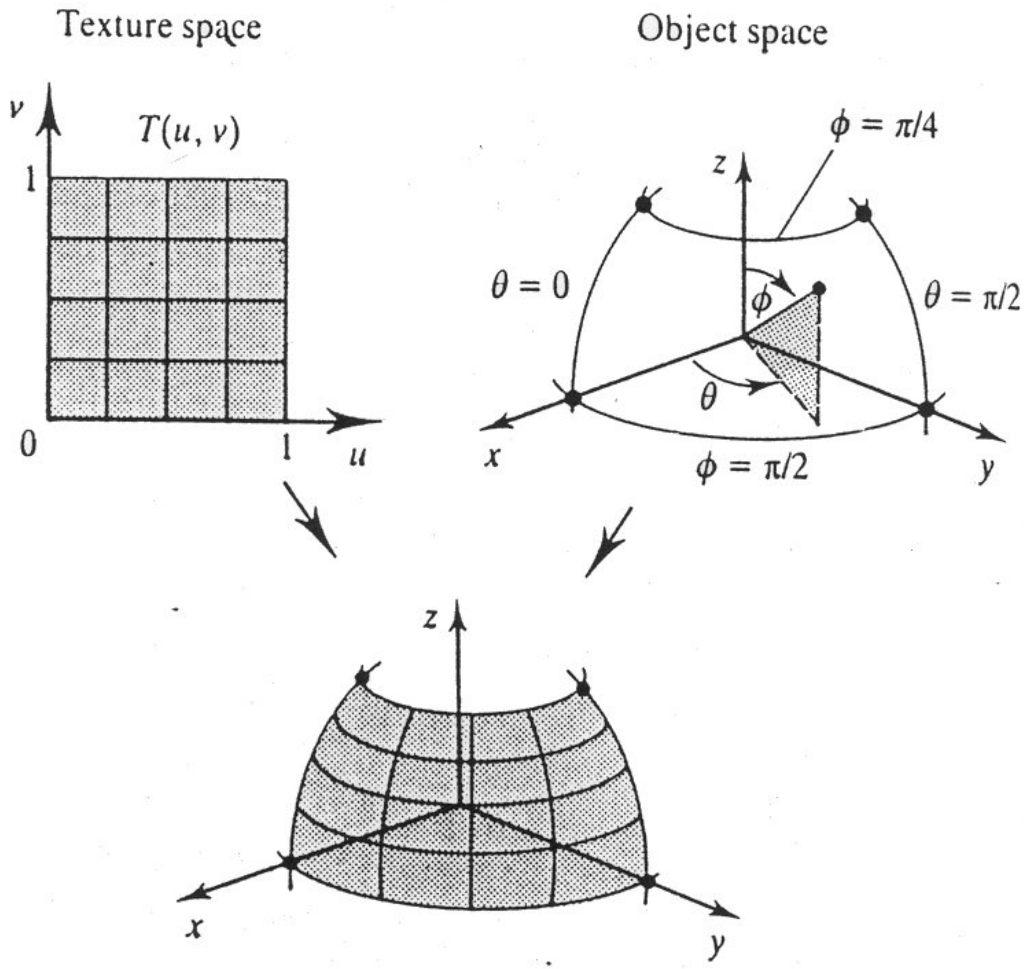
Example: Part of a sphere

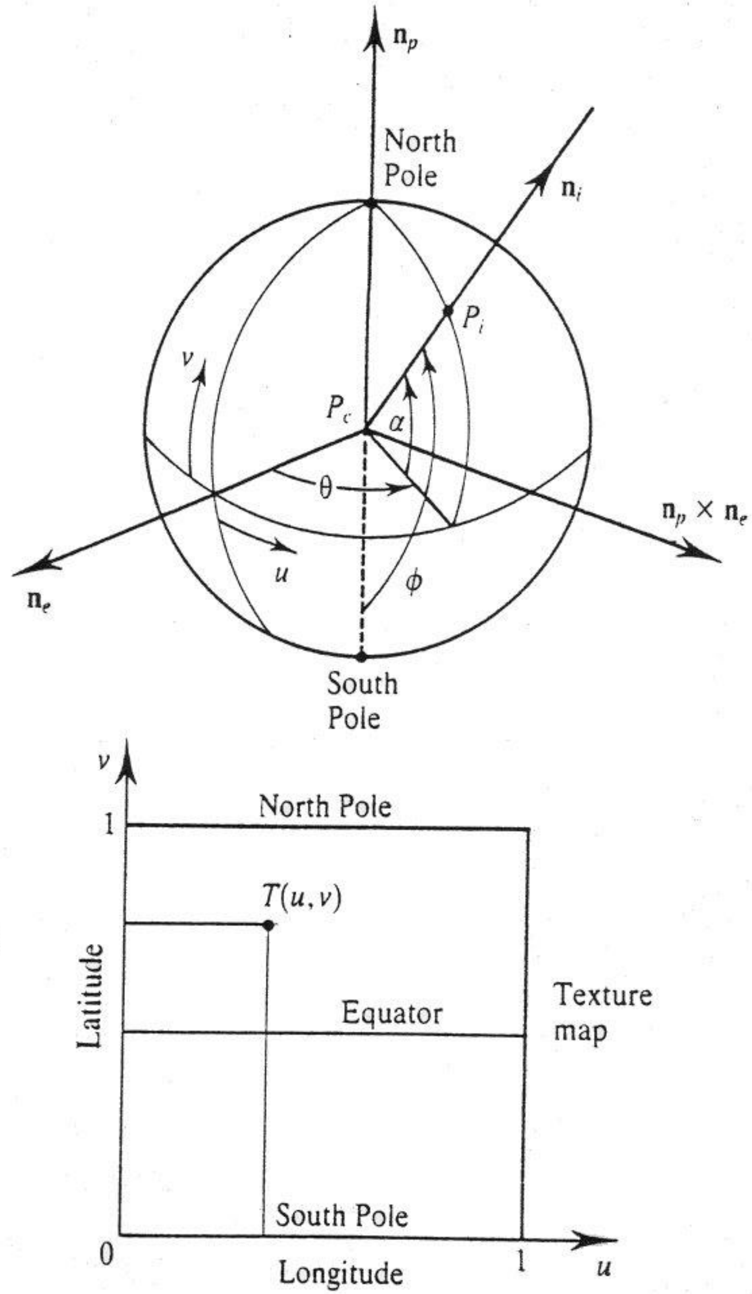
$$(x,y,z) = (r\cos\theta\sin\phi,\; r\sin\theta\sin\phi,\; r\cos\phi)$$

where

$$0 \le \theta \le \pi/2, \quad \pi/4 \le \phi \le \pi/2$$

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

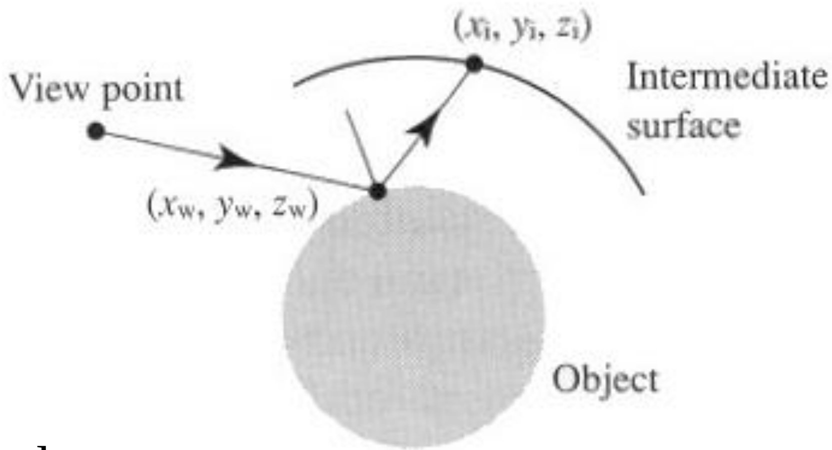




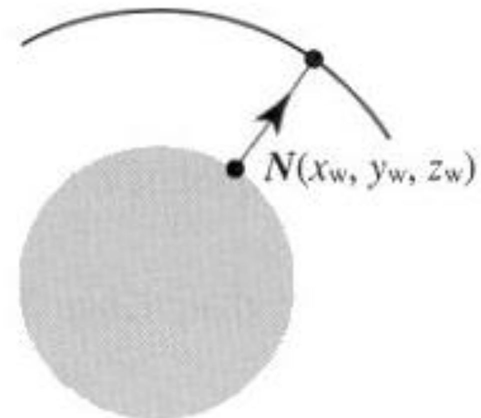
$\mathbf{n}_p$
North Pole
$\mathbf{n}_i$
P_i
v
P_c
α
θ
$\mathbf{n}_p \times \mathbf{n}_e$
u
ϕ
$\mathbf{n}_e$
South Pole
v
1
North Pole
$T(u, v)$
Latitude
Equator
Texture map
South Pole
0
Longitude
1
u

5.2 Stage 2: The four O mappings

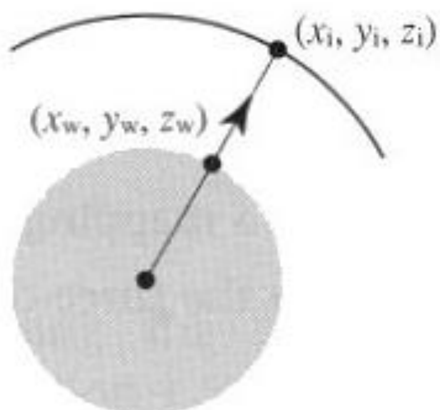
Examples using a sphere



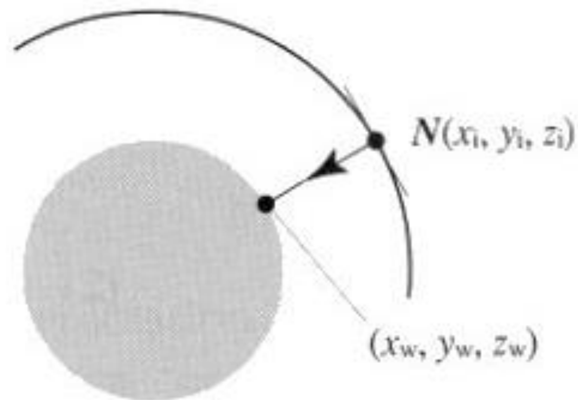
(1) Reflected ray



(2) Object normal

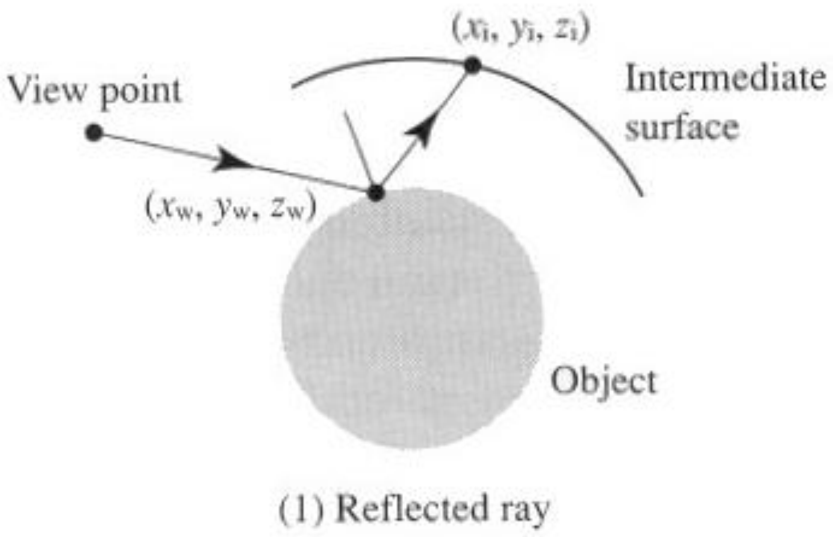


(3) Object centroid

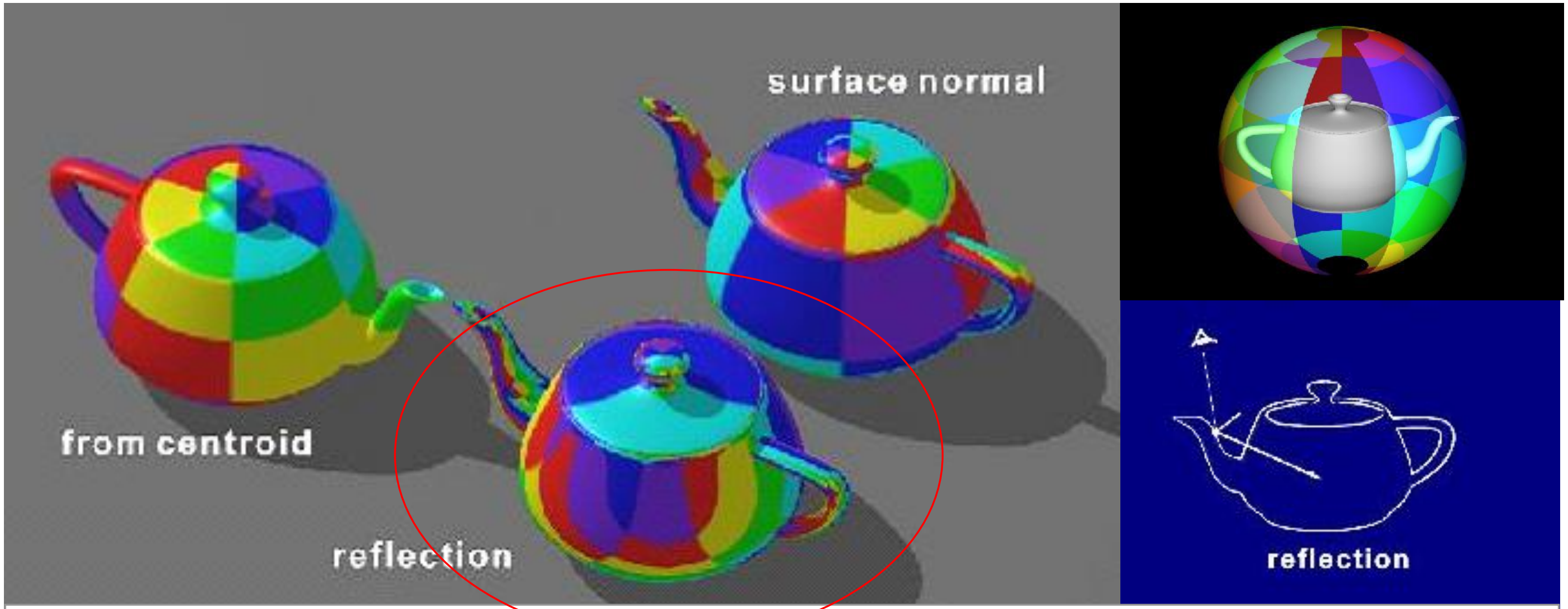


(4) Intermediate surface normal

5.2.1 O mapping 1



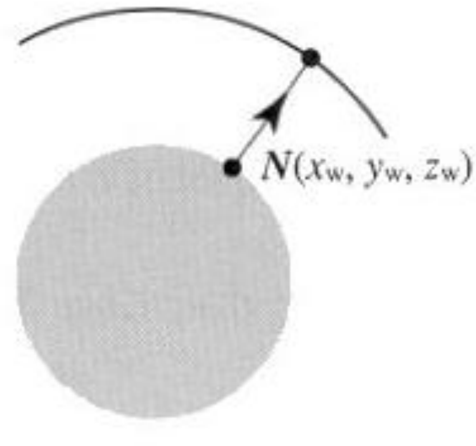
1. The intersection of the reflected ray with the intermediate surface, T’
(See environment mapping)



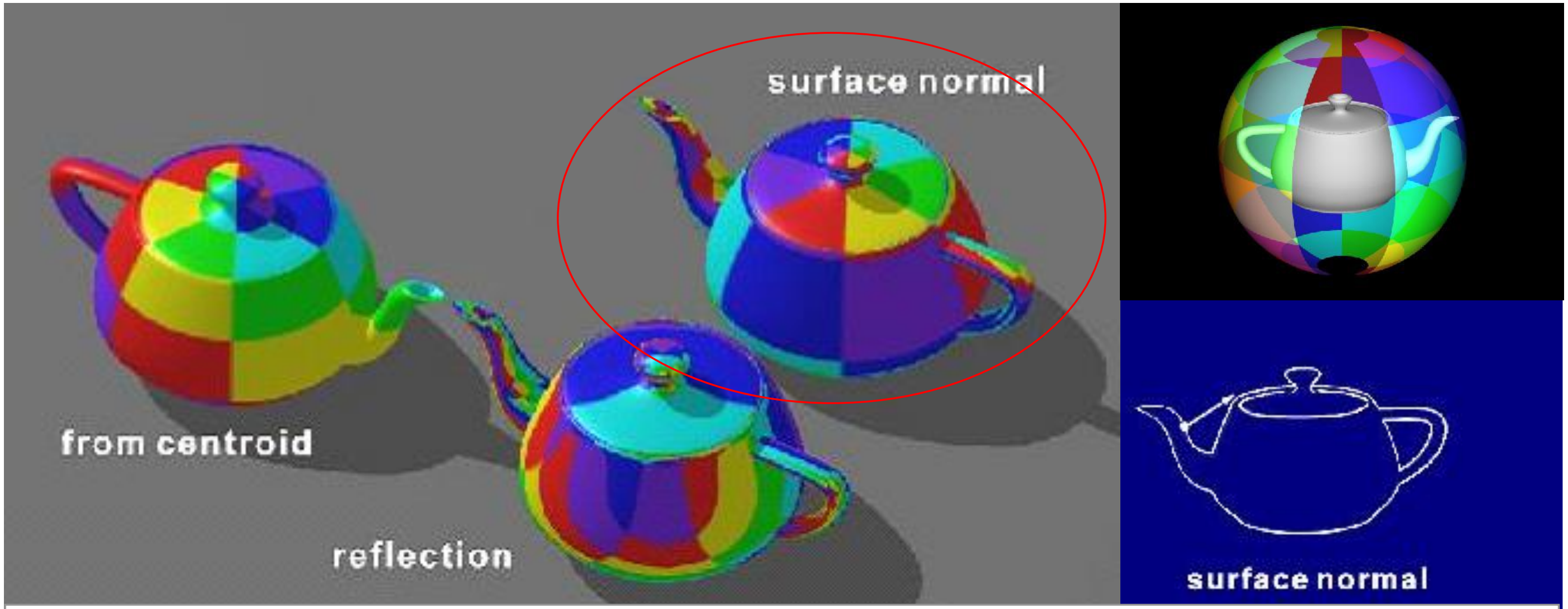
Results for a spherical map shape (The Siggraph 97 Education Slide set)

5.2.2 O mapping 2

2. The intersection of the surface normal at (x_w,y_w,z_w) with T'



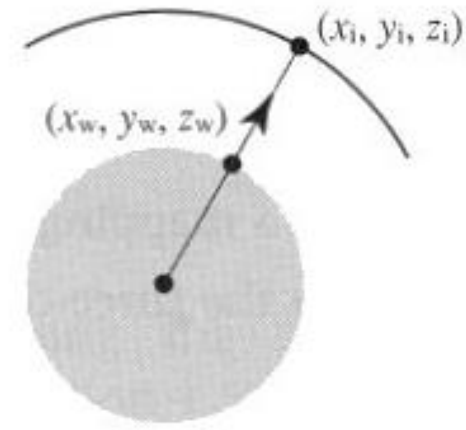
(2) Object normal



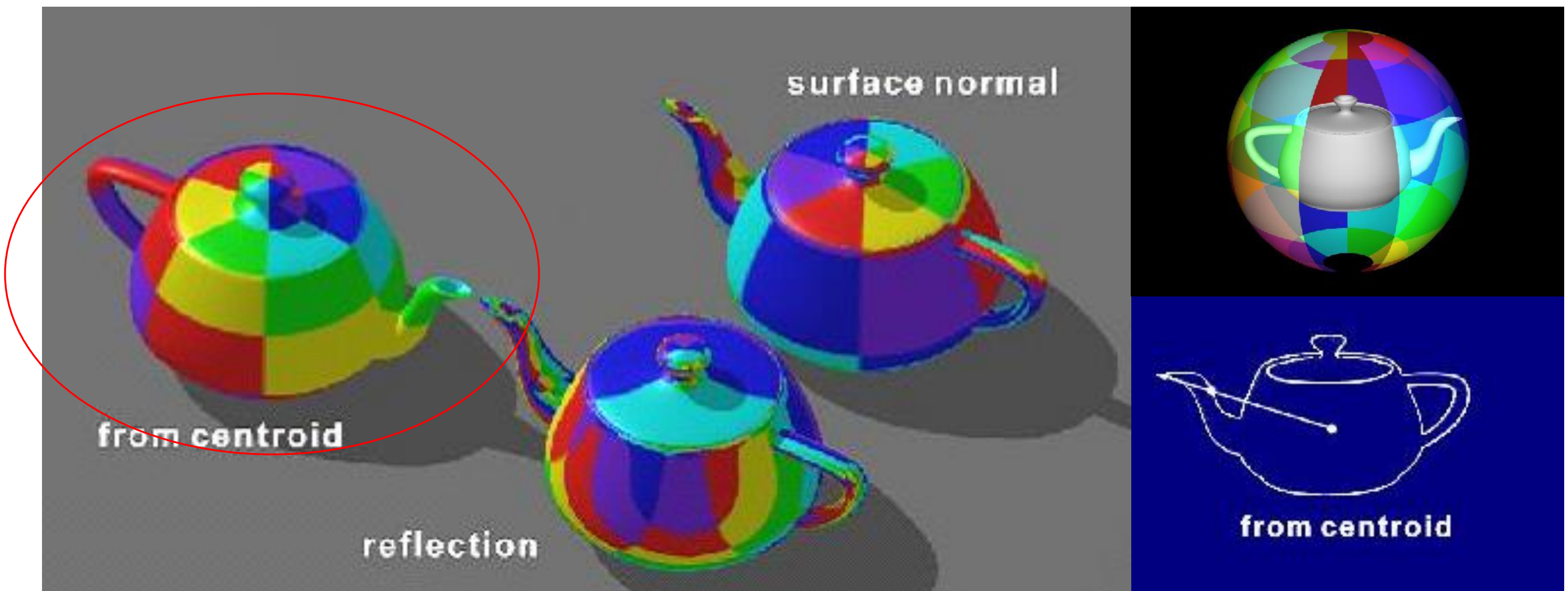
Results for a spherical map shape (The Siggraph 97 Education Slide set)

5.2.3 O mapping 3

3. The intersection of a line through (x_w, y_w, z_w) and the object centroid (point or axis) with T'



(3) Object centroid

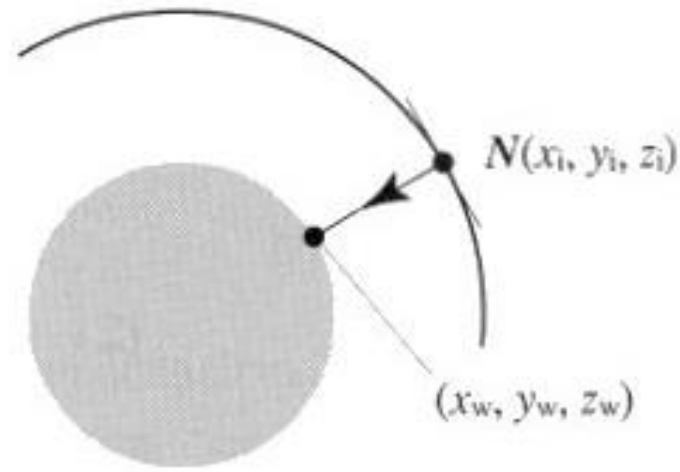


Results for a spherical map shape (The Siggraph 97 Education Slide set)

5.2.4 O mapping 4

4. The intersection of the line from (x_w,y_w,z_w) to T' whose orientation is given by the surface normal at (x_i,y_i,z_i)

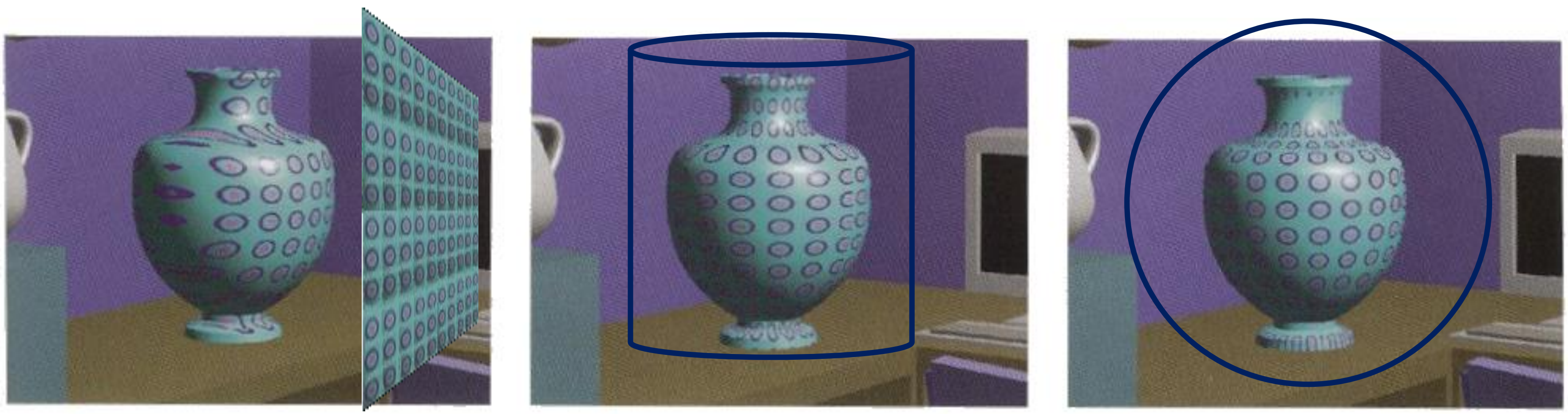
- If the intermediate surface is a plane, this is equivalent to projecting a slide in a slide projector



(4) Intermediate surface normal

5.2.5 Using different intermediate surfaces

- **Plane:** Strong distortion where object surface normal ⊥ plane normal
- **Cylinder :** Reasonably uniform mapping (symmetry)
- **Sphere:** Problems with concave regions

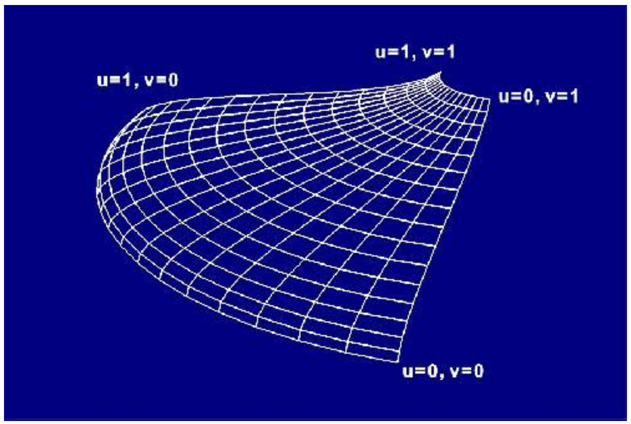


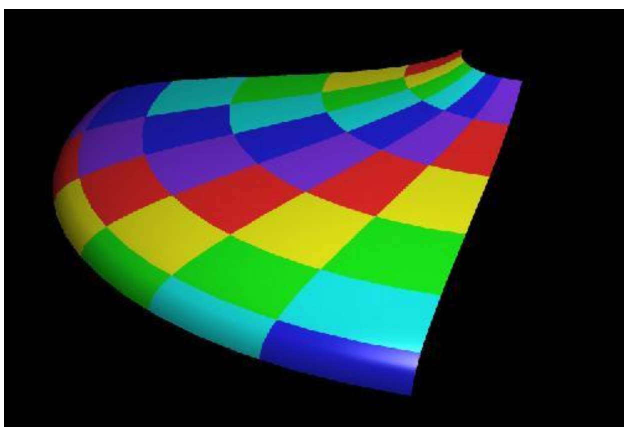
(a) (b) (c)

Figure 8.7
Examples of two-part texture mapping with a solid of revolution. The intermediate surfaces are: (a) a plane (or no surface); (b) a cylinder; and (c) a sphere.

6. Parametric patch objects

- Parameterisation is trivial for patches since they already have (u,v) parameters





The SIGGRAPH 97 Education Slide set

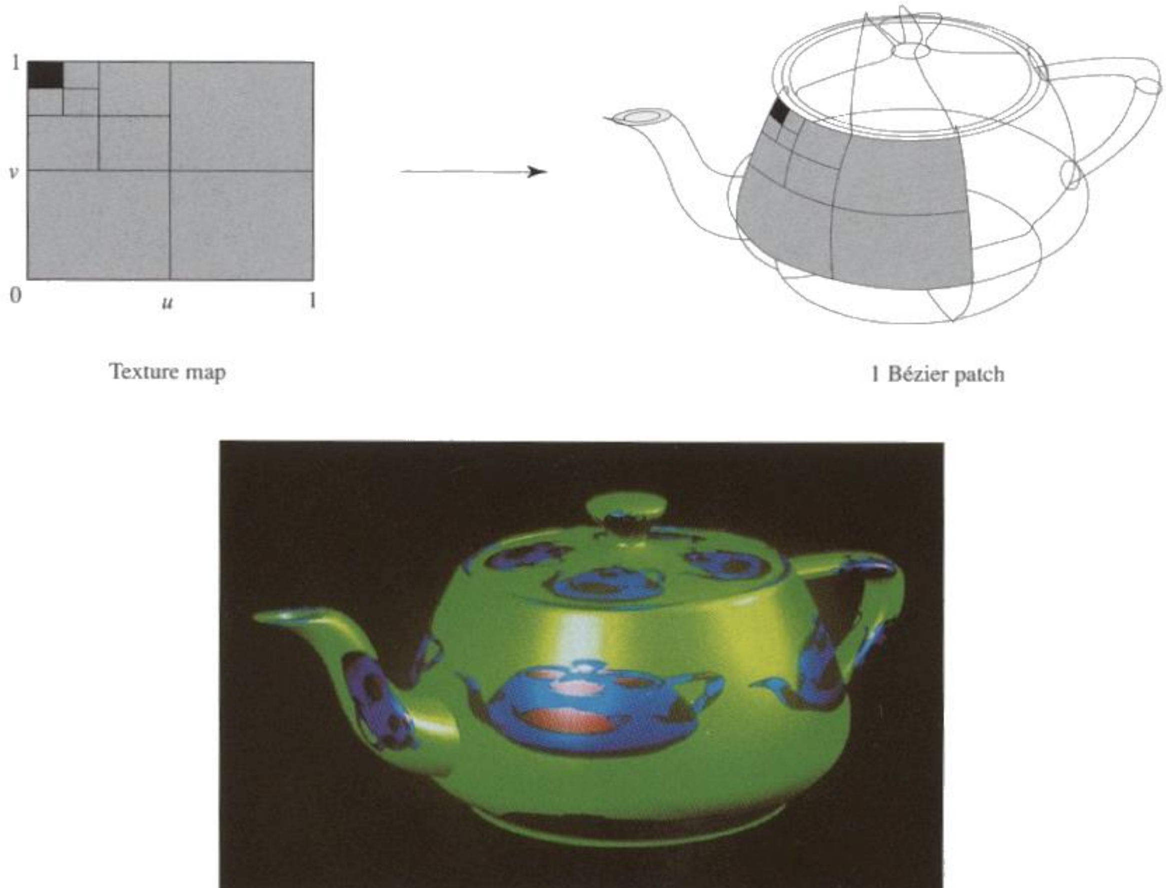
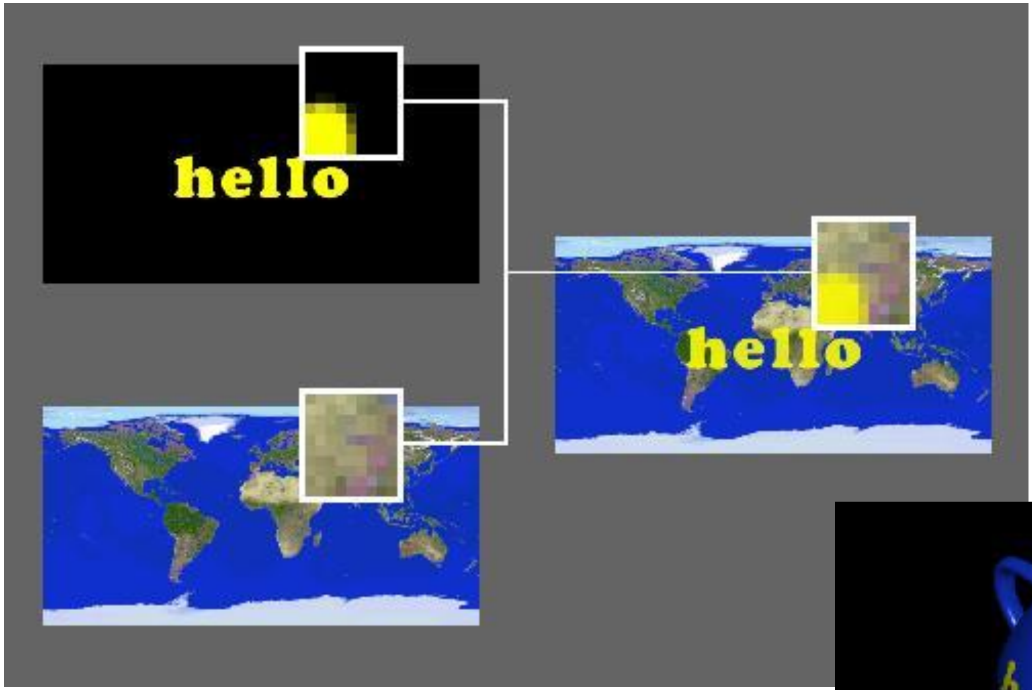
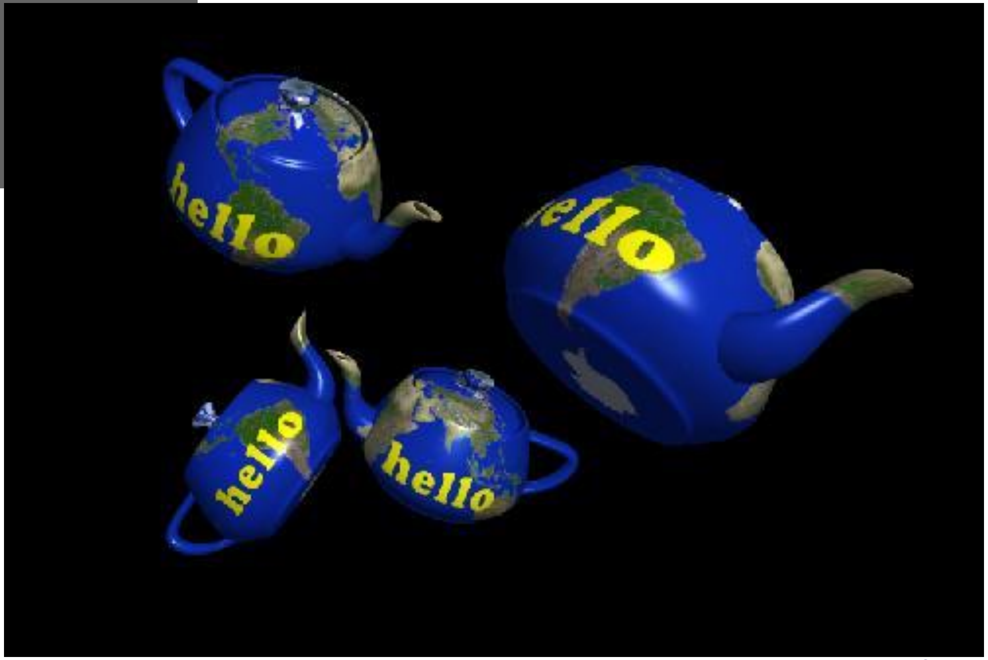


Figure 8.8 (Left) Texture map. (Right) One Bézier patch on the object. (Below) Recursive teapot. (Courtesy of Steve Maddock.)

6.1 Mixing texture maps

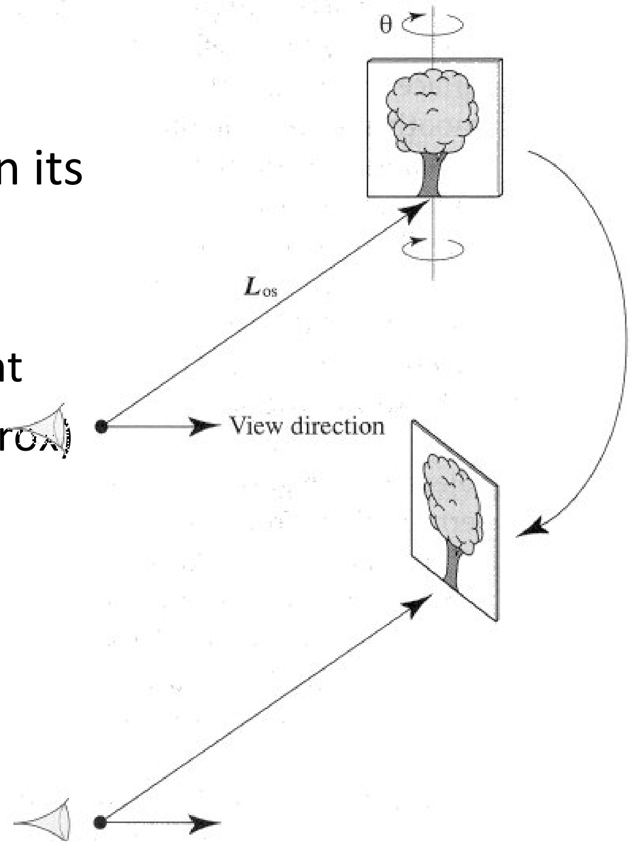


The SIGGRAPH 97 Education Slide set

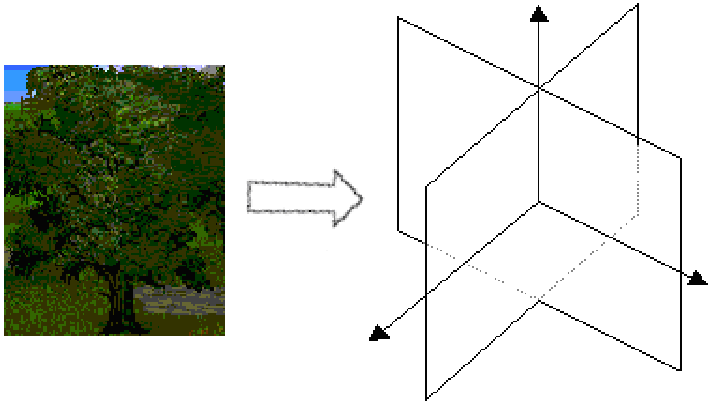


7. Billboards

- The texture map is considered as a 3D entity in its own right
  - Example: single billboard
  - It is rotated so that it is normal to the viewpoint
  - Only works well if the viewing direction is (approx) parallel to the ground plane
- More advanced techniques use multiple billboards

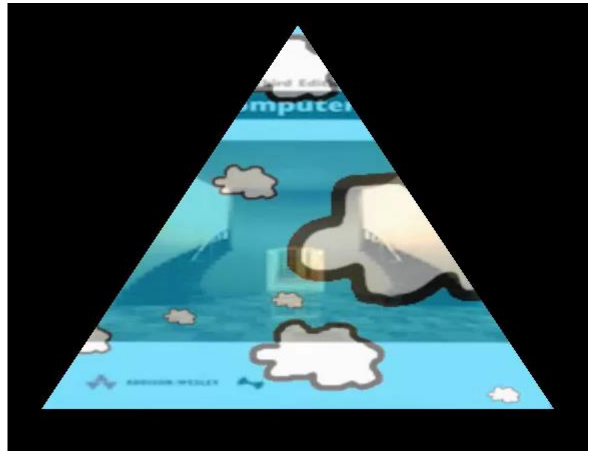


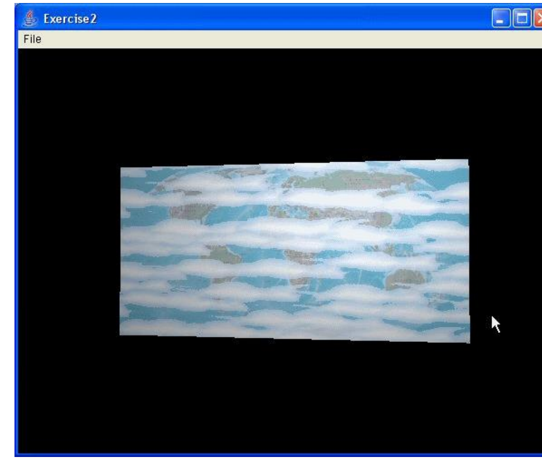
Actua Golf

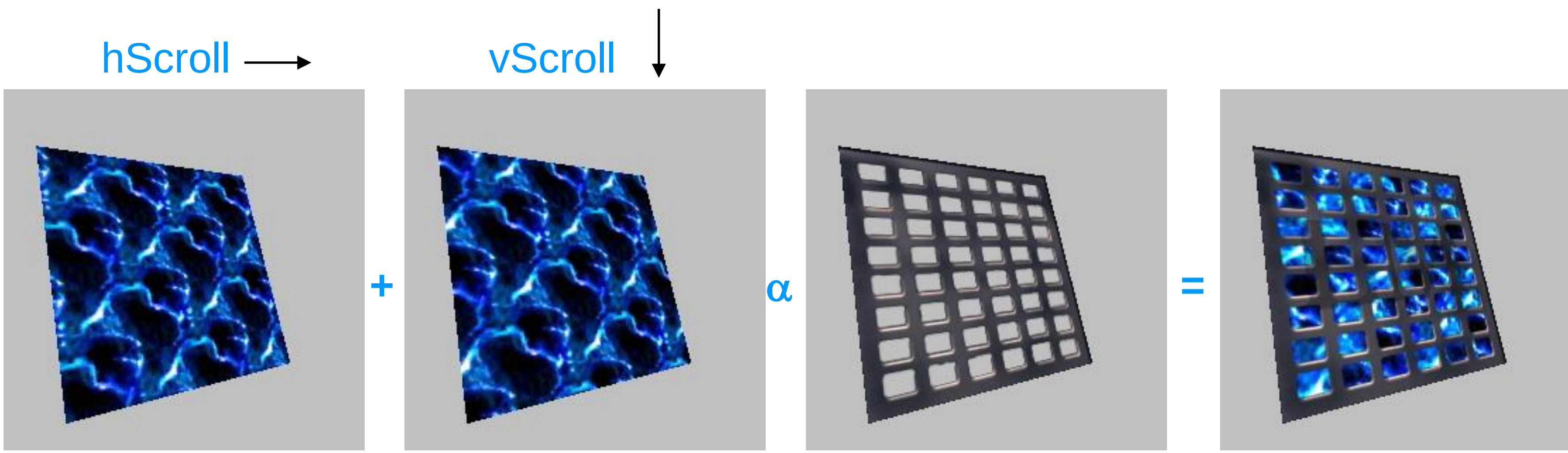


8. Animating textures

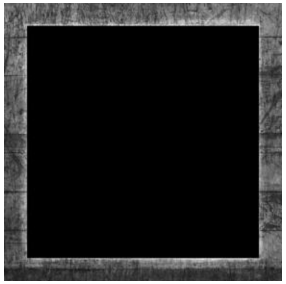
- We can use layers of textures and animate individual layers



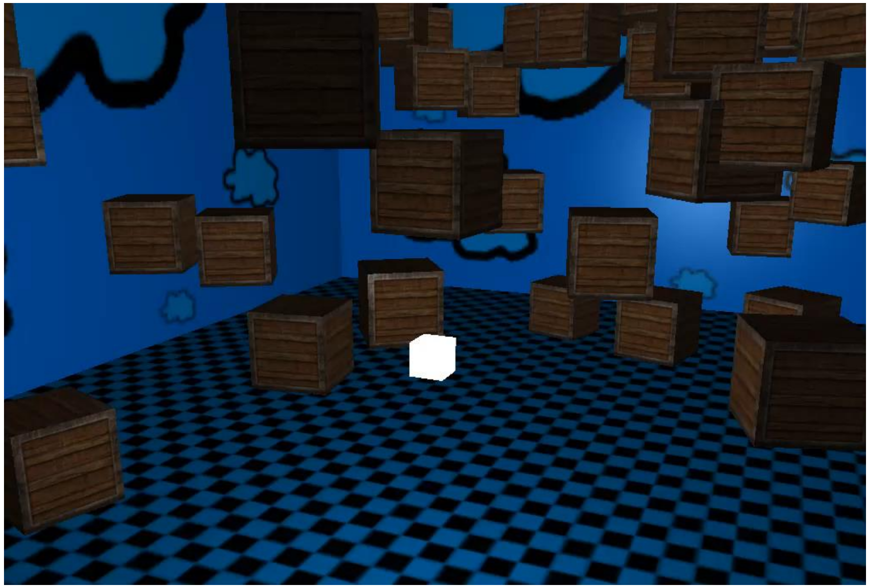


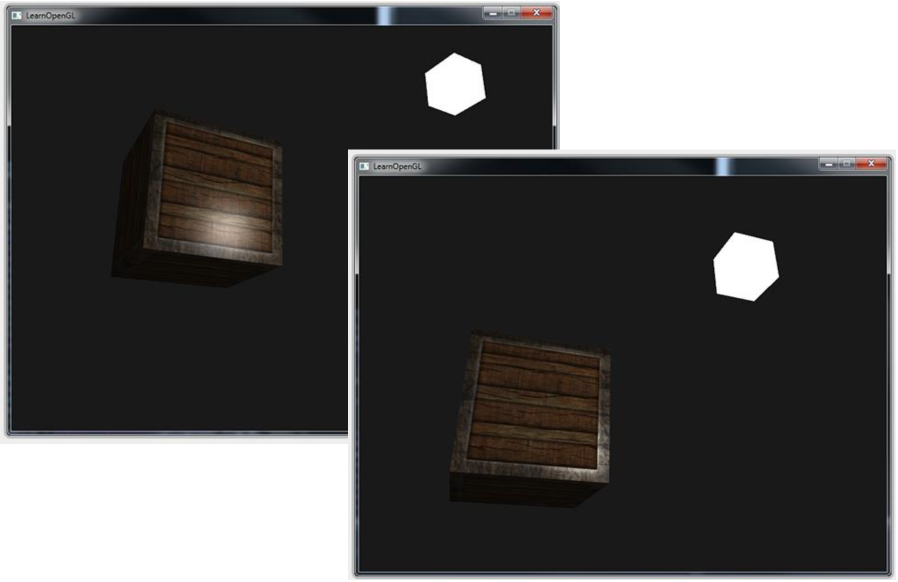


9. Diffuse and specular lighting maps



- The diffuse texture map value multiplies the diffuse (and ambient) parts of the Phong equation calculation
- The specular texture map value multiplies the specular part of the Phong equation calculation





https://learnopengl.com/#!Lighting/Lighting-maps

10. Summary

- Texture mapping applies a pattern of colour to an object
  - Texture is not ‘texture’!!
  - Texture is an add-on like shadows
- Most common approach is to apply 2D texture maps to polygon mesh objects using bilinear interpolation and inverse mapping
  - Common in hardware
  - u,v labelling of texture; u,v labelling of object vertices; interpolation over triangle; u,v look-up; pixel colour modulation
- Another useful approach is *two-stage mapping*
  - Mapping using an intermediate surface:
  - Overcomes the surface parameterisation problem in complex polygon mesh objects
- Diffuse and specular lighting maps can be used to create a range of lighting effects

Appendix: O mapping number 3: Shrinkwrap

1. Inverse map four pixel points to four points (x_w, y_w, z_w) on the surface of the object
2. Apply the O mapping to find the point (θ, h) on the surface of the cylinder.

$(x_w, y_w, z_w) \rightarrow (\theta, h) = (\tan^{-1}(y_i/x_i), z_i)$

3. Apply the S mapping to find the point (u,v) corresponding to (θ, h).