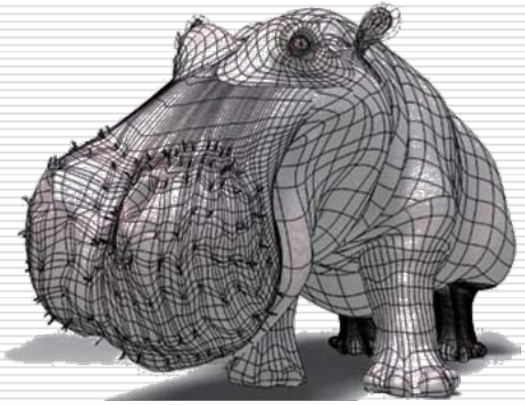
Computer Graphics

Bing-Yu Chen
National Taiwan University

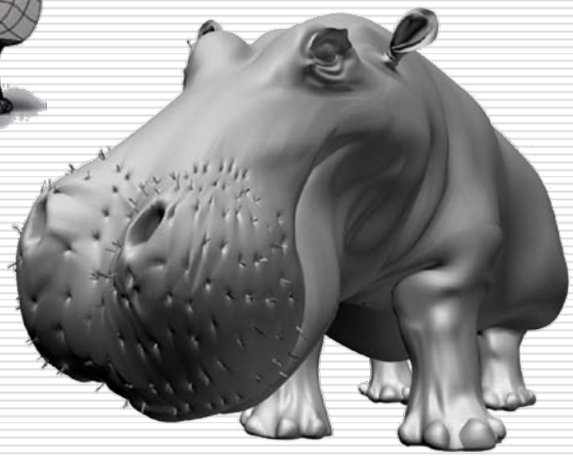
Texture Mapping

- Texture Mapping
- Texture Aliasing
- MIPmaps
- Environment Mapping
- Bump Mapping
- Displacement Mapping
- Shadow Maps
- Solid Textures

The Quest for Visual Realism



Model



Model with Shading

Model with Shading and Textures

The Limits of Geometric Modeling

- Although graphics cards can render over 10 million polygons per second, that number is insufficient for many phenomena
  - Clouds
  - Grass
  - Terrain
  - Skin

Texture Mapping

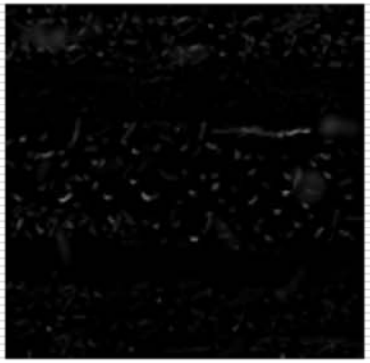
- Previously, we assume that reflection properties such as are constant within each triangle.
- However, some objects have complex appearance which arises from variation in reflection properties.
- The common technique to handle this kind of variation is to store it as a function or a pixel-based image and “map” it onto a surface.
- The function is called ***texture map*** and the process is called ***texture mapping***.

Texture Maps

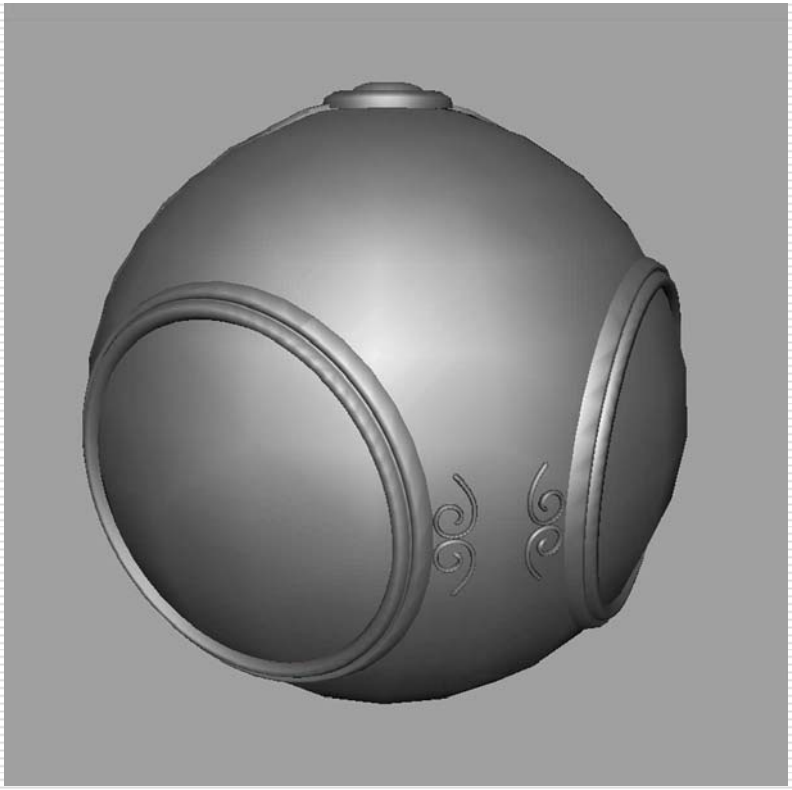
- How is texture mapped to the surface?
  - Dimensionality: 1D, 2D (image), 3D (solid)
  - Procedural v.s. table look-up
  - Texture coordinates (s,t)
    - Surface parameters (u,v)
    - Projection: spherical, cylindrical, planar
    - Reparameterization
- What does texture control?
  - Surface color and transparency
  - Illumination: environment maps, shadow maps
  - Reflection function: reflectance maps
  - Geometry: displacement and bump maps

Texture Maps

Tom Porter's Bowling Pin



Texture Mapping



geometric model

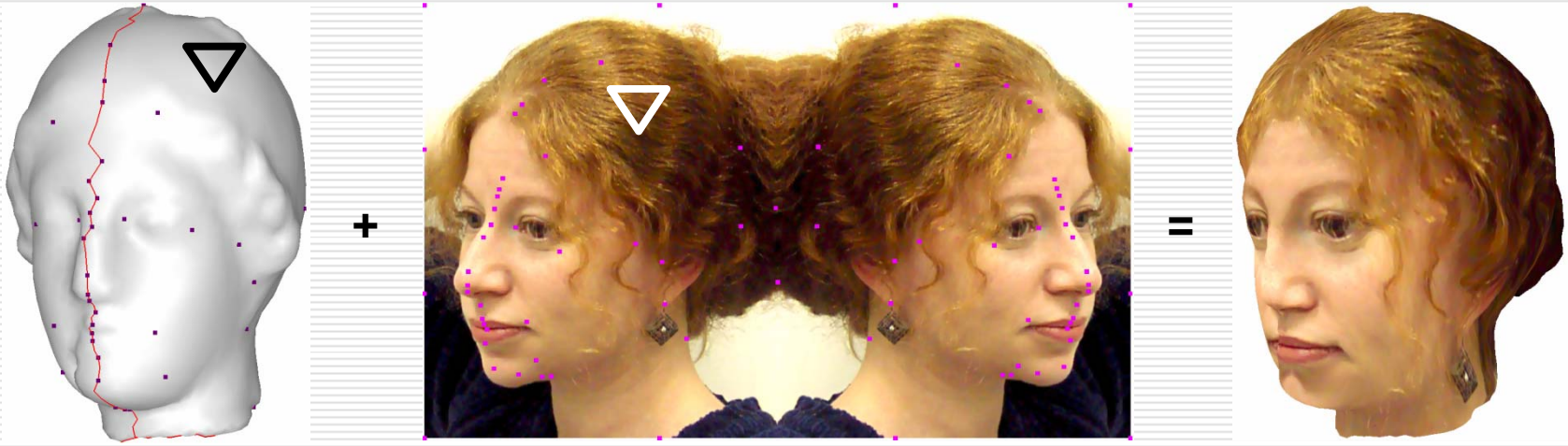
texture mapped

Texture Mapping

2D mapping

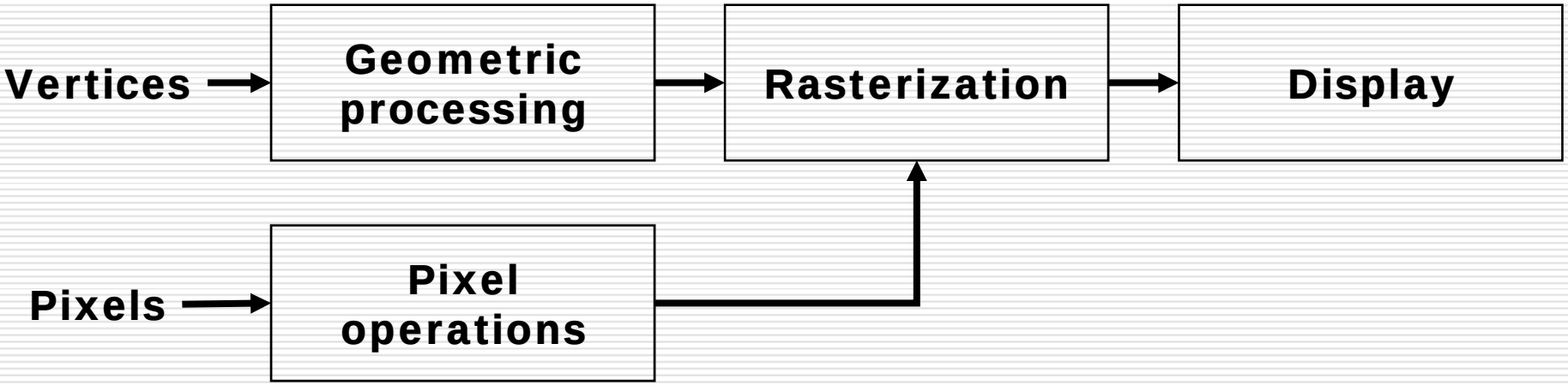
3D mapping

Decal Textures

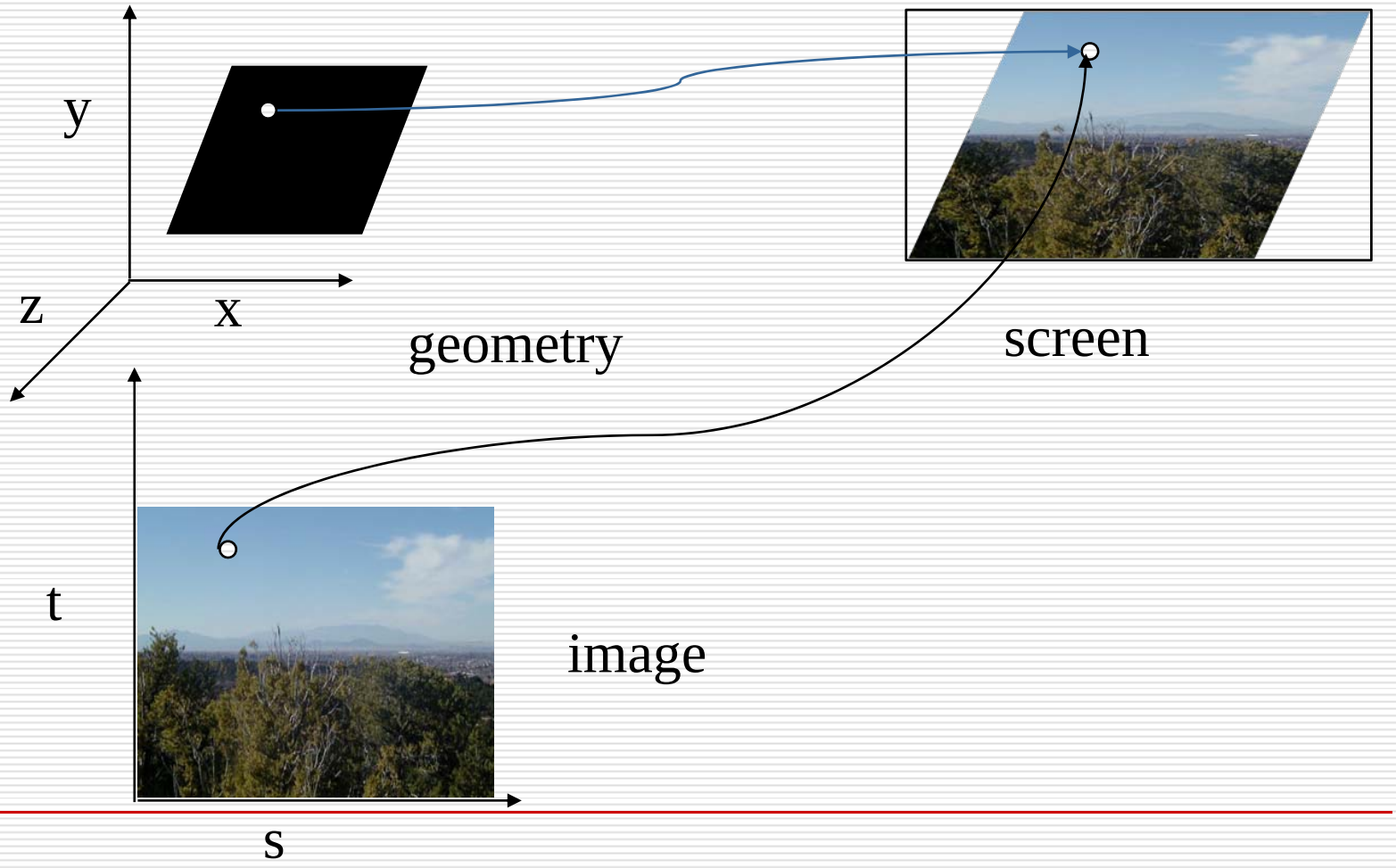


Where does mapping take place?

- Mapping techniques are implemented at the end of the rendering pipeline
  - Very efficient because few polygons pass down the geometric pipeline

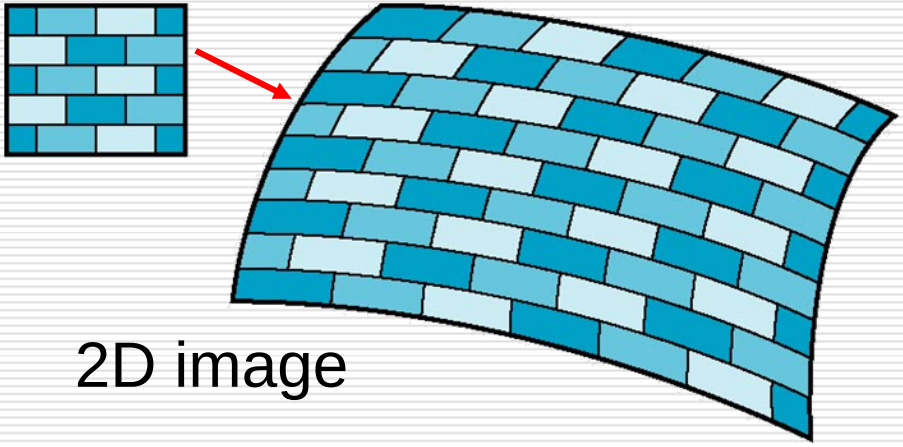


Simple Texture Mapping



Is it simple?

- Although the idea is simple---map an image to a surface---there are 3 or 4 coordinate systems involved



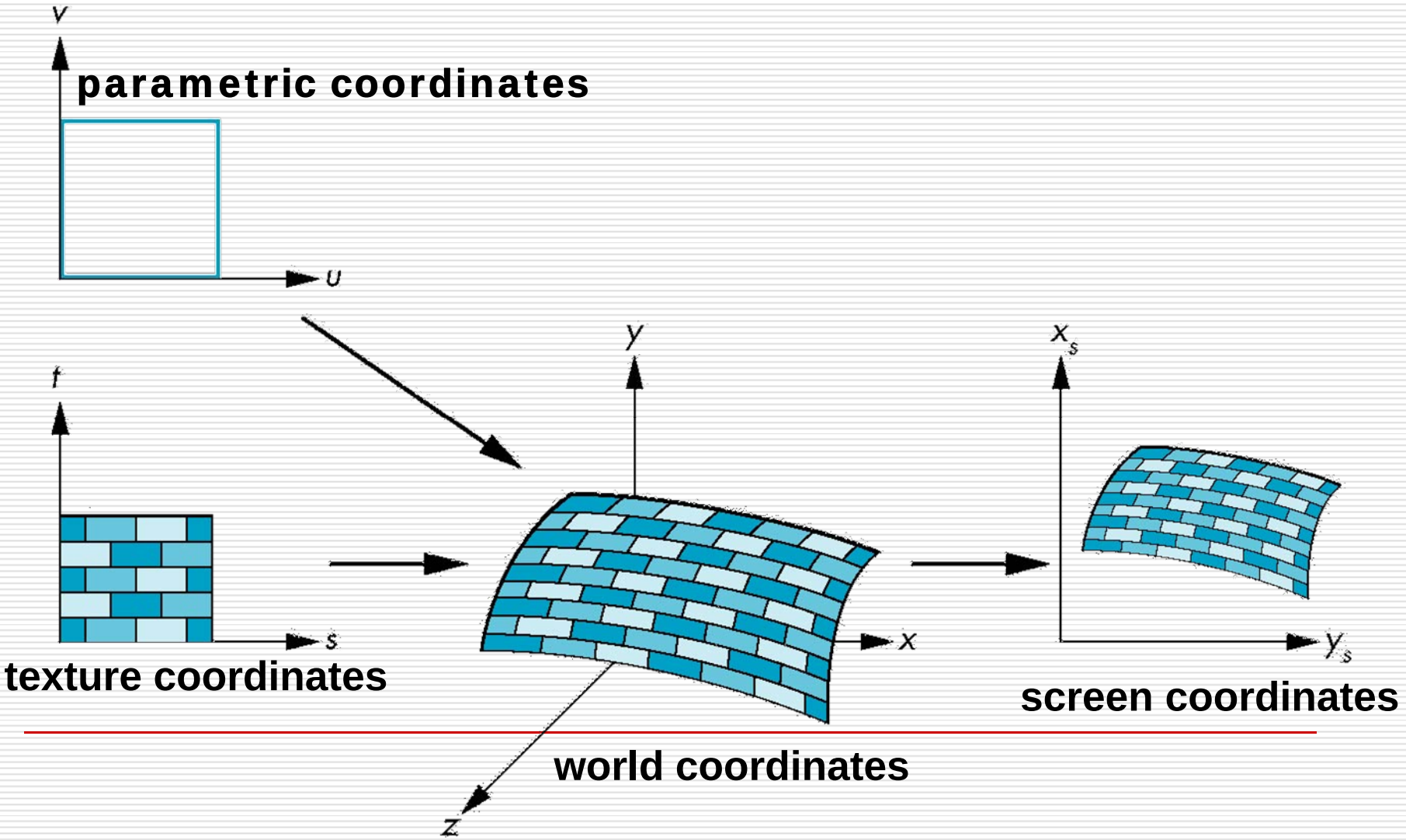
2D image

3D surface

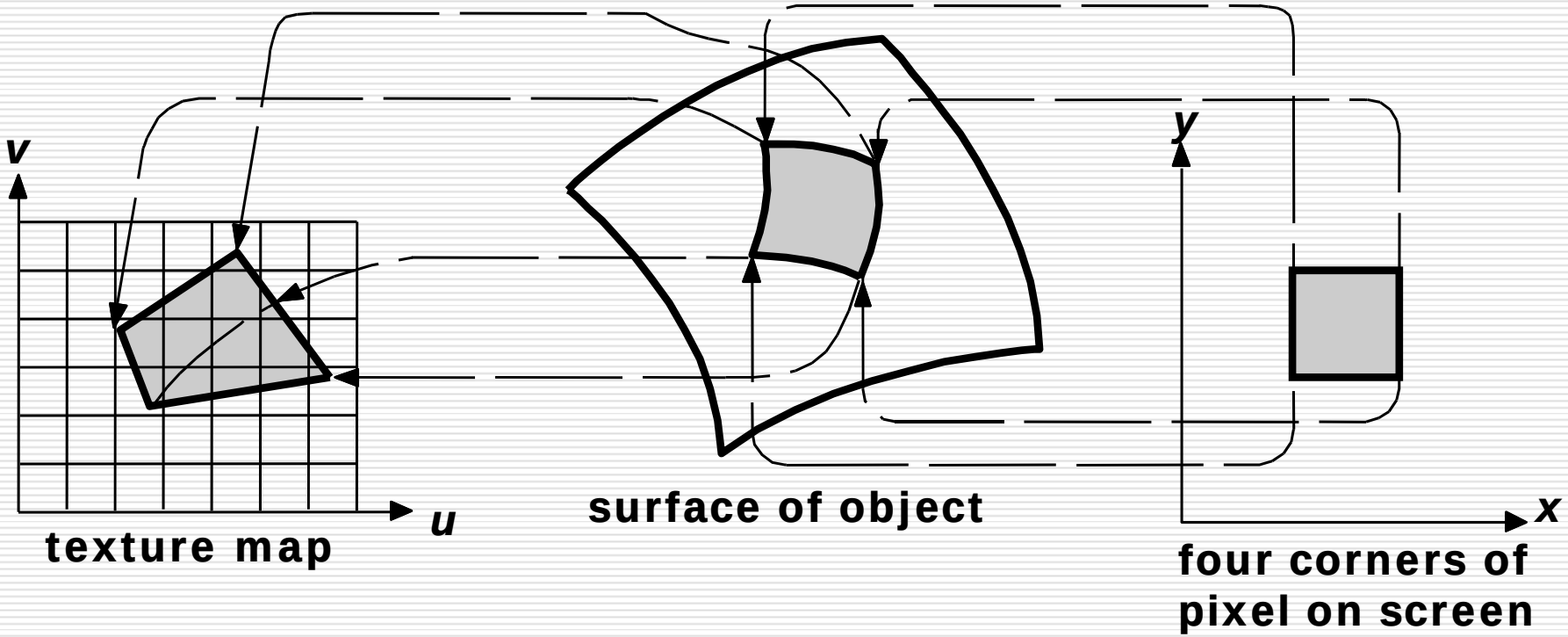
Coordinate Systems

- Parametric Coordinates
  - may be used to model curved surfaces
- Texture Coordinates
  - used to identify points in the image to be mapped
- World Coordinates
  - conceptually, where the mapping takes place
- Screen Coordinates
  - where the final image is really produced

Texture Mapping

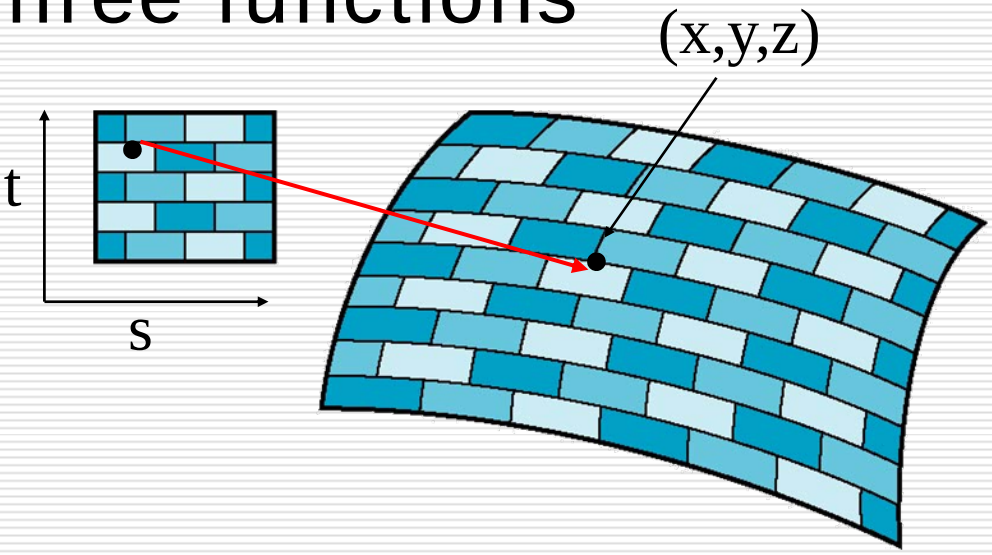


Texture Mapping = Pattern Mapping



Mapping Functions

- Basic problem is how to find the maps
- Consider mapping from texture coordinates to a point a surface
- Appear to need three functions
  - x = x(s,t)
  - y = y(s,t)
  - z = z(s,t)
- But we really want to go the other way



Backward Mapping

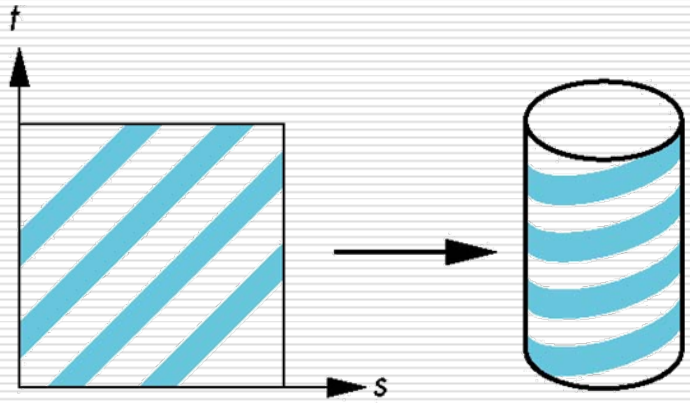
- We really want to go backwards
  - Given a pixel, we want to know to which point on an object it corresponds
  - Given a point on an object, we want to know to which point in the texture it corresponds
    - Need a map of the form

      $s = s(x,y,z)$

      $t = t(x,y,z)$
- Such functions are difficult to find in general

Two-part mapping

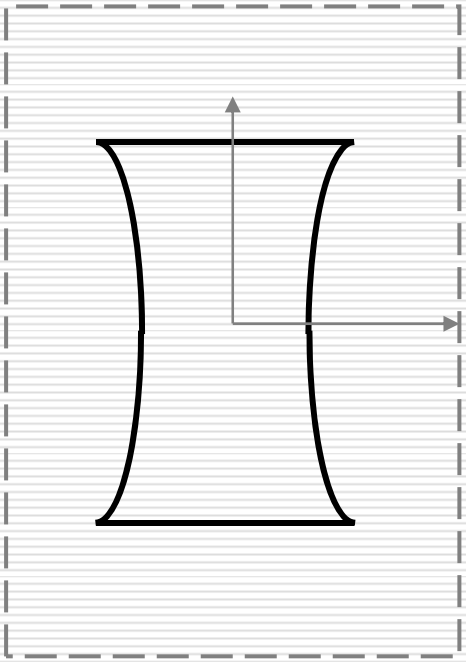
- One solution to the mapping problem is to first map the texture to a simple intermediate surface
- Example: map to cylinder



Cylindrical Mapping

- parametric cylinder
  - $x = r \cos 2\pi u$
  - $y = r \sin 2\pi u$
  - $z = v/h$
- maps rectangle in u, v space to cylinder of radius r and height h in world coordinates
  - $s = u$
  - $t = v$
- maps from texture space

Cylindrical mapping

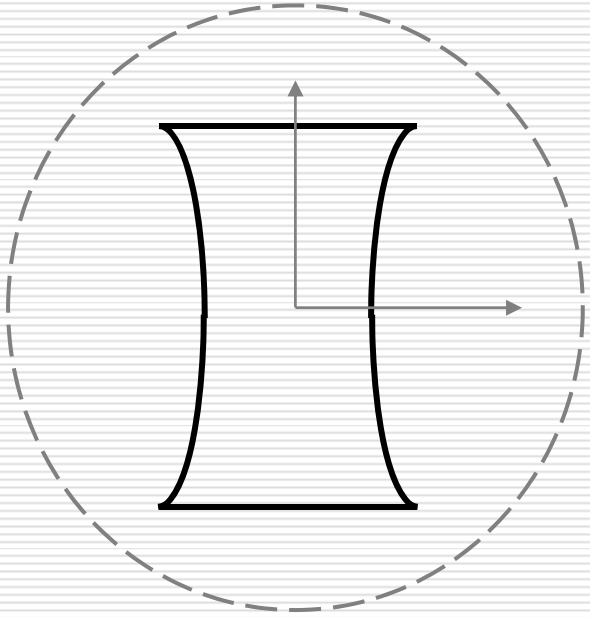


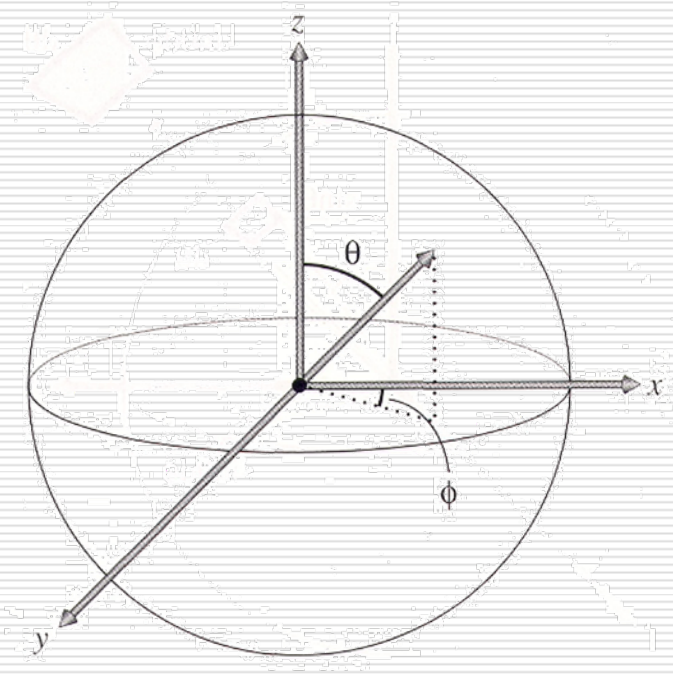
Spherical Map

- We can use a parametric sphere
  - $x = r \cos 2\pi u$
  - $y = r \sin 2\pi u \cos 2\pi v$
  - $z = r \sin 2\pi u \sin 2\pi v$
- in a similar manner to the cylinder but have to decide where to put the distortion

- Spheres are use in environmental maps

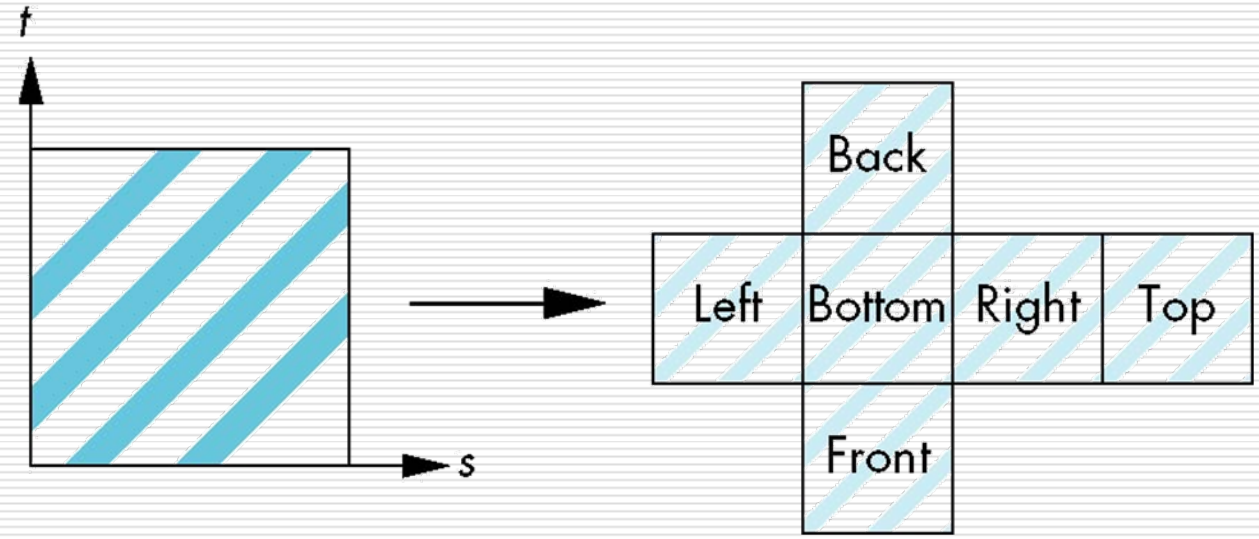
Spherical mapping



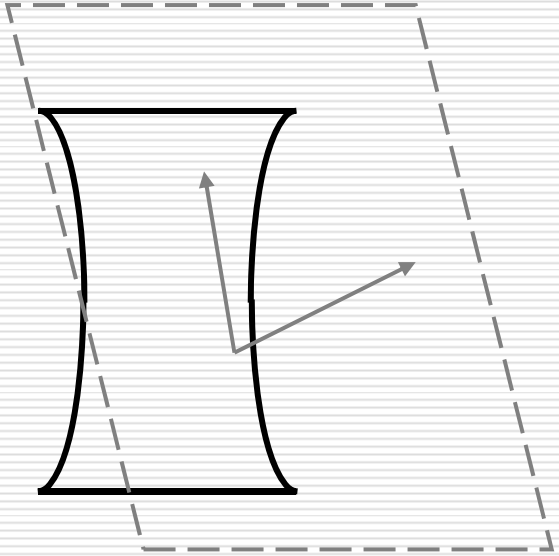


Box Mapping

- Easy to use with simple orthographic projection
- Also used in environmental maps

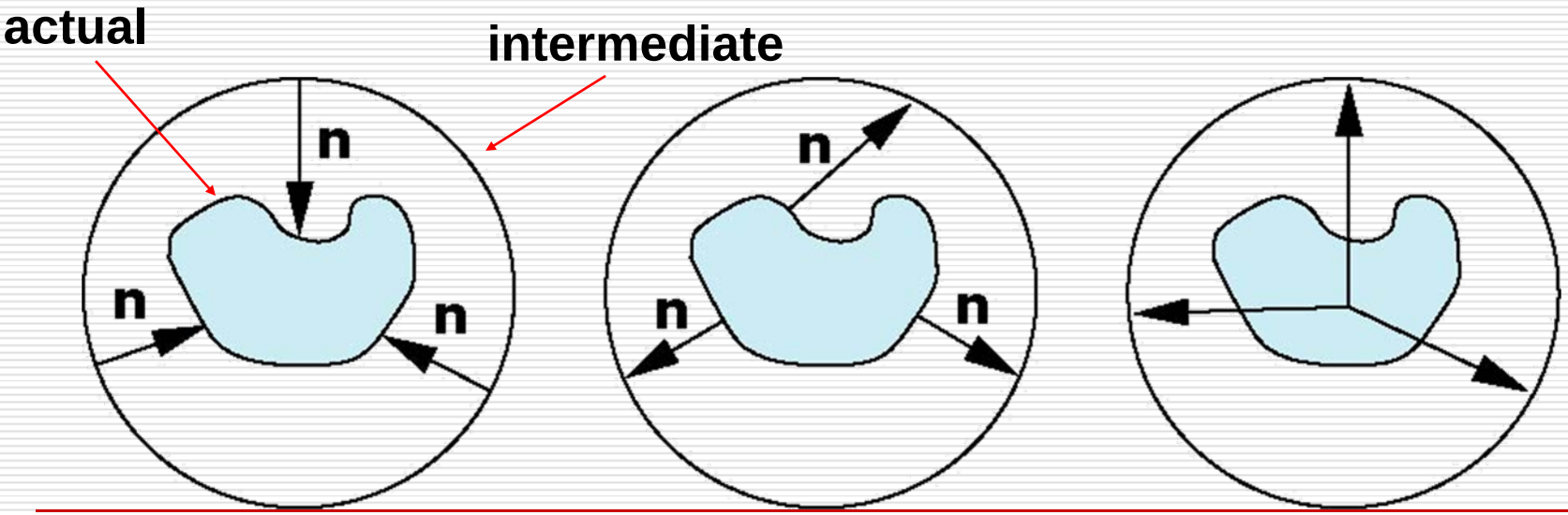


Planar mapping

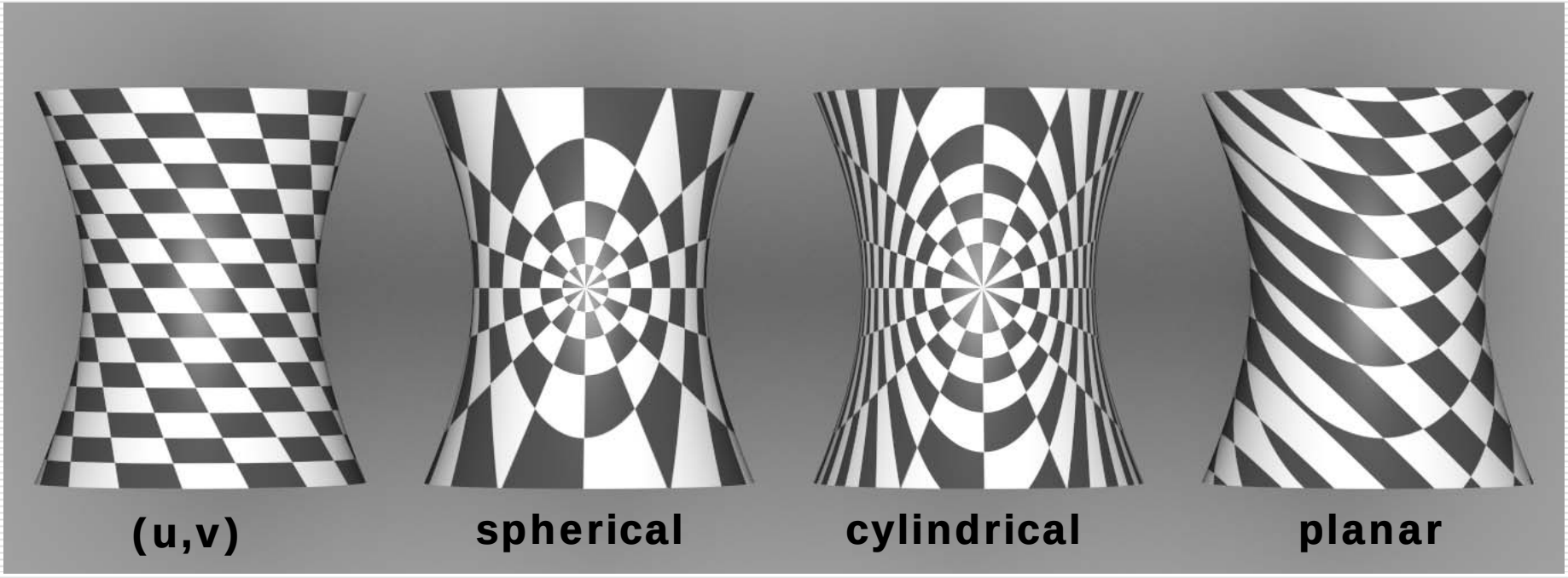


Second Mapping

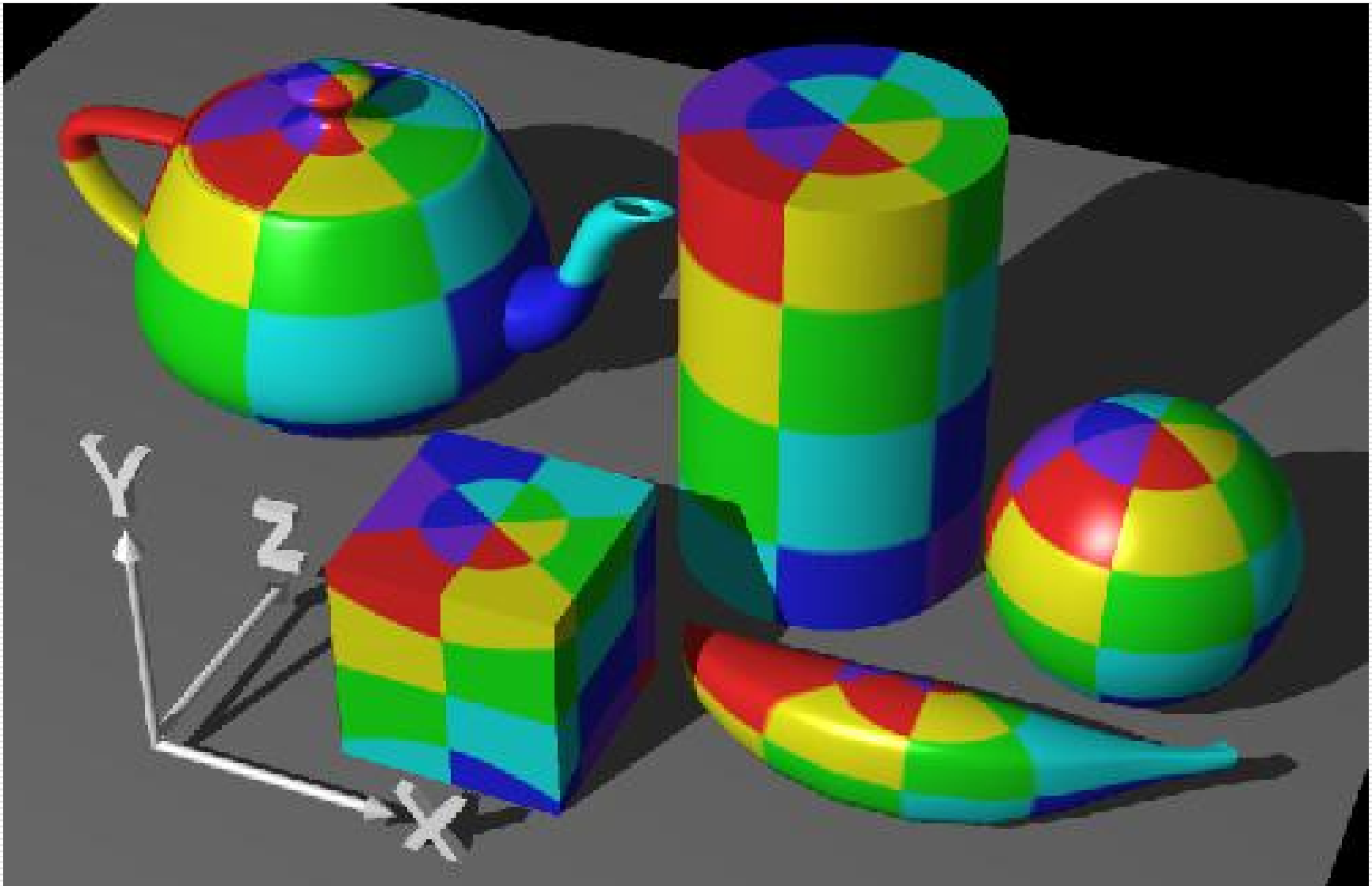
- Map from intermediate object to actual object
  - Normals from intermediate to actual
  - Normals from actual to intermediate
  - Vectors from center of intermediate



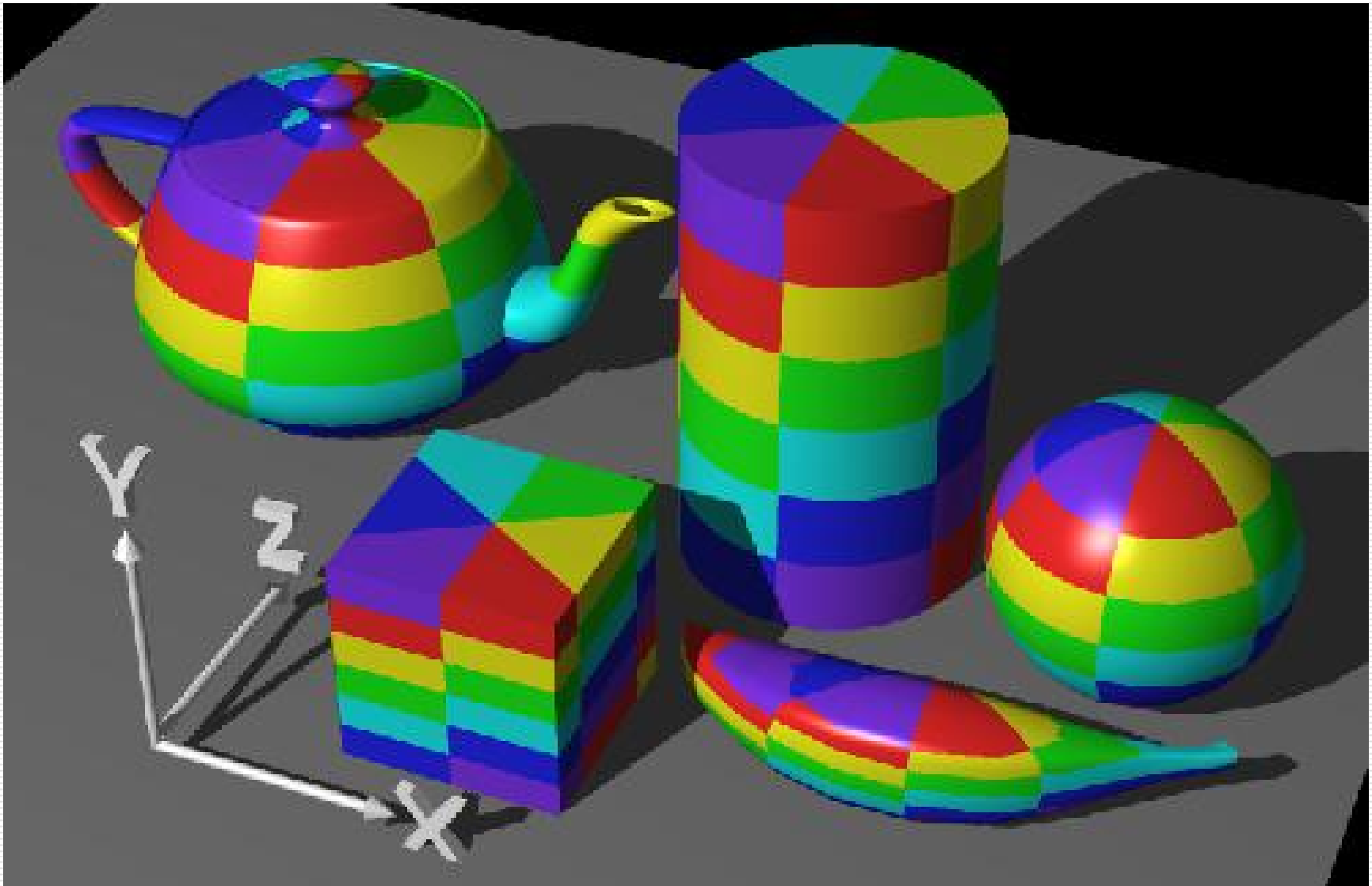
Texture Mapping



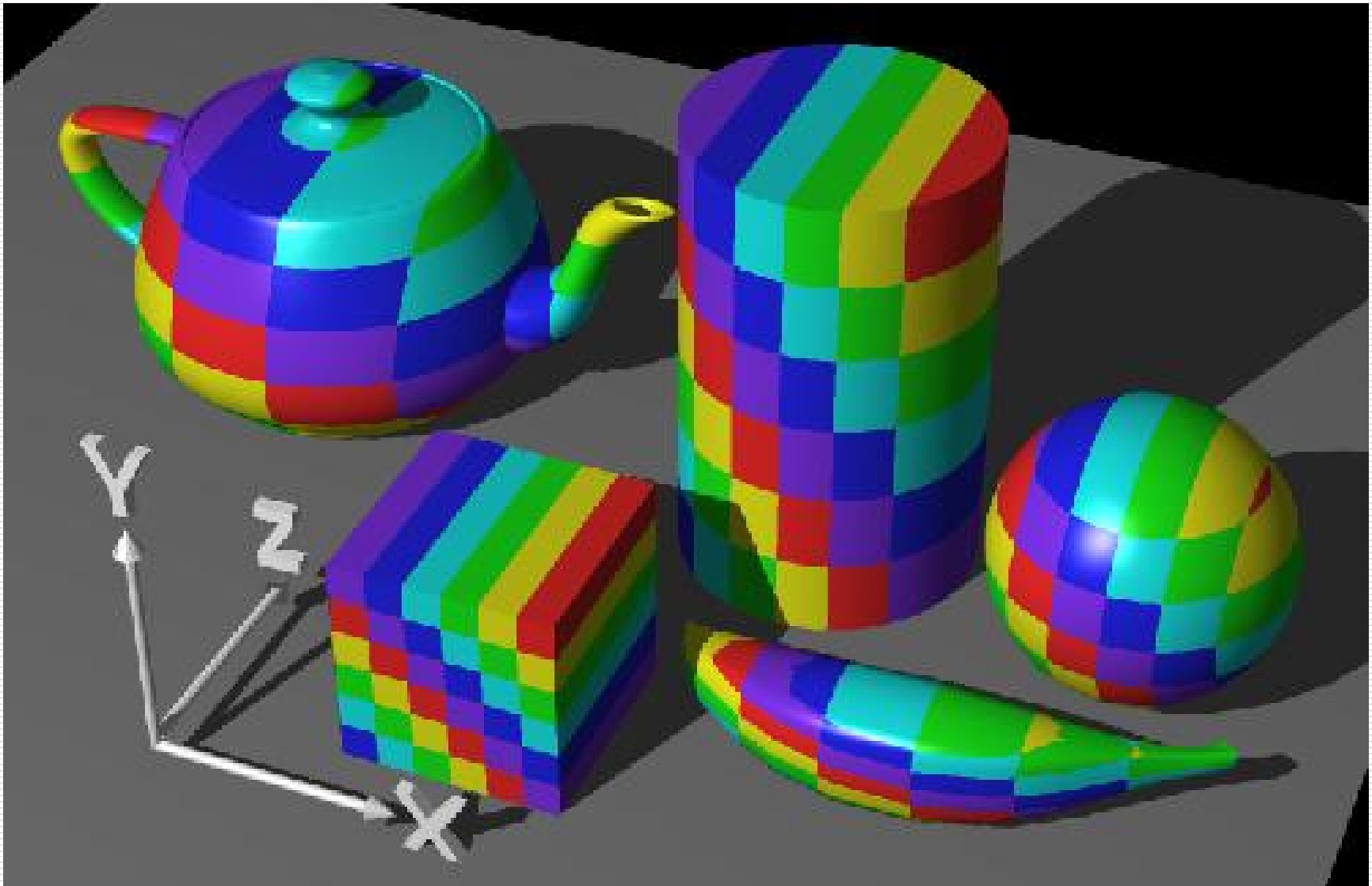
Spherical Mapping



Cylindrical Mapping

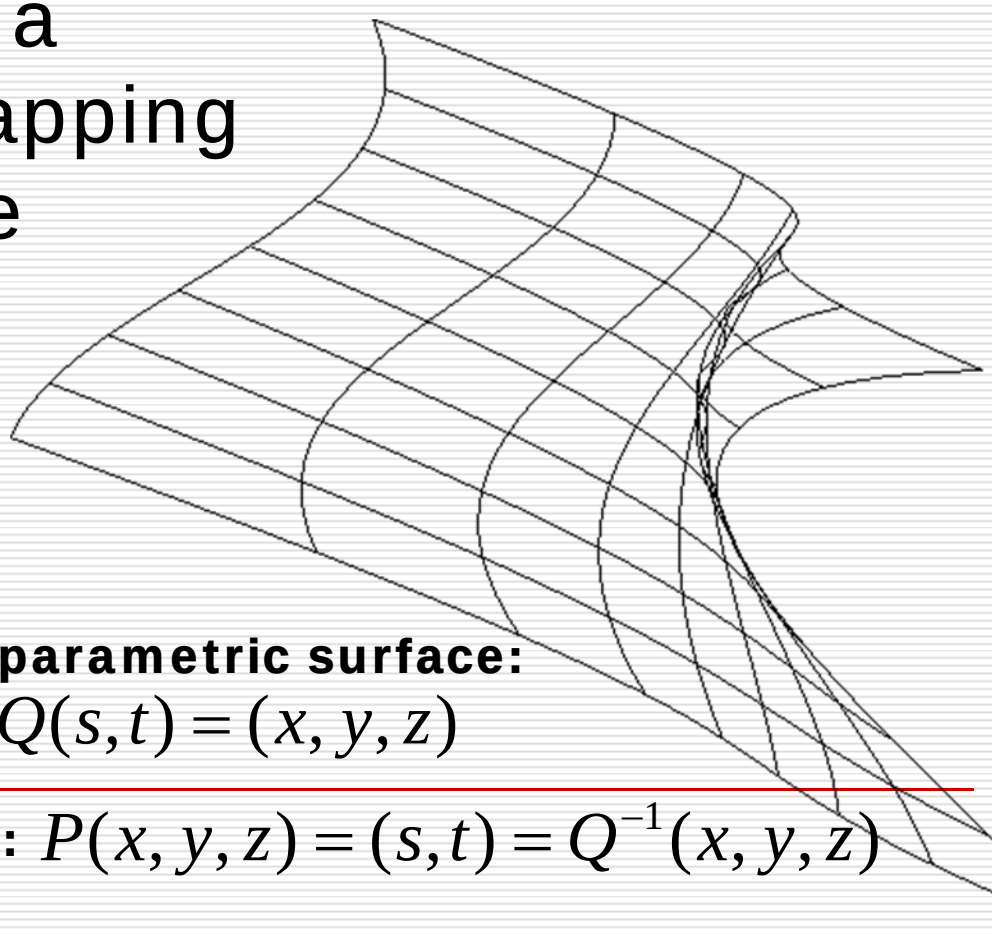


Planar Mapping



Parameterization

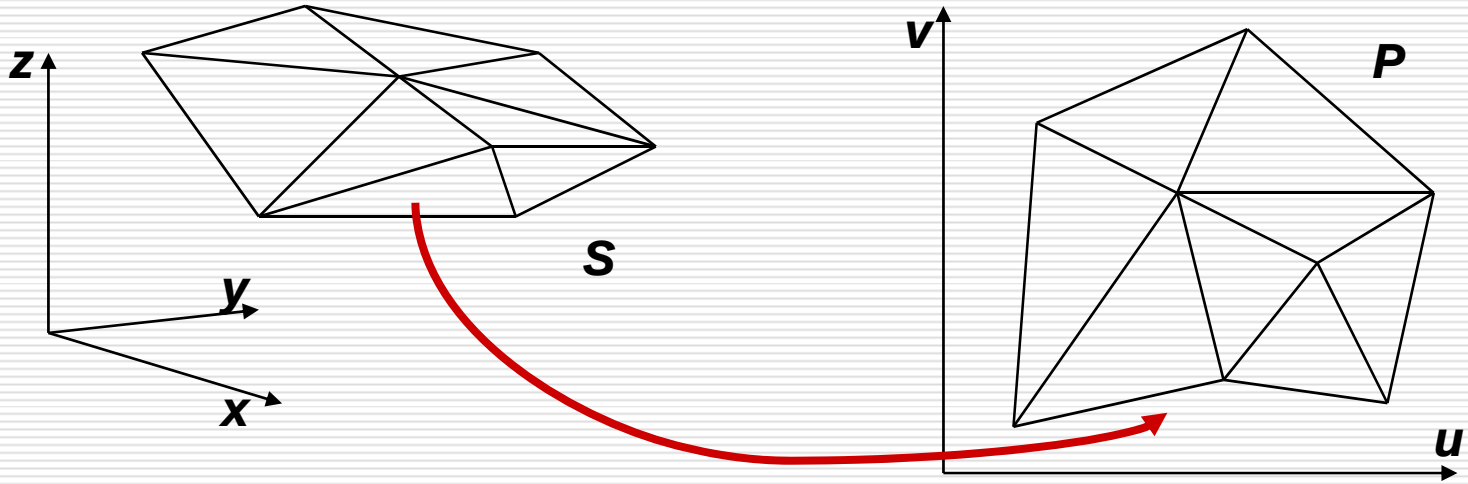
- A parameterization of a surface is a one-to-one mapping from a suitable domain to the surface.



parametric surface:
$Q(s,t) = (x, y, z)$

surface parameterization: $P(x, y, z) = (s,t) = Q^{-1}(x, y, z)$

Parameterization

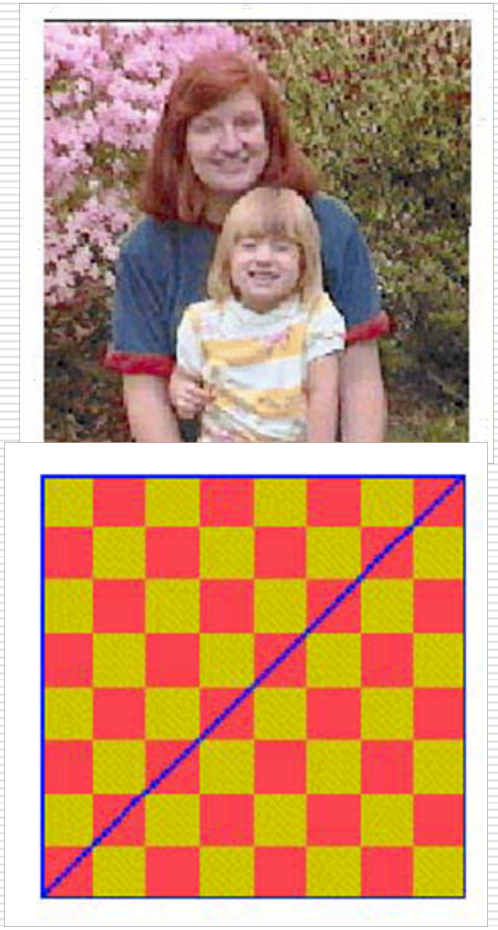


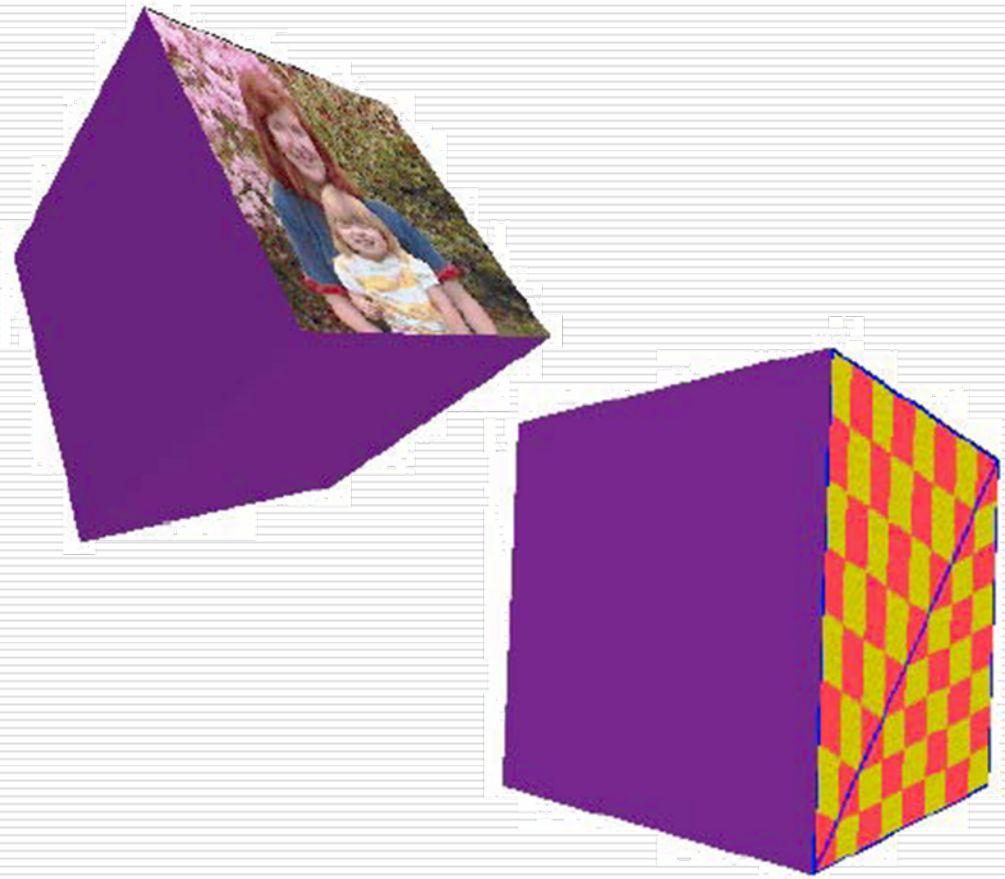
- How to get ***P*** from ***S*** ?
  - for each vertex of ***S***, find its (u,v)
  - from (u,v) of ***P***, map image to ***S***
- A parameterization of a surface is a mapping ρ: (x,y,z)->(u,v) from 3D space to 2D space

Texture Mapping & Polygon Rasterization

- Problems of linearly interpolating texture coordinates in screen space.
  - Similar problems in the interpolations of colors for Gouraud shading and normals for Phong shading.
- Problems in antialiasing
  - Expanding and shrinking

Linear Interpolation of Textures

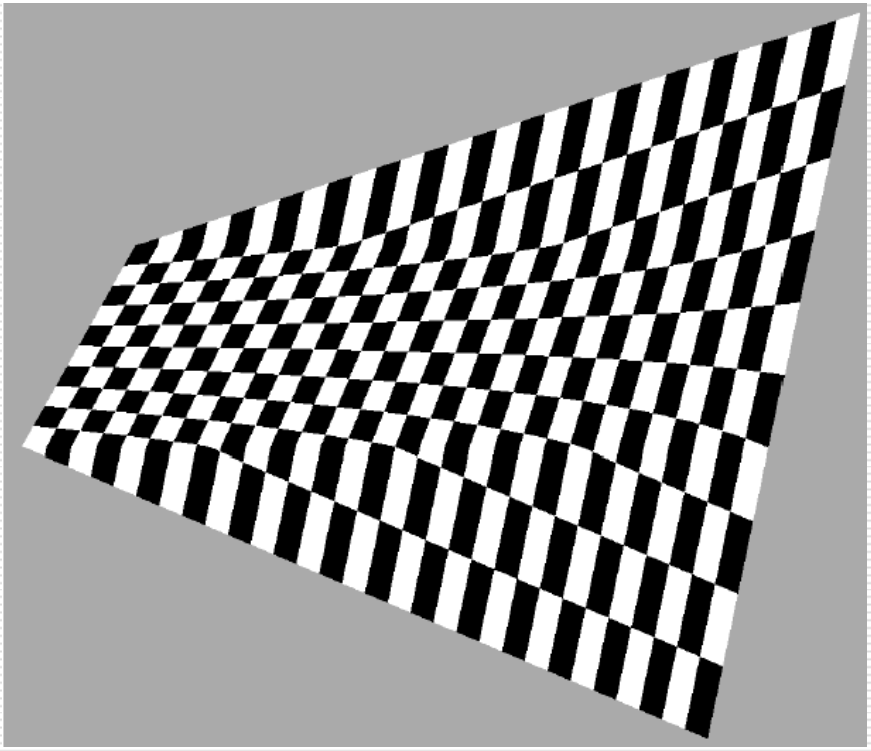


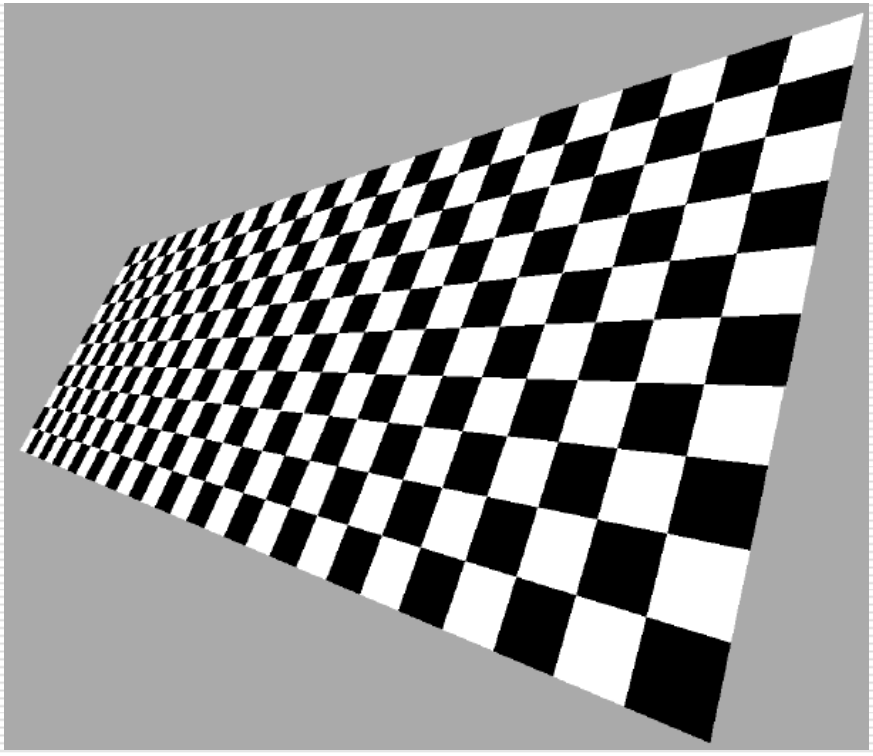


Perspective Correction

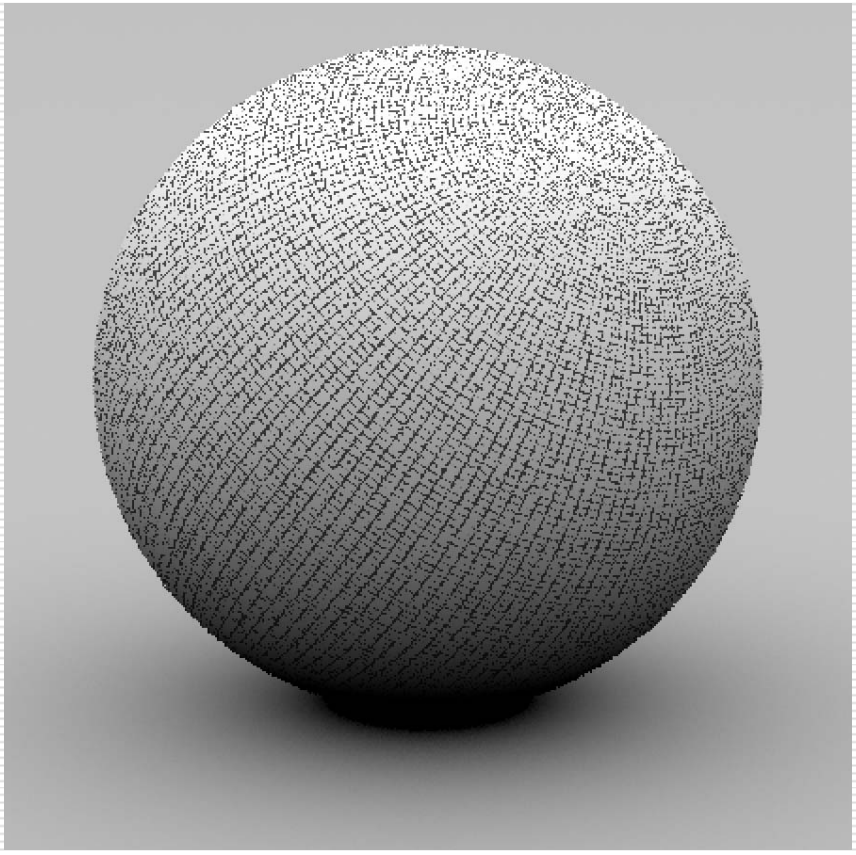
- Dividing the texture coordinates by w
- Linearly interpolating (u/w, v/w, 1/w)
- At each pixel, dividing (interpolated) u/w and v/w by (interpolated) 1/w

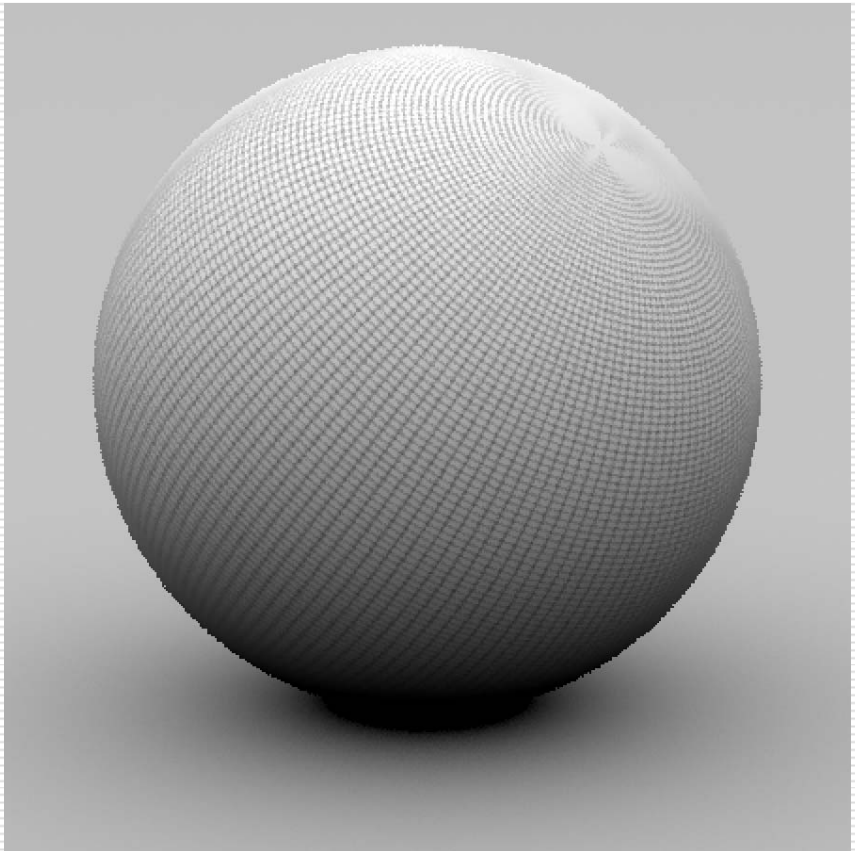
Example





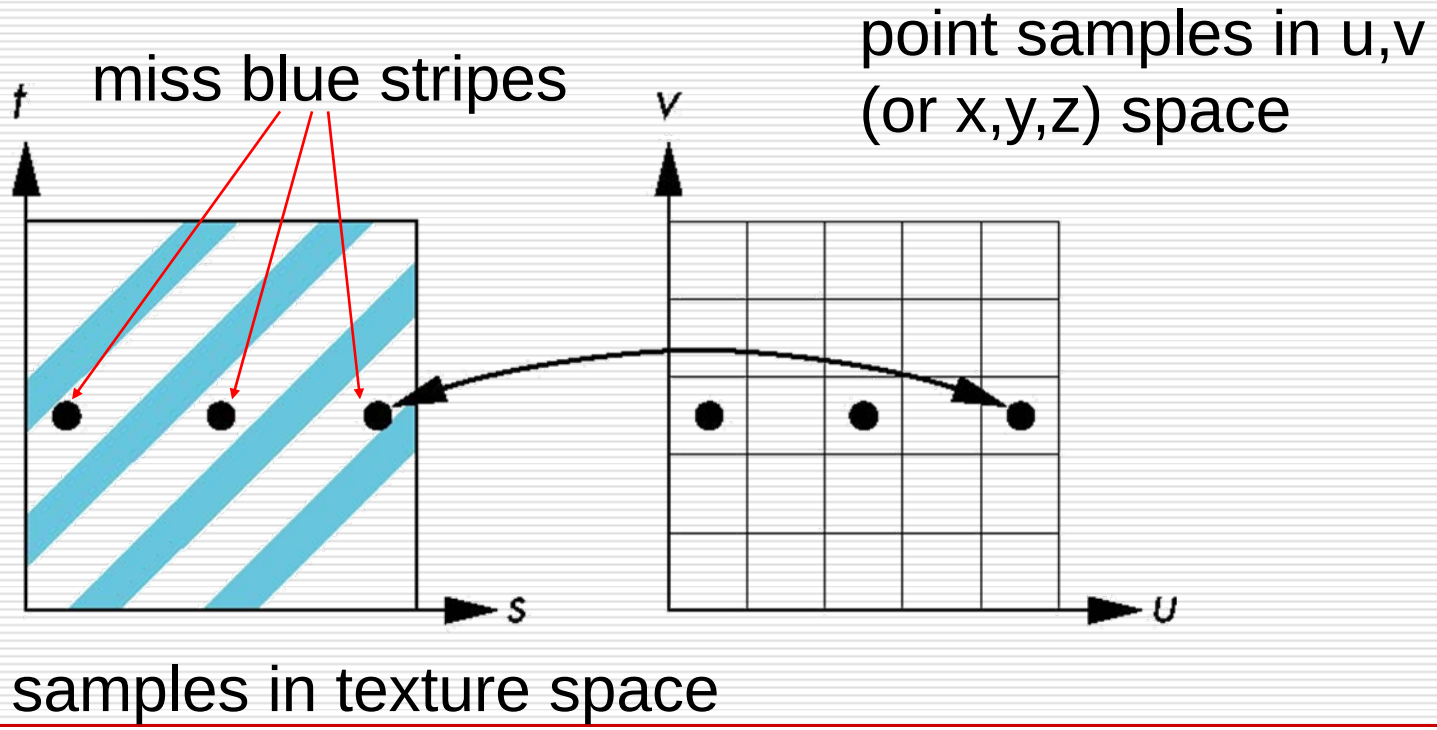
Antialiasing





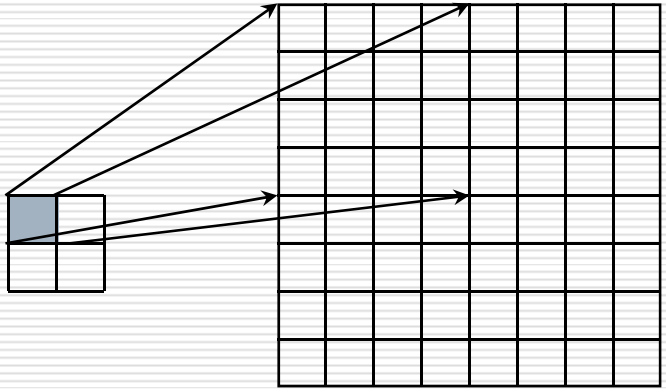
Aliasing

- Point sampling of the texture can lead to aliasing errors

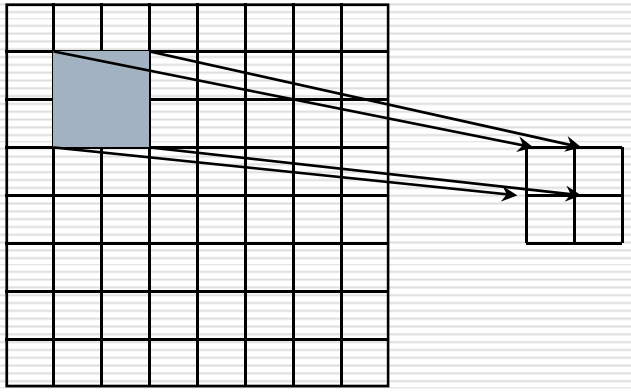


Magnification and Minification

- Example:



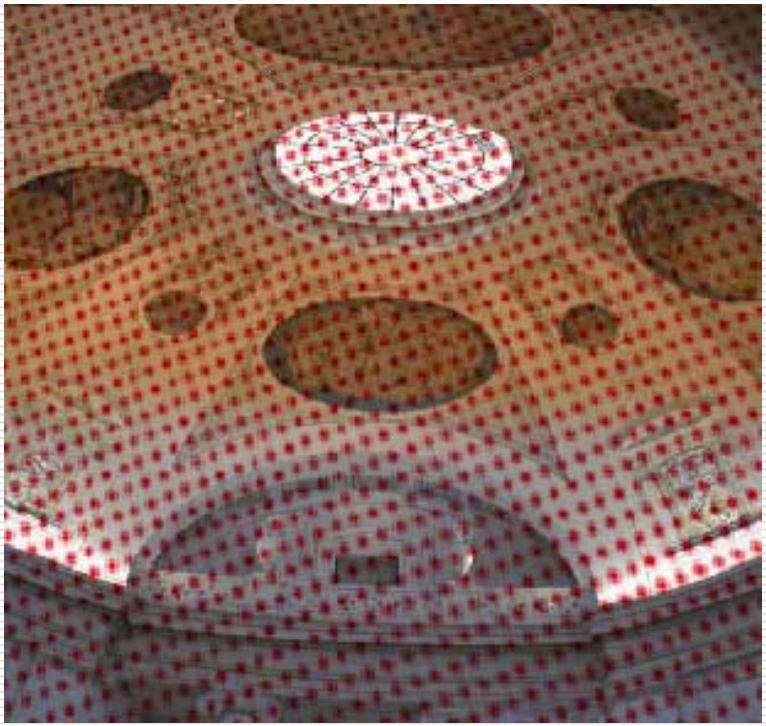
Magnification



Minification

Sampling Texture Maps

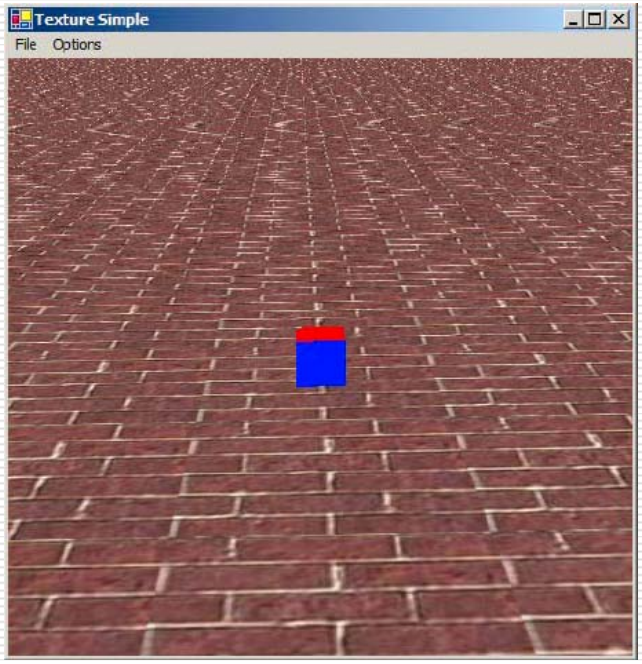
over-sampling



under-sampling

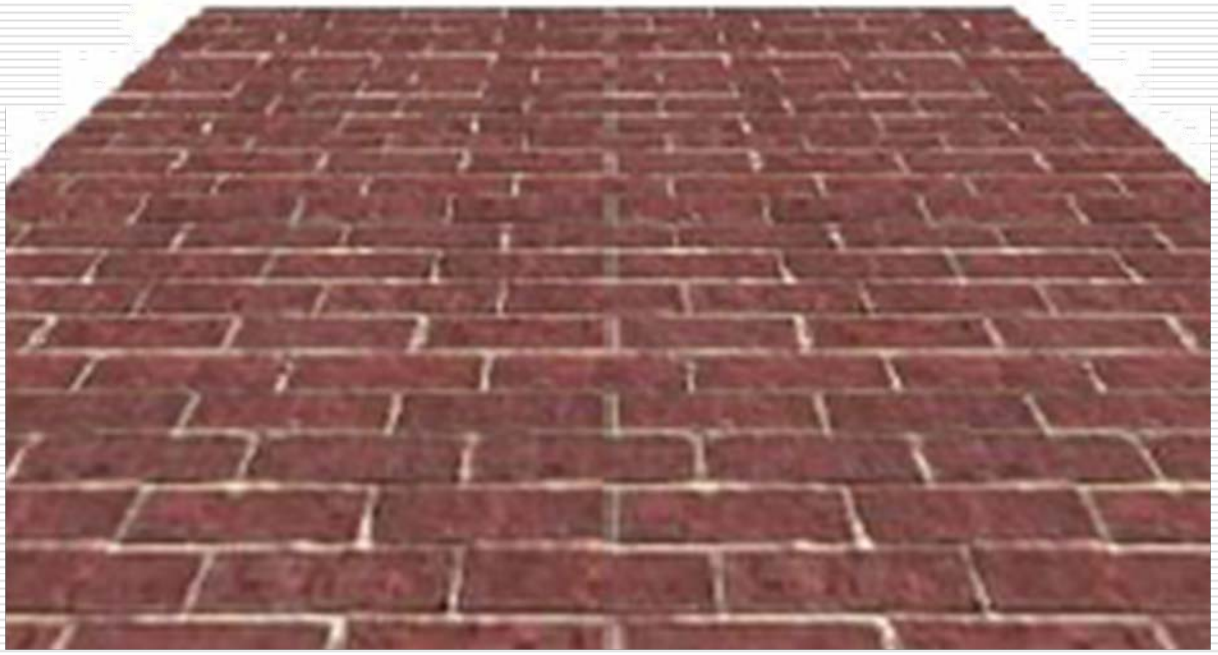
Over-sampling

- indicative of aliasing
  - high-frequency details showing up in areas where we expect to see low frequencies

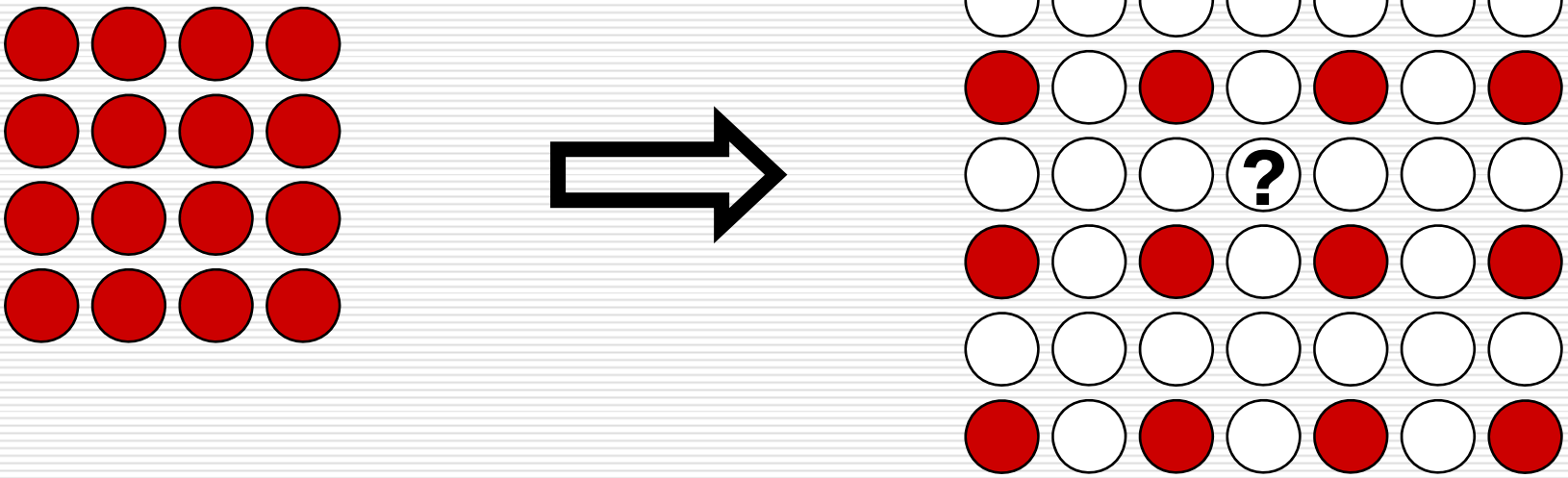


Spatial Filtering

- *prefilter* the texture to remove the high frequencies that show up as artifacts

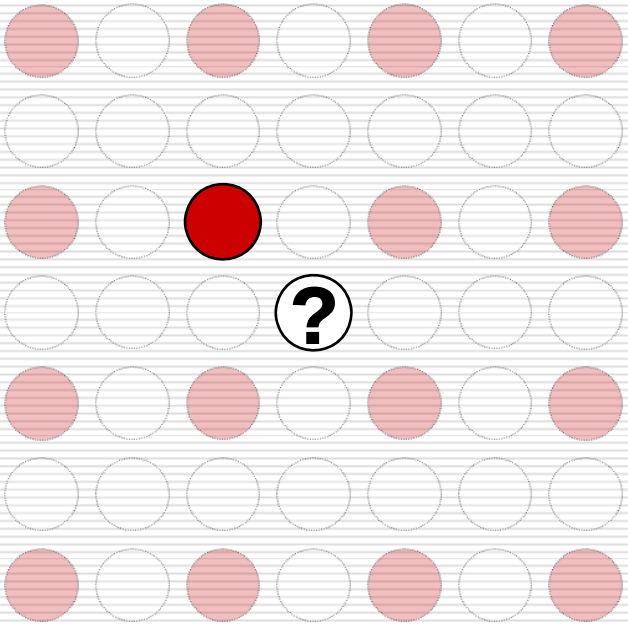


Changing Resolution



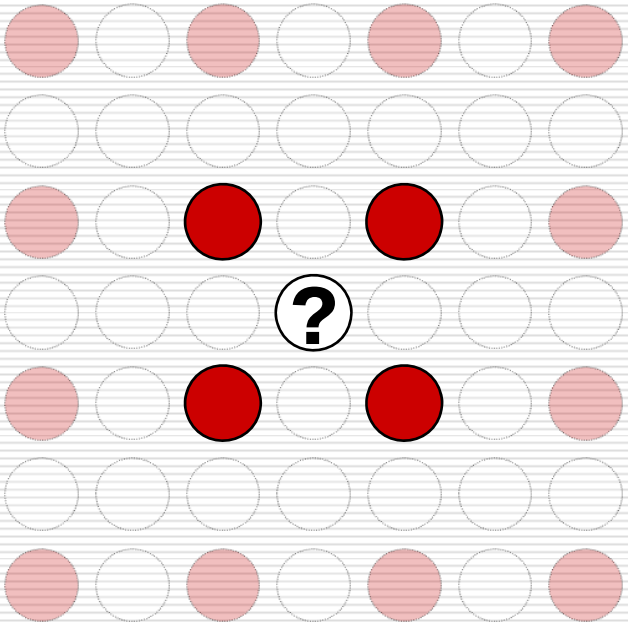
Nearest Neighbor

- a.k.a.
  zero order interpolation
- use 1 nearest neighbor



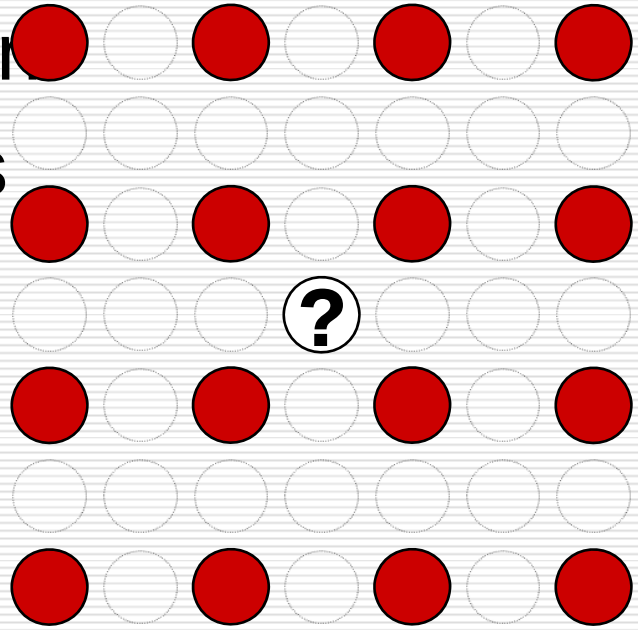
Bilinear

- a.k.a.
  first order interpolation
- use 4 nearest neighbors



Bicubic

- a.k.a.
  second order interpolation
- use 16 nearest neighbors




nearest neighbor


bilinear

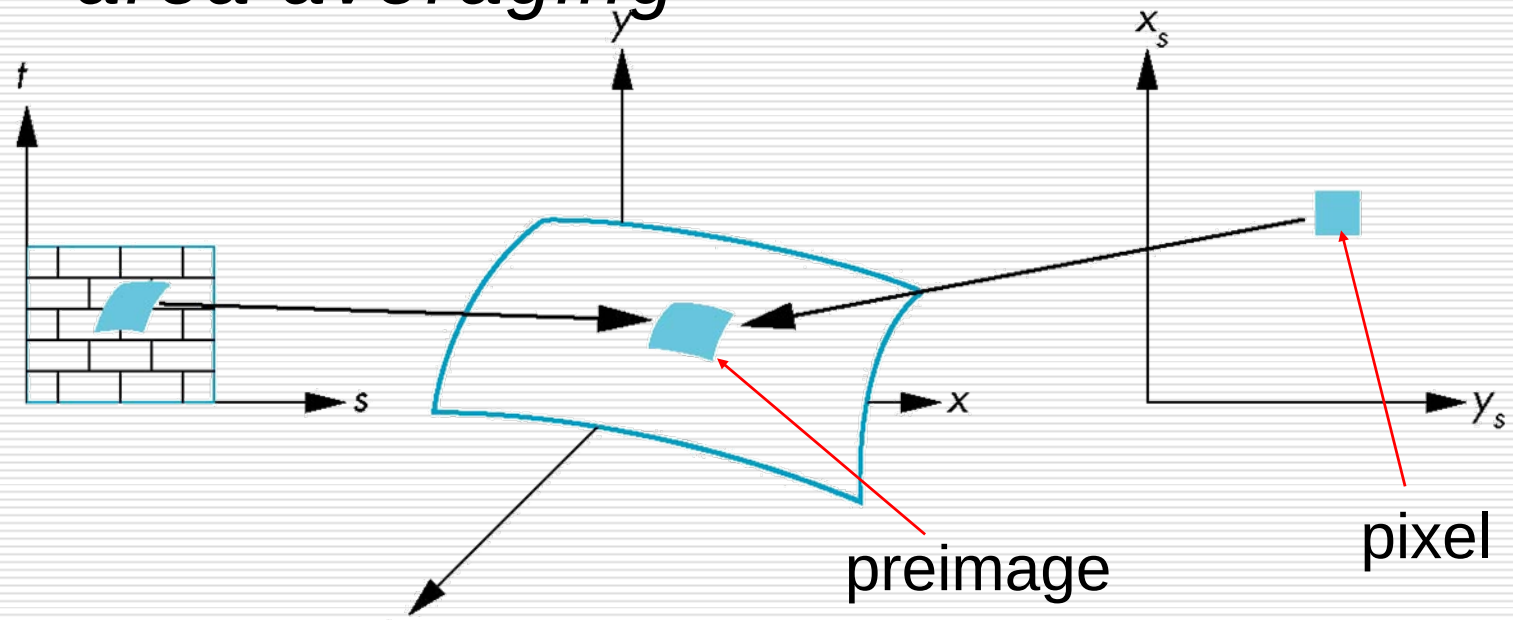

bicubic


ground truth


bicubic

Area Averaging

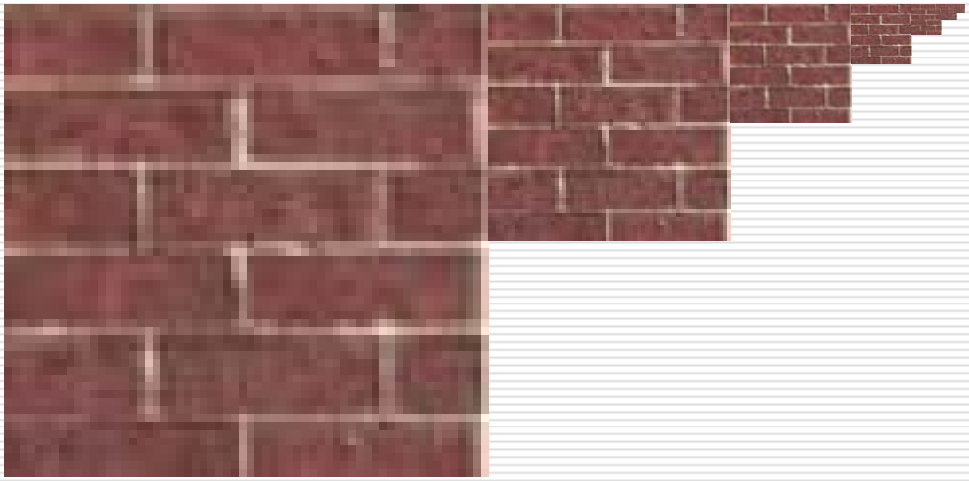
- A better but slower option is to use *area averaging*



- Note that *preimage* of pixel is curved

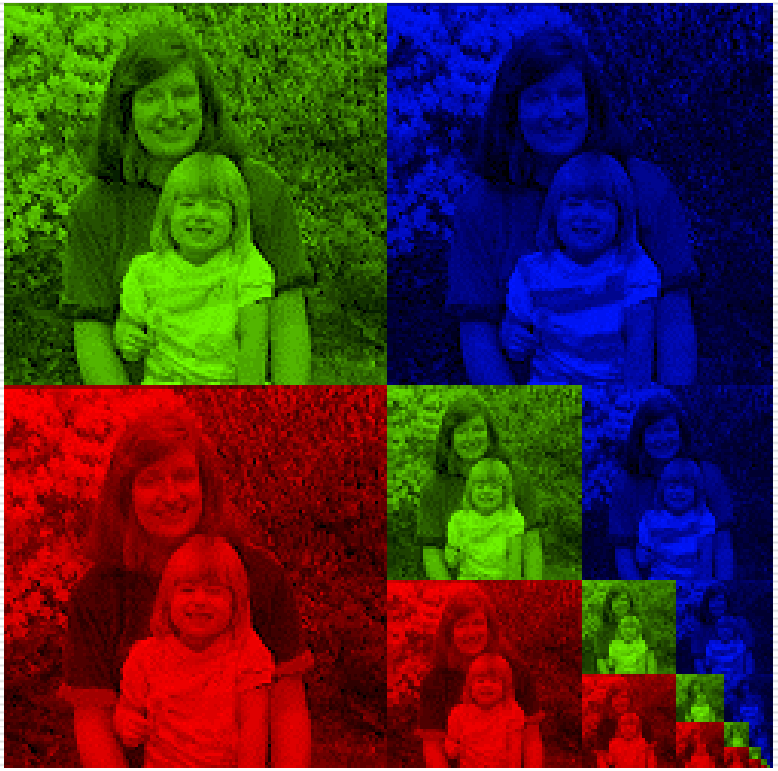
MIP Mapping

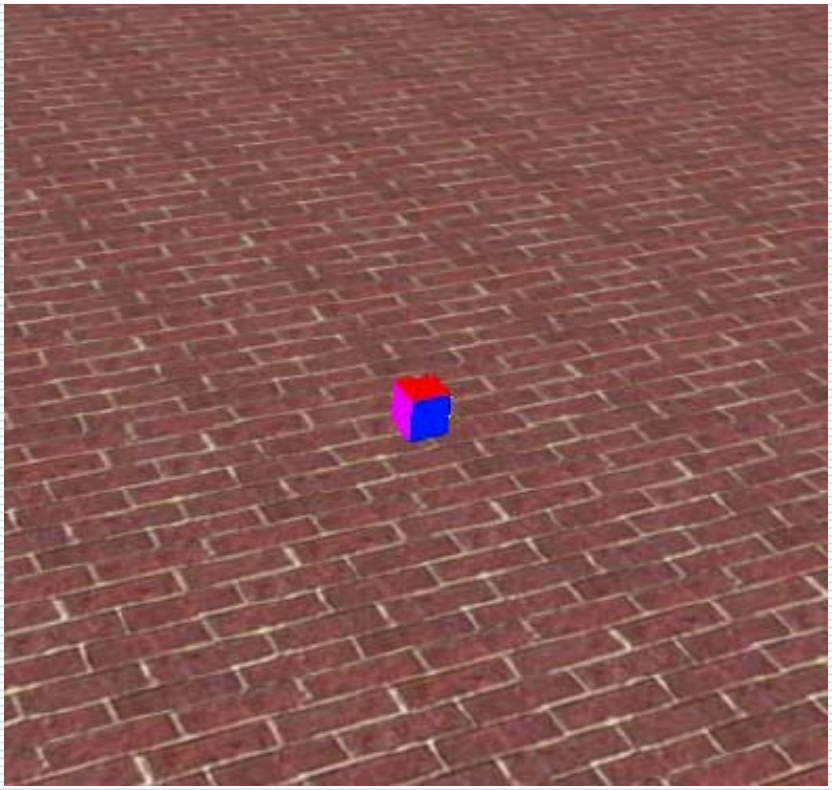
- MIP Mapping is one popular technique for precomputing and performing this prefiltering



- Computing this series of filtered images requires only a small fraction of additional storage over the original texture

Storing MIP Maps





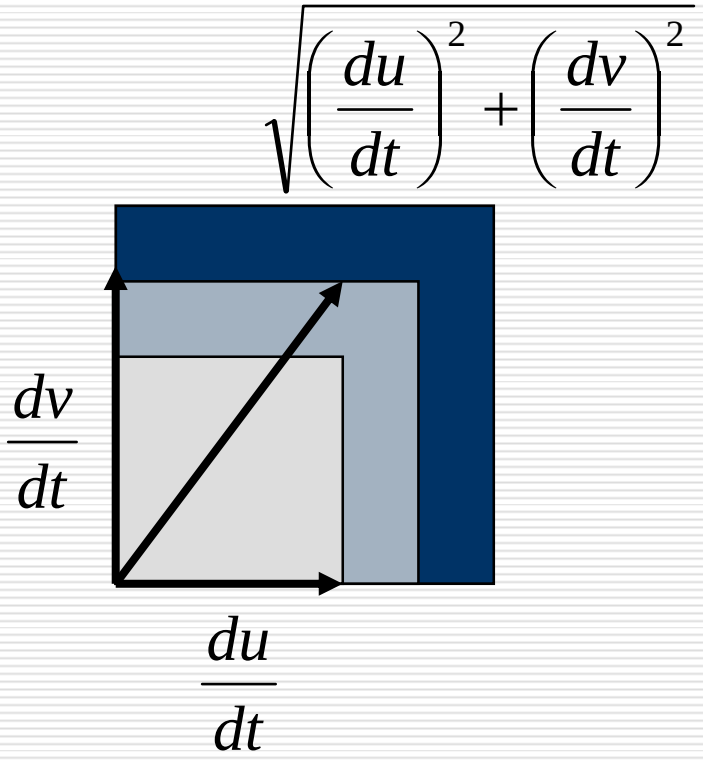
Finding the MIP Level

$$\frac{du}{dt} = \frac{du}{ds}\frac{ds}{dt} = (u_2 - u_1)\frac{w_1 w_2}{(w_1 + t(w_2 - w_1))^2}$$

$$\frac{dv}{dt} = \frac{dv}{ds}\frac{ds}{dt} = (v_2 - v_1)\frac{w_1 w_2}{(w_1 + t(w_2 - w_1))^2}$$

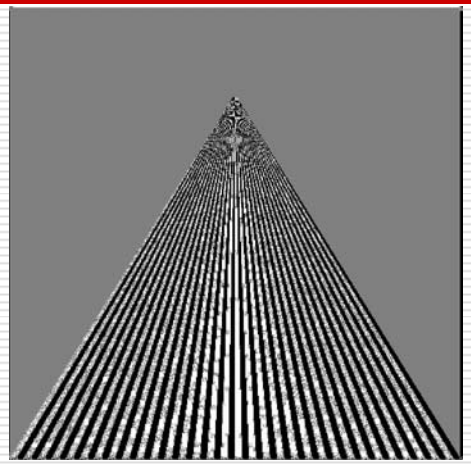
$$level = \log_2\left(\sqrt{\left(\frac{du}{dt}\right)^2 + \left(\frac{dv}{dt}\right)^2}\right)$$

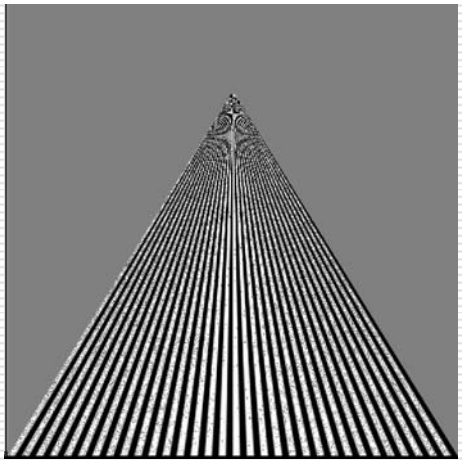
$$level = \log_2\left(Max\left(\left|\frac{du}{dt}\right|, \left|\frac{dv}{dt}\right|\right)\right)$$



Example

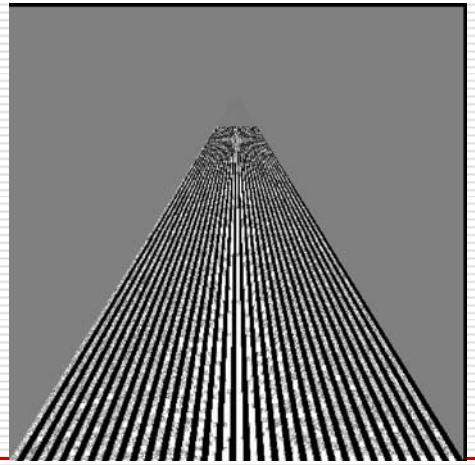
point sampling

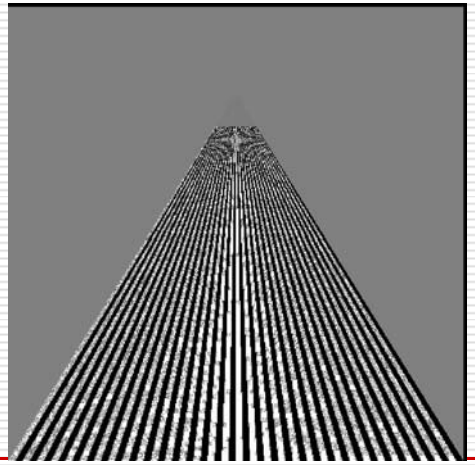




linear filtering

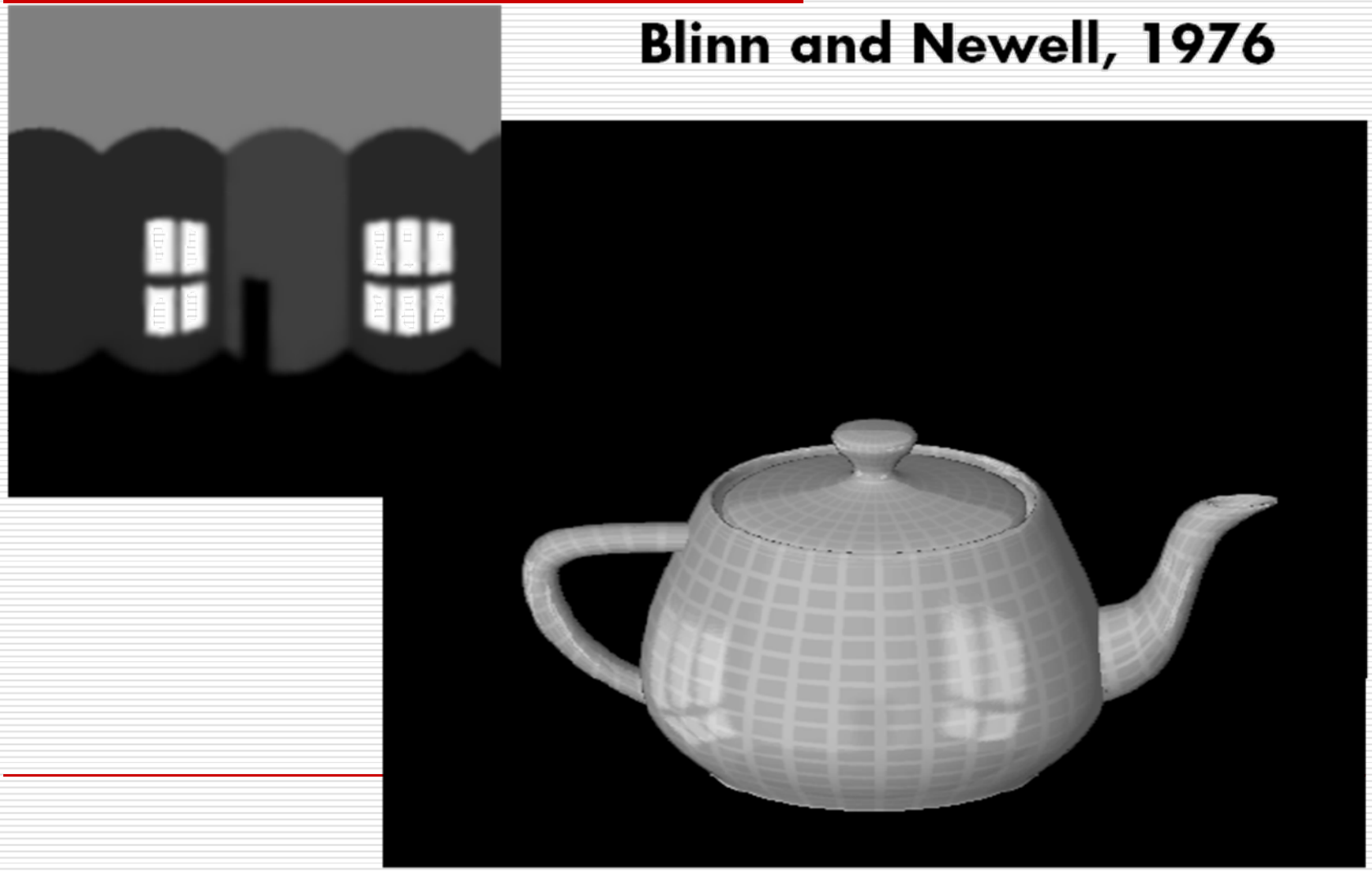
mipmapped point sampling





mipmapped linear filtering

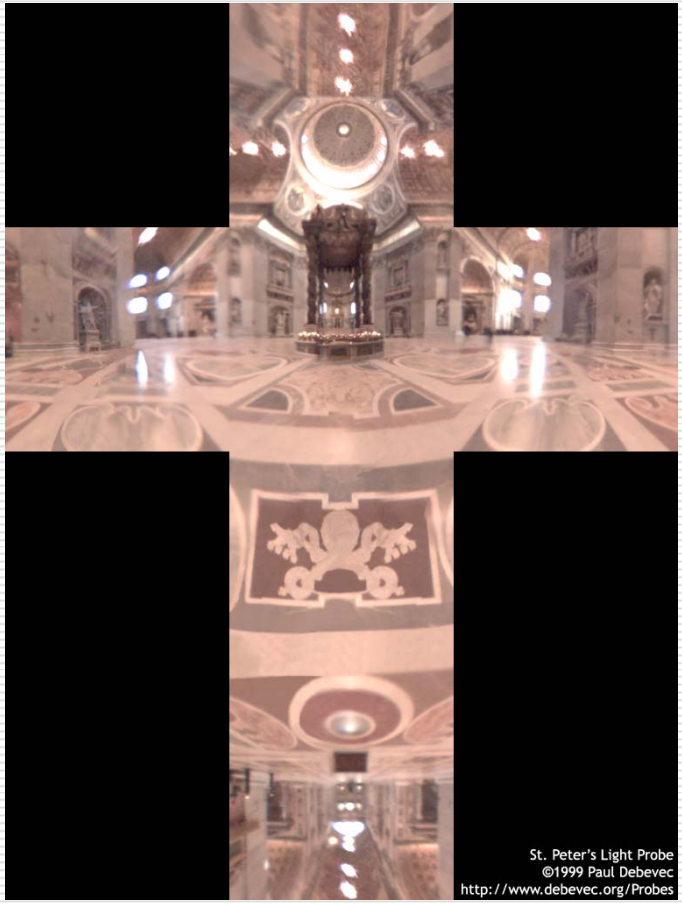
Reflection Maps



Environment Mapping

Sphere Mapping

Box Maps

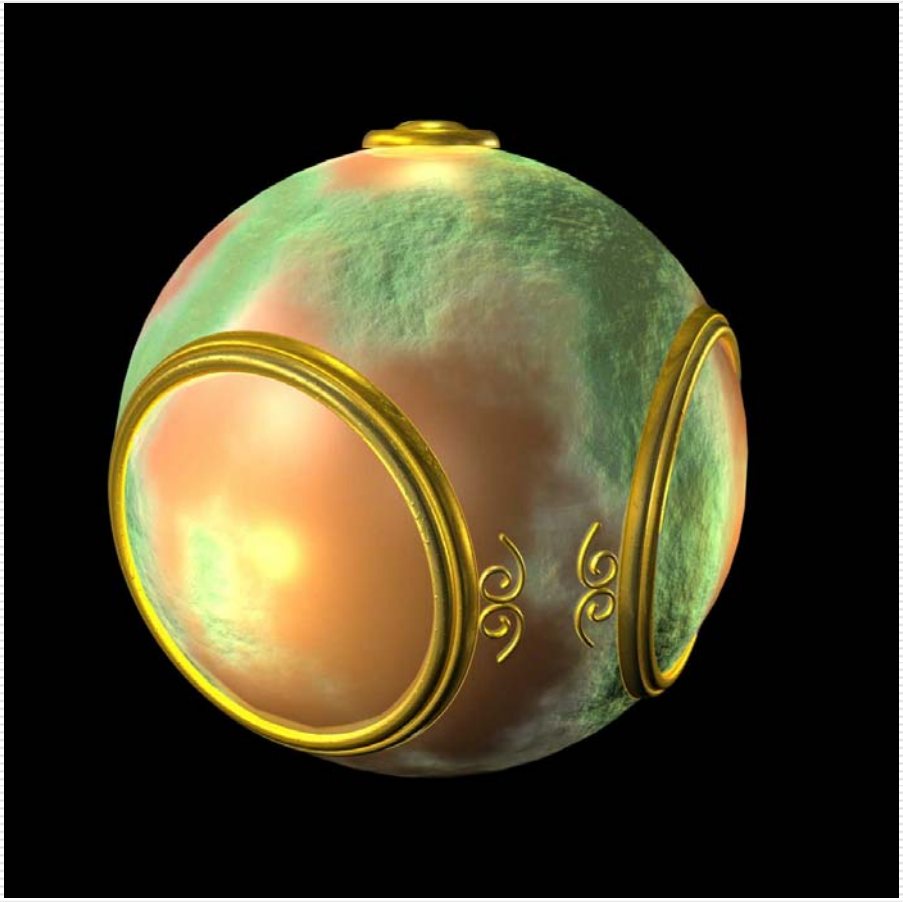


Environment Maps

ray traced

environment map

Bump Mapping



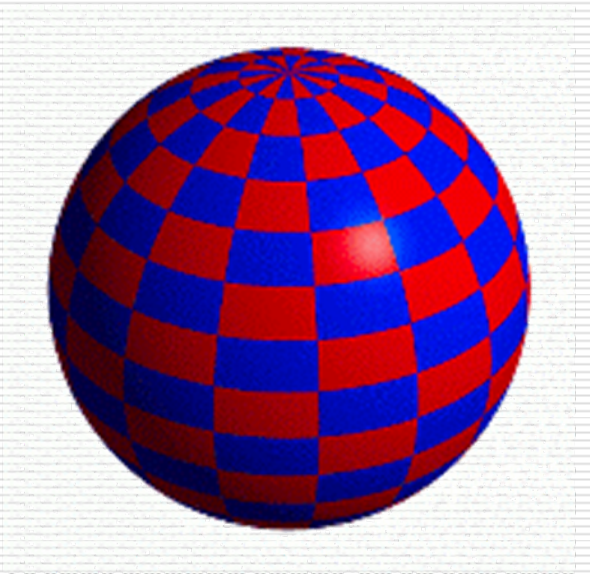
Bump Mapping

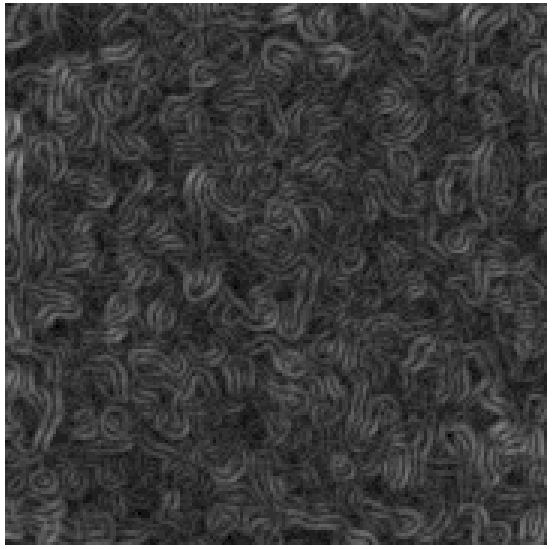
- Textures can be used for more than just color

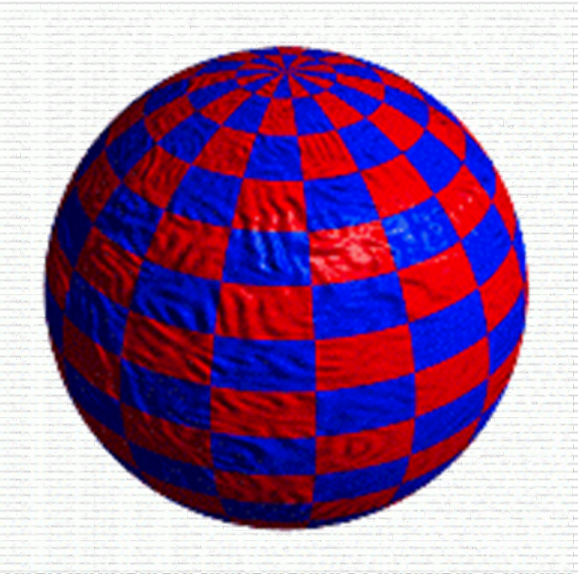
$$I = k_{\mathrm{a}} I_{\mathrm{a}} + \sum_i f_{\mathrm{att}_i} I_{\mathrm{p}_i} [k_{\mathrm{d}} (\vec{N} \bullet \vec{L}_i) + k_{\mathrm{s}} (\vec{R}_i \bullet \vec{V})^n]$$

- In bump mapping, a texture is used to perturb the normal:
  - The normal is perturbed in each parametric direction according to the partial derivatives of the texture.

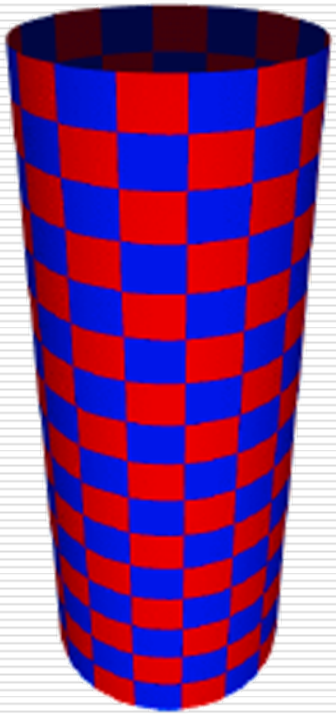
Bump Mapping

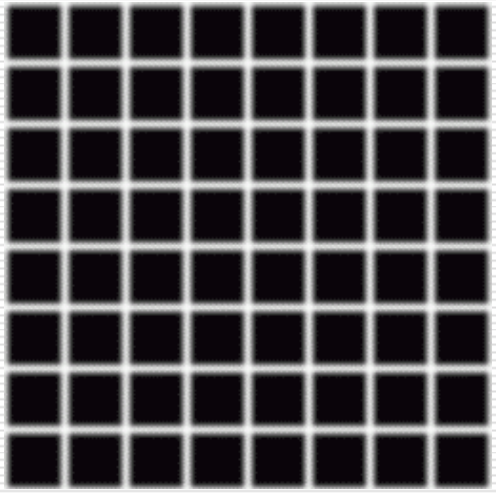


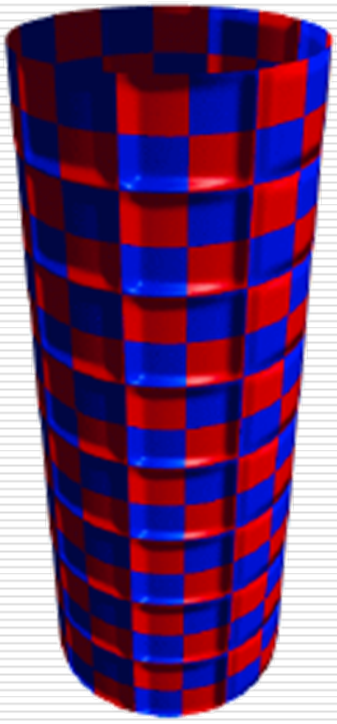




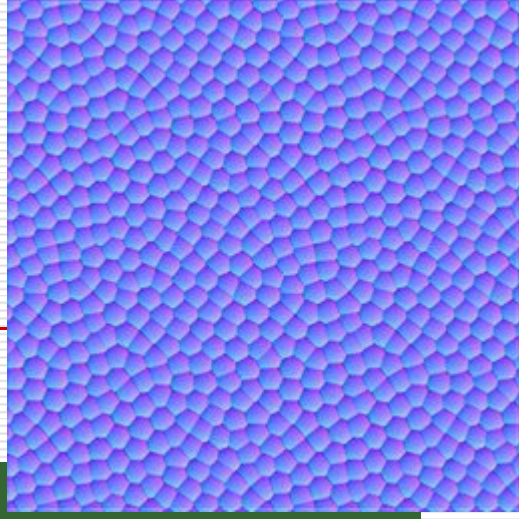
Bump Mapping

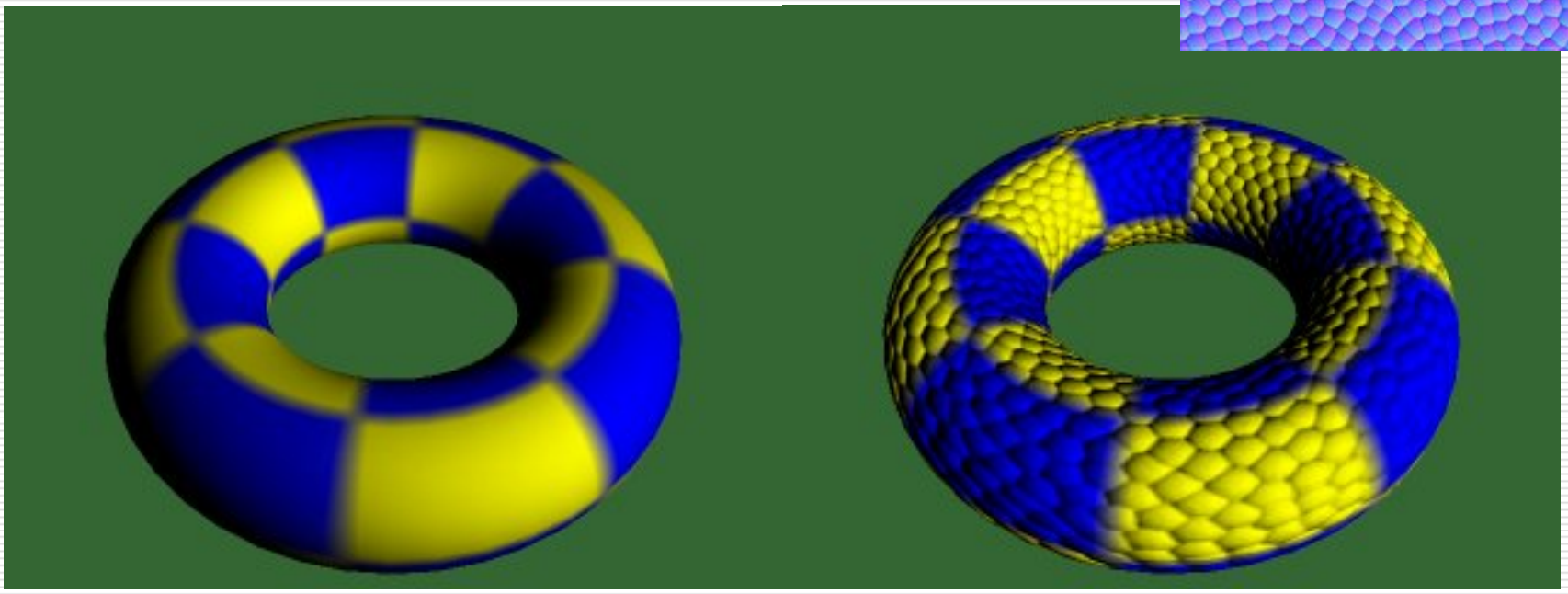






Bump Mapping

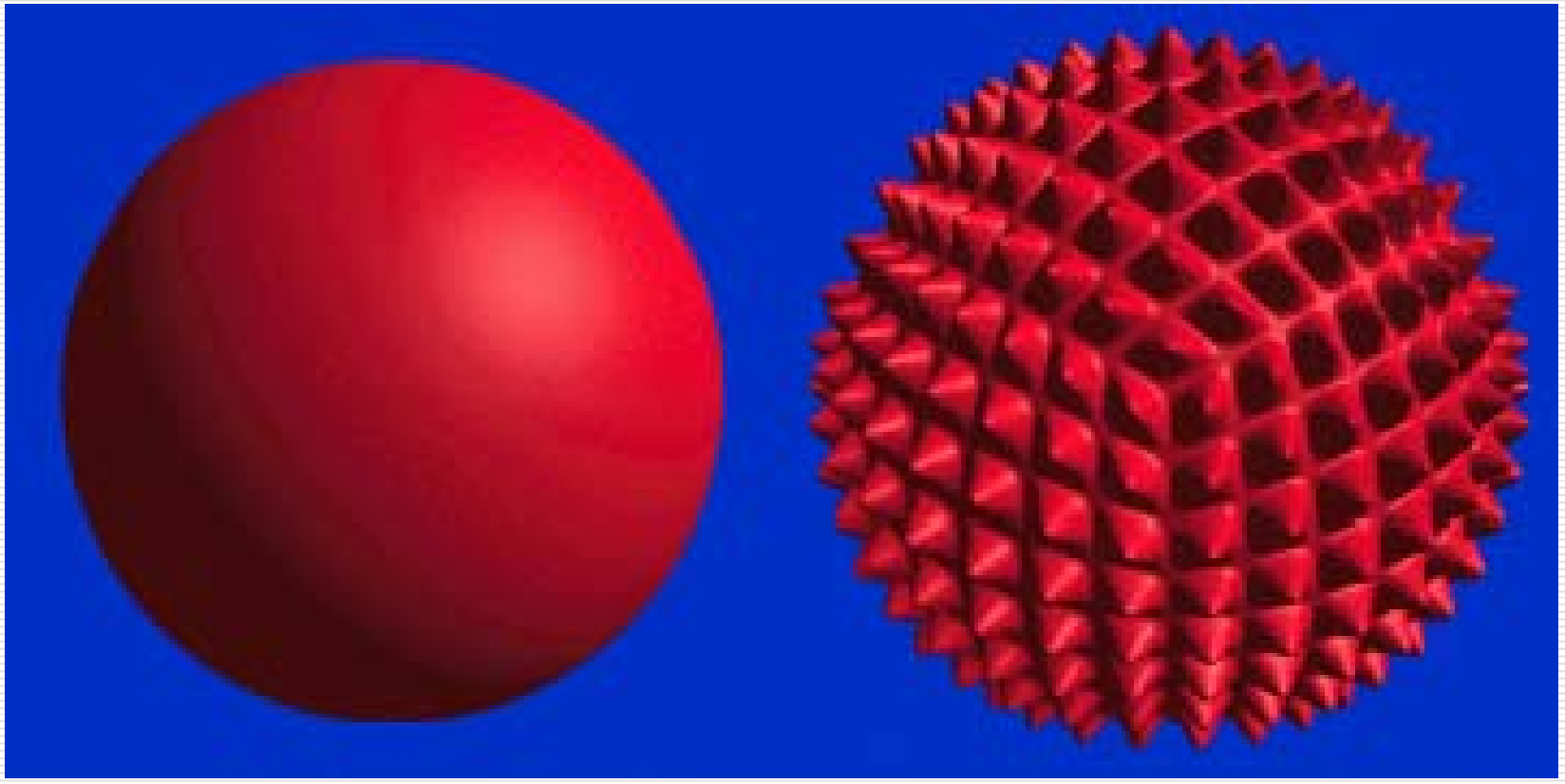




Displacement Mapping

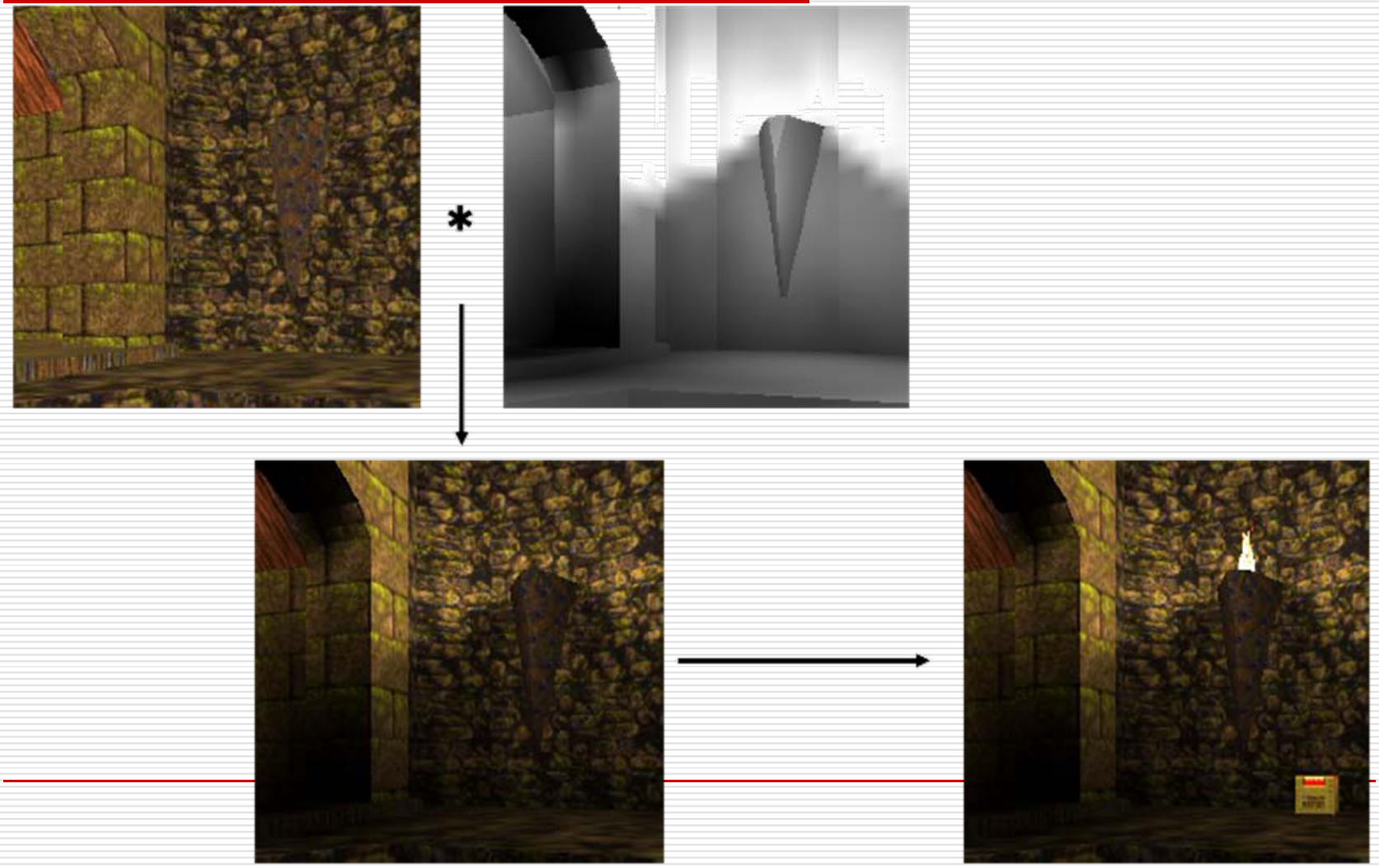
- In displacement mapping, a texture is used to perturb the surface geometry itself
  - Silhouettes are correct
  - Requires doing additional hidden surface calculations

Displacement Mapping



Bump Mapping & Displacement Mapping

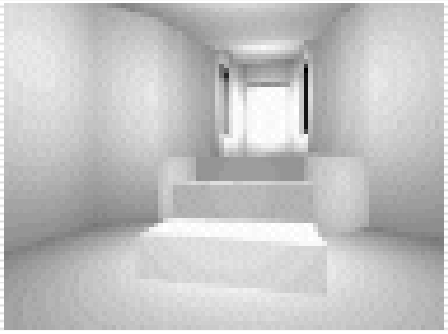
Illumination Maps



Texture Mapping in Quake

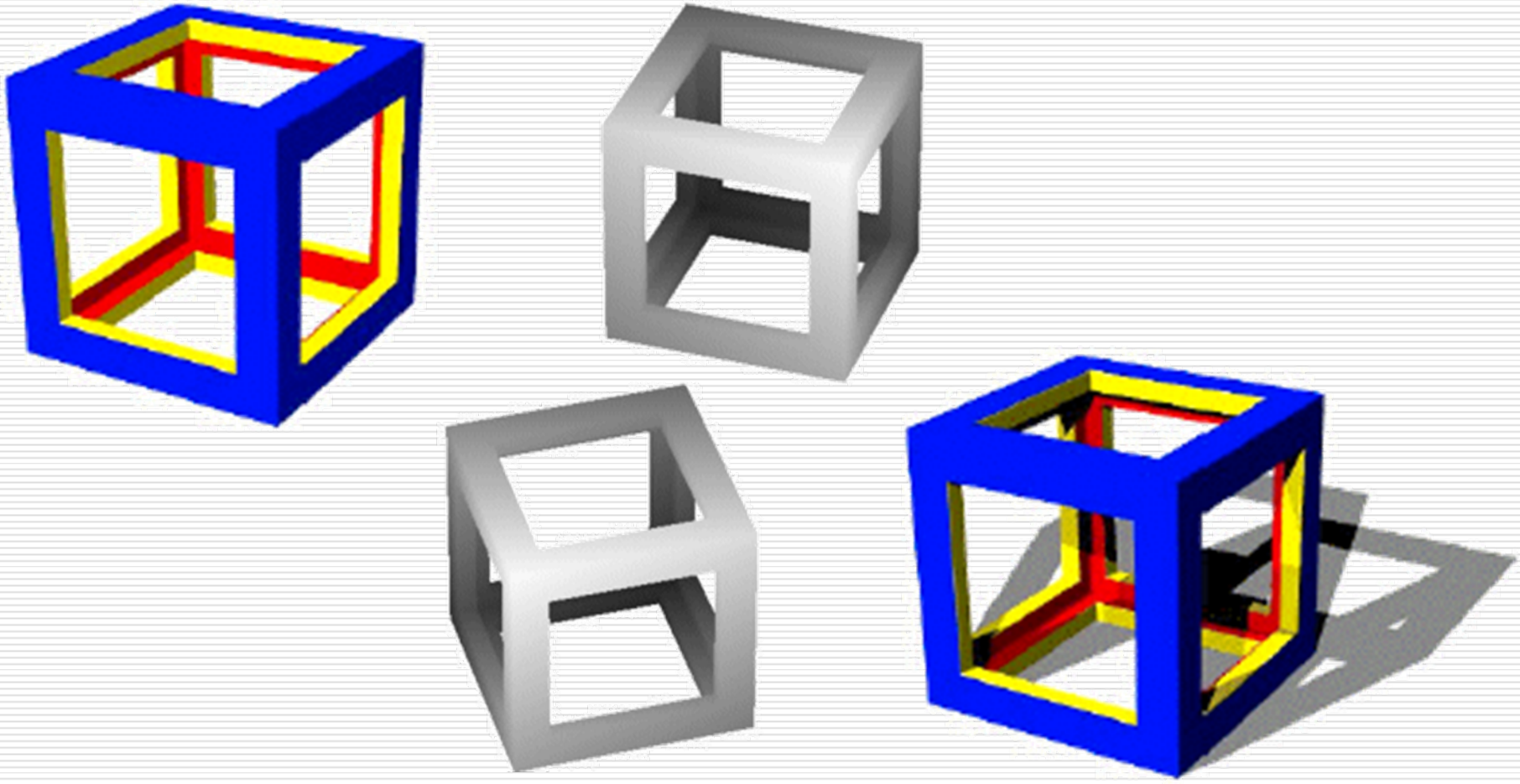
Texture Only

Texture & Light Maps



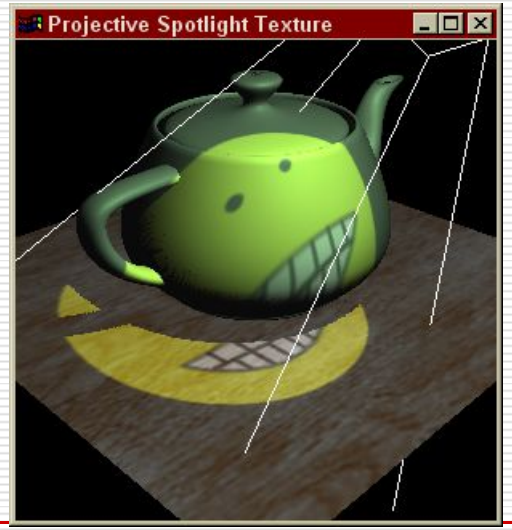
Light Map

Shadow Maps



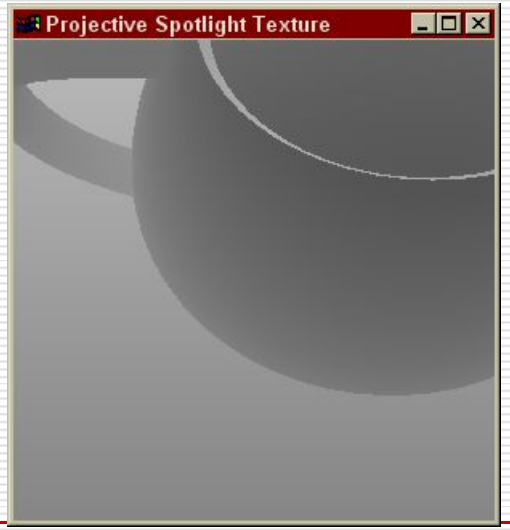
Basic Steps of Shadow Maps

- Render the scene from the light's point of view,
- Use the light's depth buffer as a texture (shadow map),
- Projectively texture the shadow map onto the scene,
- Use "texture color" (comparison result) in fragment shading.



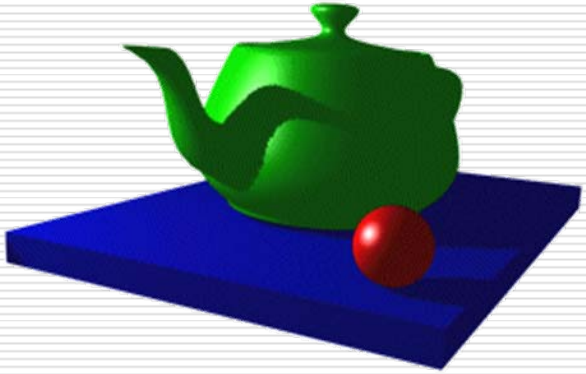
Eye's View

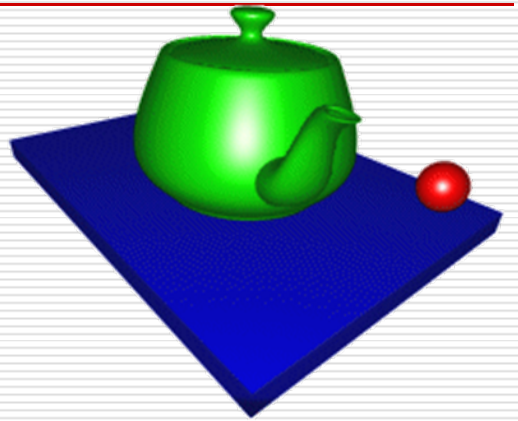
Light's View

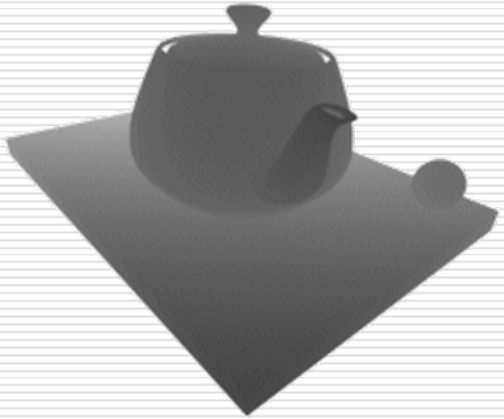


Depth/Shadow Map

Shadow Buffer



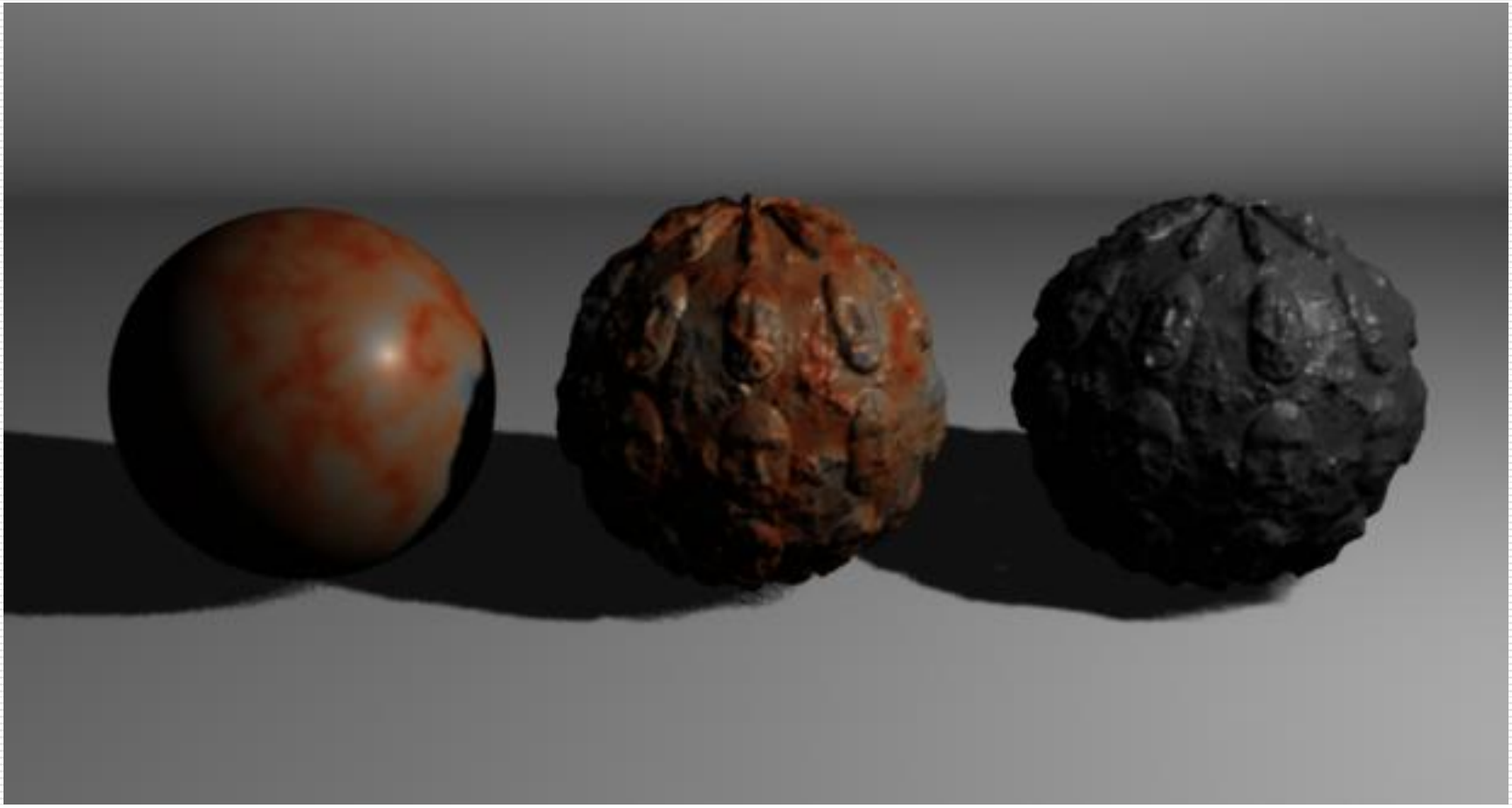




$$I = k_e + k_a I_a + \sum_i S_i f_{att_i} I_{p_i} [k_d (\vec{N} \bullet \vec{L}_i) + k_s (\vec{R}_i \bullet \vec{V})^n]$$

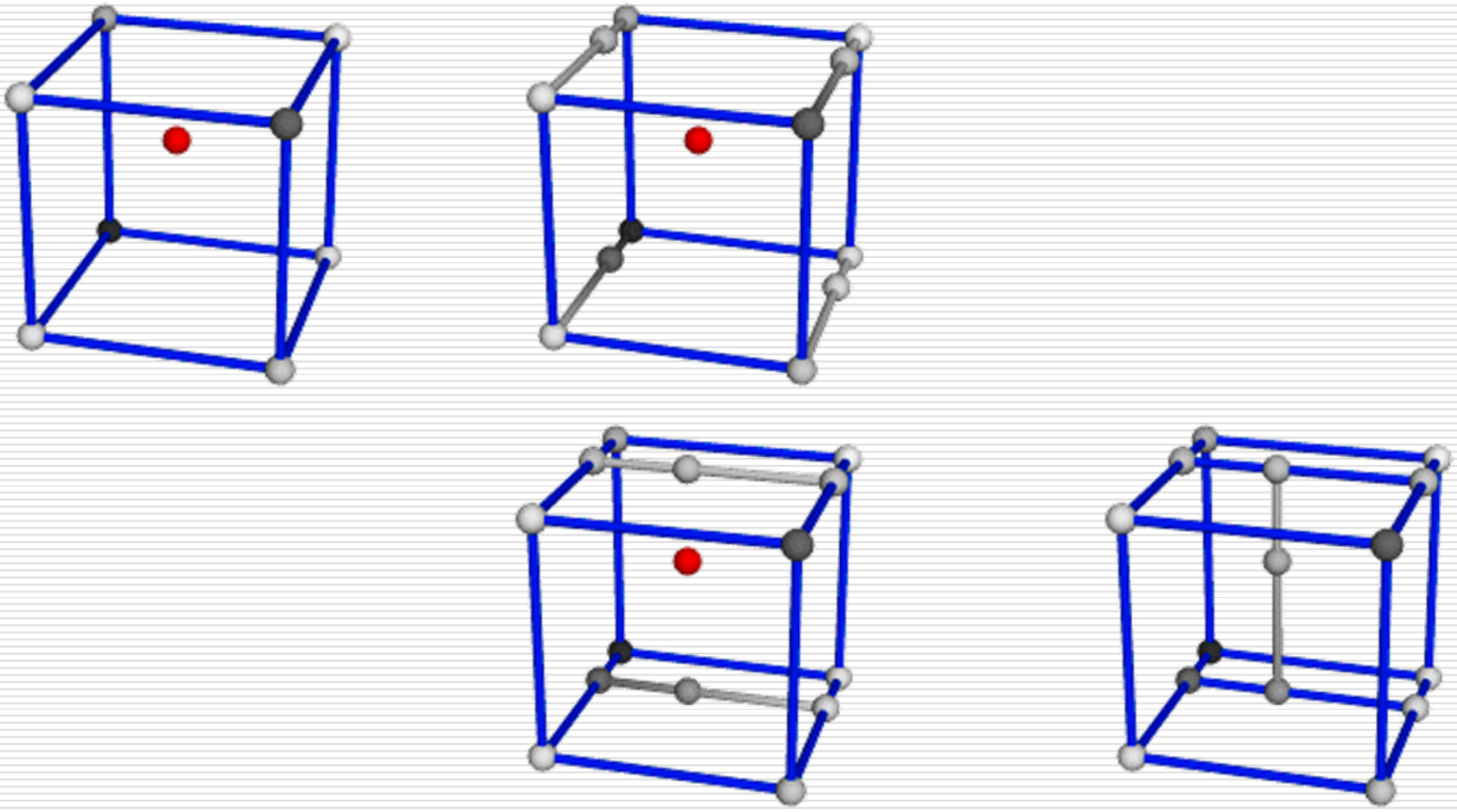
Three Dimensional or Solid Textures

Solid Textures



Solid Textures

Noise (Tri-linear) Interpolation



Turbulence