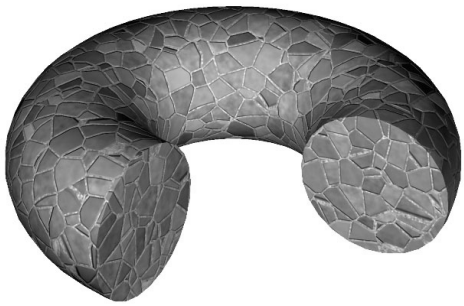
Texture Mapping



- Motivation: Add interesting and/or realistic detail to surfaces of objects.
- Problem: Fine geometric detail is difficult to model and expensive to render.
- Idea: Modify various shading parameters of the surface by mapping a function (such as a 2D image) onto the surface.

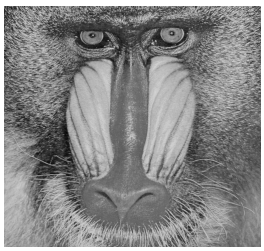
Texture Mapping Example

- Given an image, think of it as a 2D function from $[0,1]^2$ (texture coordinates) to the RGB color space: $T(u,v) \rightarrow (r,g,b)$
- For each geometric primitive, define a mapping M that maps points on the surface to texture coordinates: $M(x,y,z) = (u,v)$
- To shade a pixel corresponding to a point (x,y,z) on the surface, use the color:

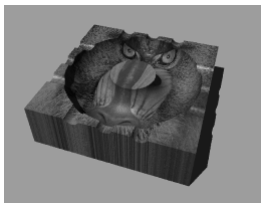
$$(r,g,b) = T(M(x,y,z))$$

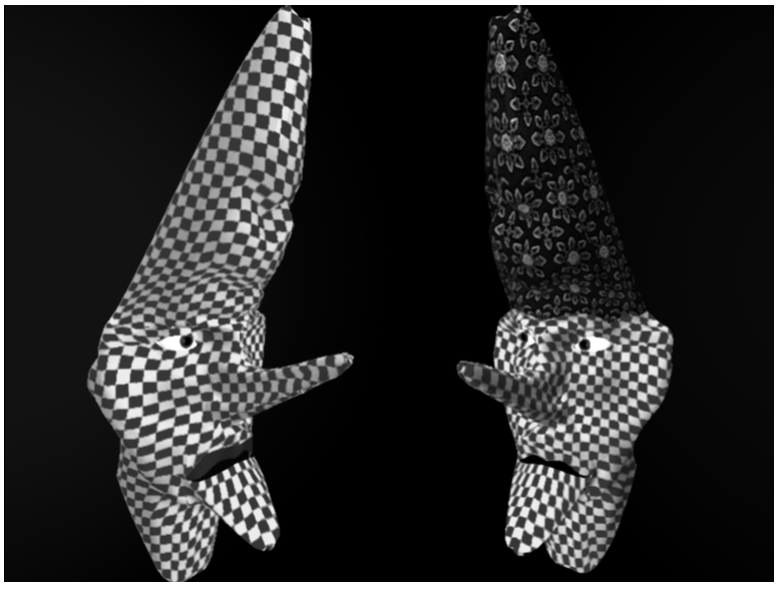
Texture Mapping Example

- Texture:



- Result:

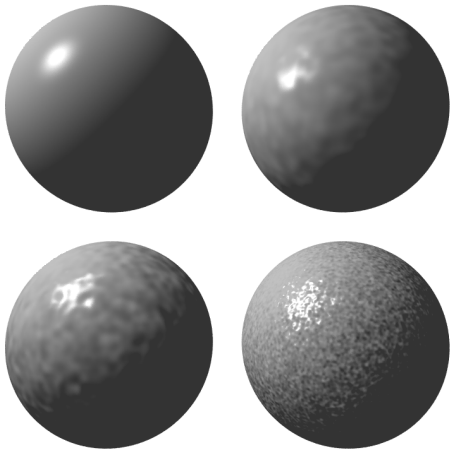




Affected Parameters

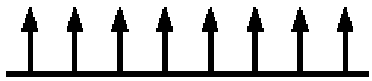
- Final color
- Reflectance (either diffuse or specular)
- Surface normal (bump mapping)
- Transparency
- Reflected color (environment mapping)
- Any combination of the above

Bump Mapping



Bump Mapping (Blinn 78)

Smooth surface:



Bumpy surface:

Bump-mapped surface:

Parametrizing Objects

- Certain objects have a natural parametrization (e.g., Bezier patches)
- Polygons (triangles): each vertex is assigned a pair of texture coordinates (u,v). Inside, linear interpolation is used.
- How do we handle a more complex object?

Two-Step Texture Mapping

(Bier and Sloan 1986)

- Step I: define a mapping between the texture and some intermediate surface:
  - plane
  - cylinder
  - sphere
  - cube
- Step II: Project intermediate surface onto object surface

Intermediate Surface Projections

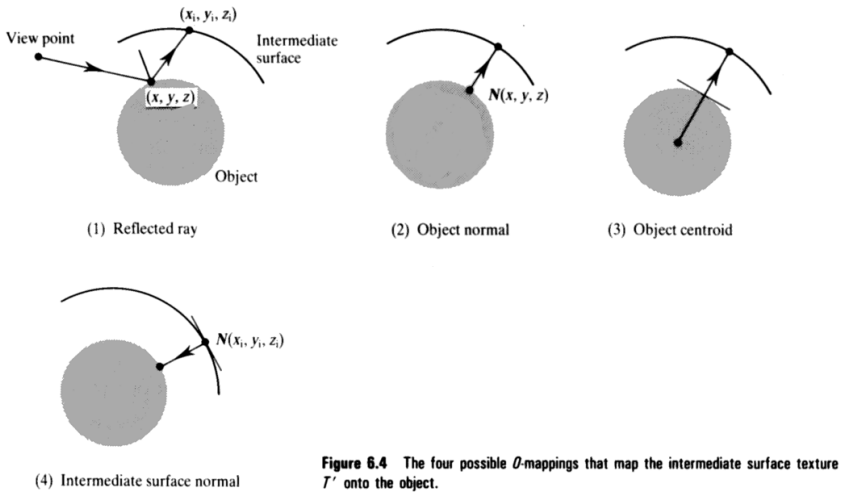
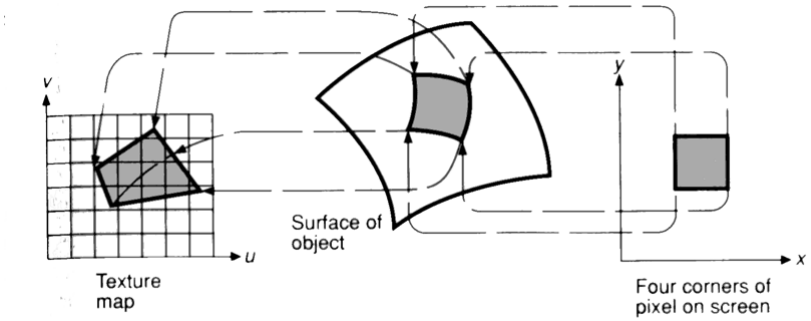


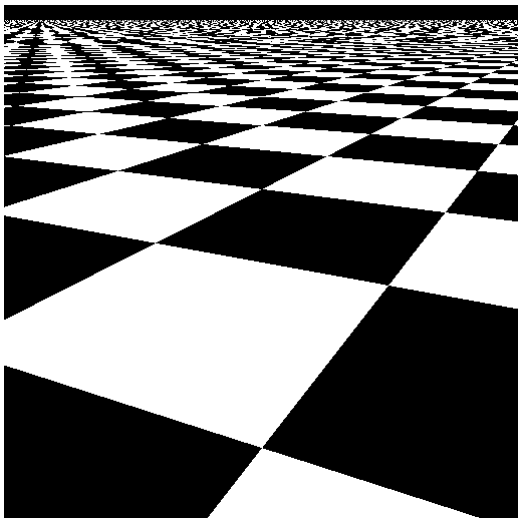
Figure 6.4 The four possible O-mappings that map the intermediate surface texture T' onto the object.

Texture Anti-Aliasing

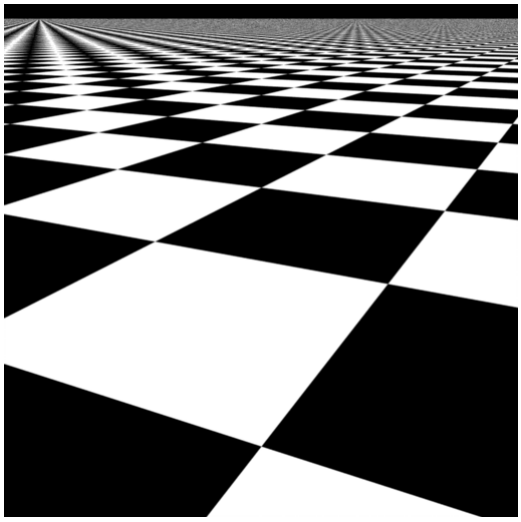
- A single screen space pixel might correspond to many texels:



Unfiltered Texture:



Filtered Texture:



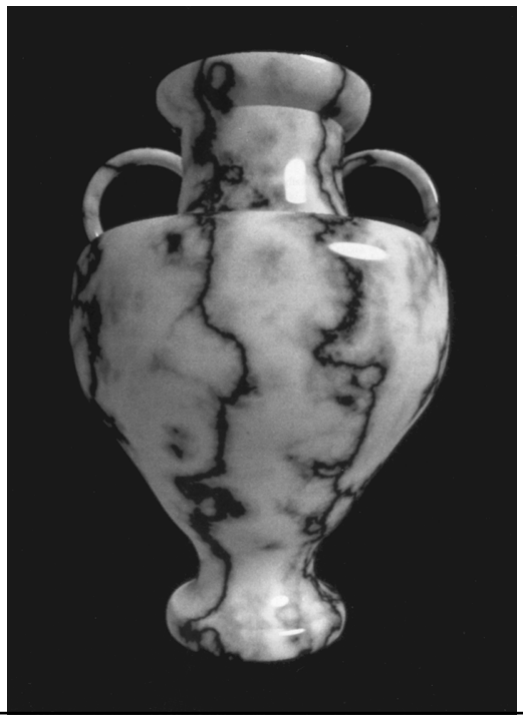
Texture Pre-Filtering

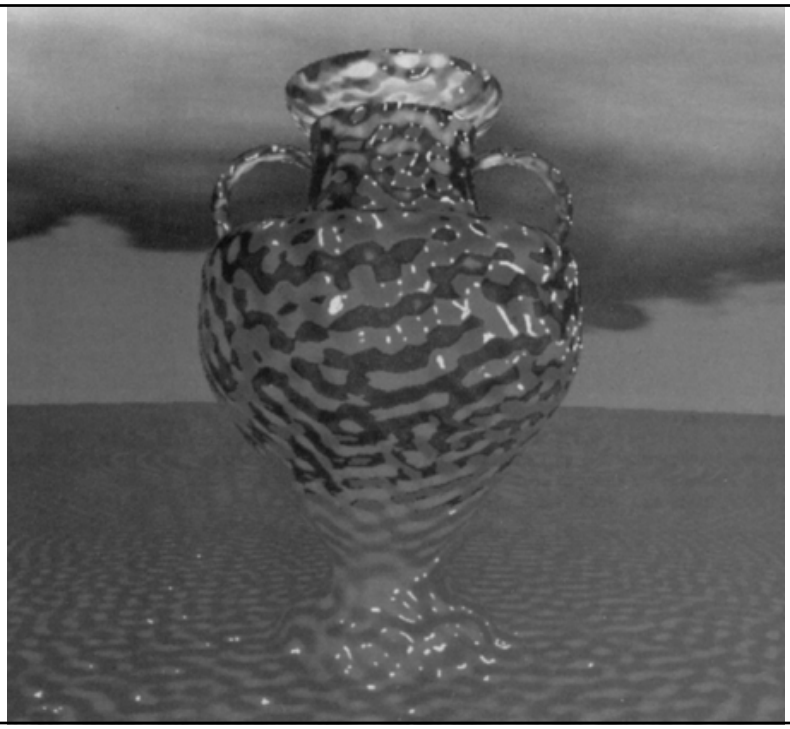
- Problem: filtering the texture during rendering is too slow for interactive performance.
- Solution: pre-filter the texture in advance
  - Summed area tables - gives the average value of each axis-aligned rectangle in texture space
  - Mip-maps (tri-linear interpolation) - supported by most of today's texture mapping hardware

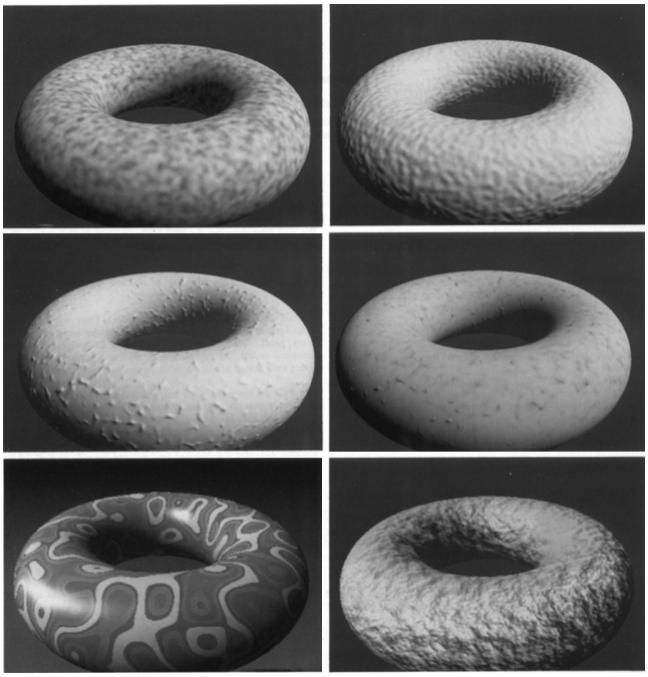
Solid Textures

(Peachey 1985, Perlin 1985)

- Problem: mapping a 2D image/function onto the surface of a general 3D object is a difficult problem:
  - Distortion
  - Discontinuities
- Idea: use a texture function defined over a 3D domain - the 3D space containing the object
  - Texture function can be digitized or procedural







Procedural Textures

- Advantages:
  - compact representation (code vs. data)
  - unlimited resolution
  - unlimited extent
  - controllable via parameters
- Disadvantages:
  - Can be difficult to program and debug
  - Can be difficult to predict and control
  - Typically slower to evaluate
  - Can be difficult to pre-filter

Reaction-Diffusion Textures Turk 91; Witkin & Kass 91

- Reaction-diffusion is a mathematical model for generation of natural patterns, arising due to local non-linear interactions of excitation and inhibition.
- First proposed by Alan Turing in 1952