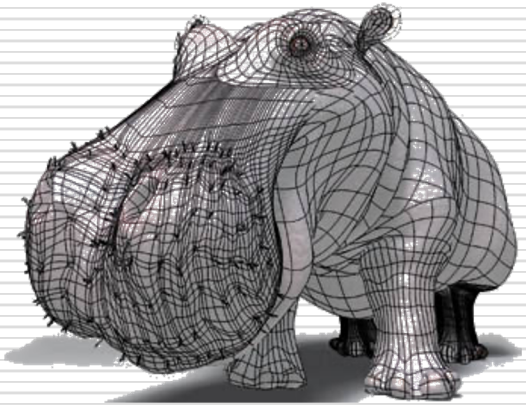
Game Programming

Robin Bing-Yu Chen
National Taiwan University

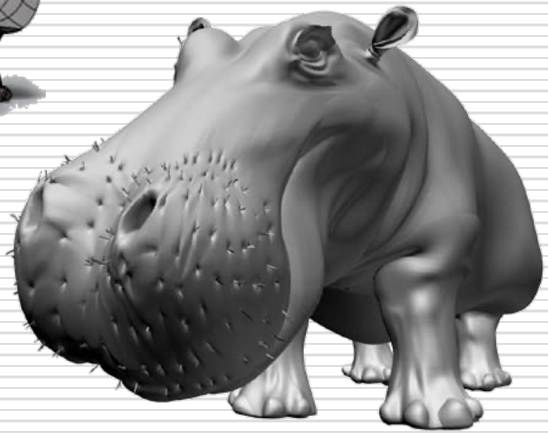
Game Texturing

- Texture Mapping
- Environment Mapping
- Bump Mapping
- Shadow Maps

The Quest for Visual Realism



Model



Model with Shading

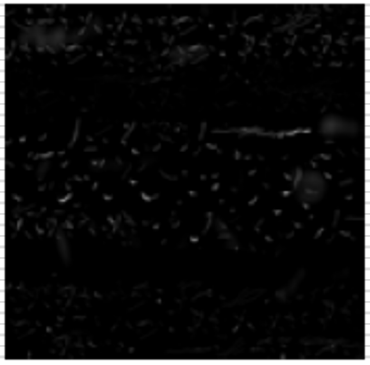
Model with Shading and Textures

Texture Mapping

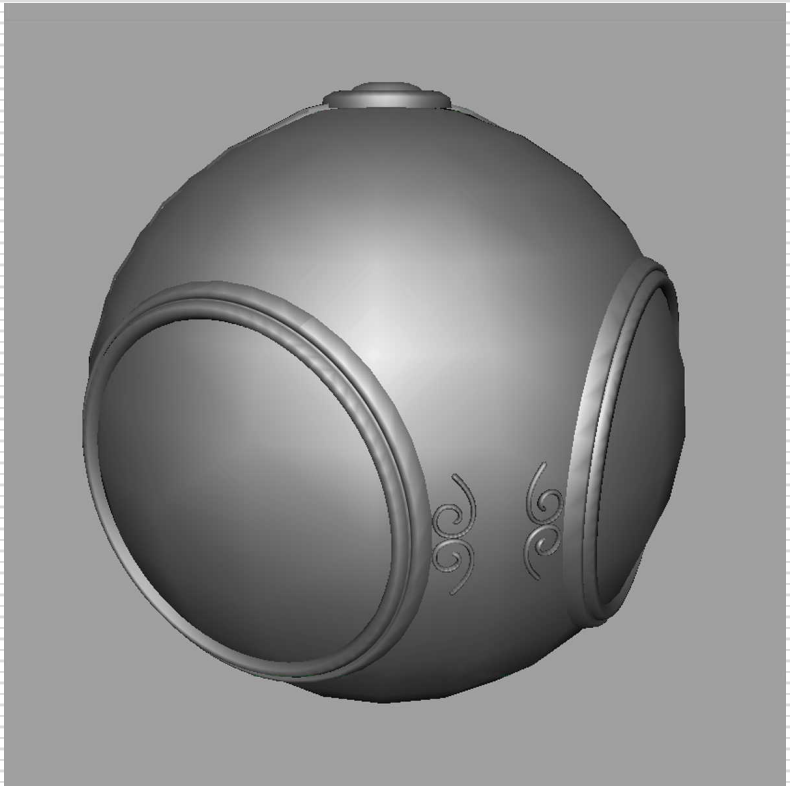
- Previously, we assume that reflection properties such as are constant within each triangle.
- However, some objects have complex appearance which arises from variation in reflection properties.
- The common technique to handle this kind of variation is to store it as a function or a pixel-based image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

Texture Maps

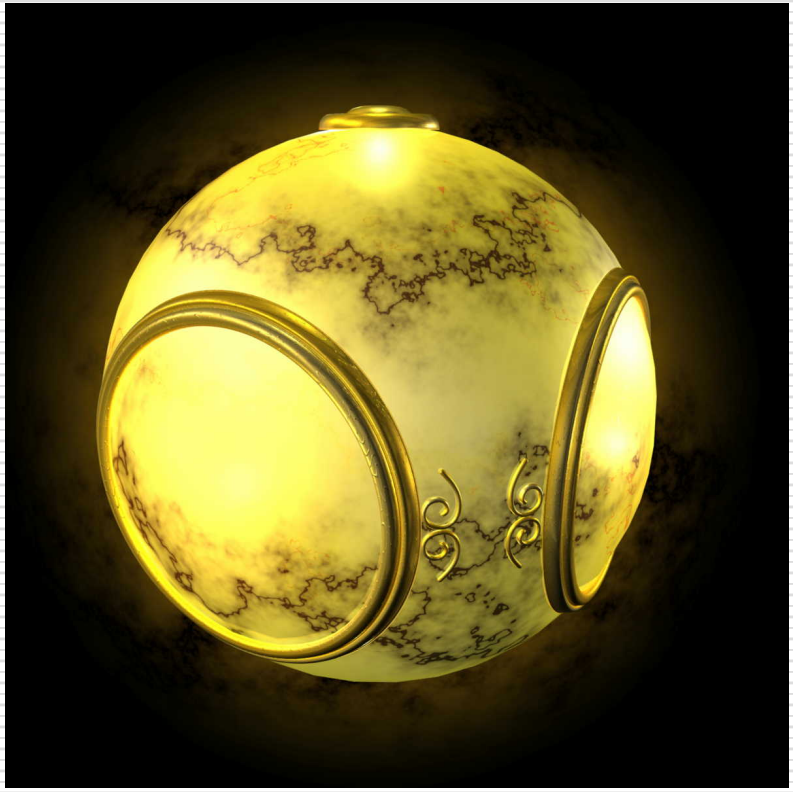
Tom Porter's Bowling Pin



Texture Mapping



geometric model



texture mapped

Texture Maps

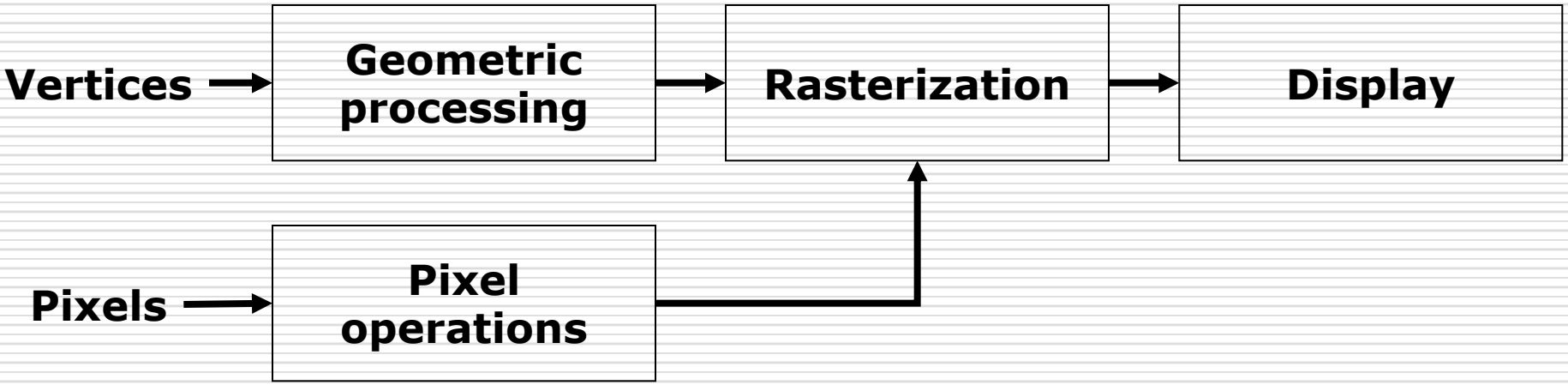
- How is texture mapped to the surface?
  - Dimensionality: 1D, 2D (image), 3D (solid)
  - Procedural v.s. table look-up
  - Texture coordinates (s,t)
    - Surface parameters (u,v)
    - Projection: spherical, cylindrical, planar
    - Reparameterization
- What does texture control?
  - Surface color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

Texture Mapping

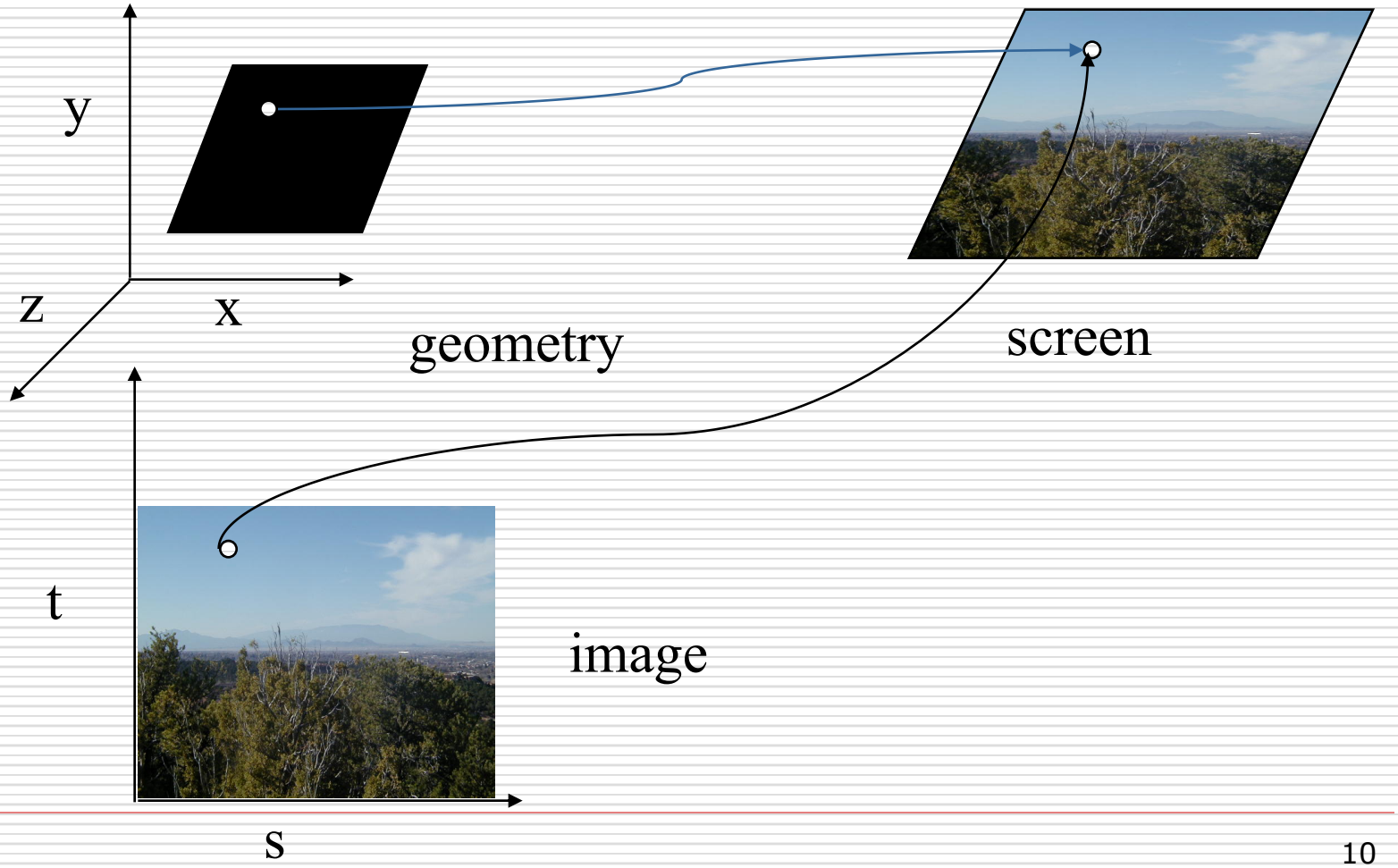
2D mapping **3D mapping**

Where does mapping take place?

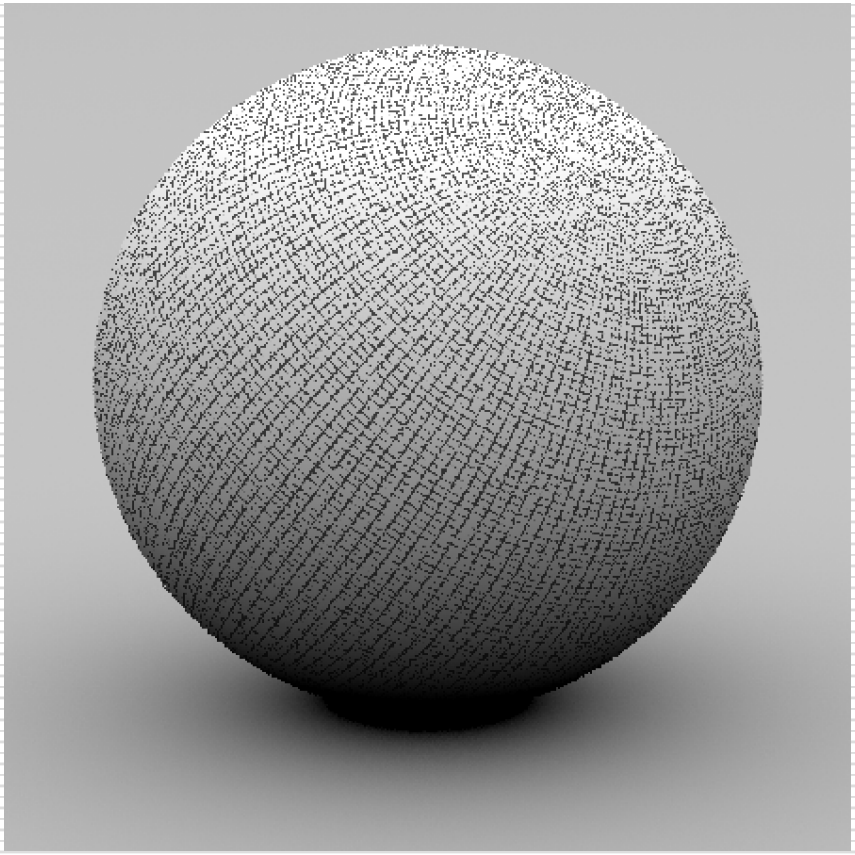
- Mapping techniques are implemented at the end of the rendering pipeline
  - Very efficient because few polygons pass down the geometric pipeline

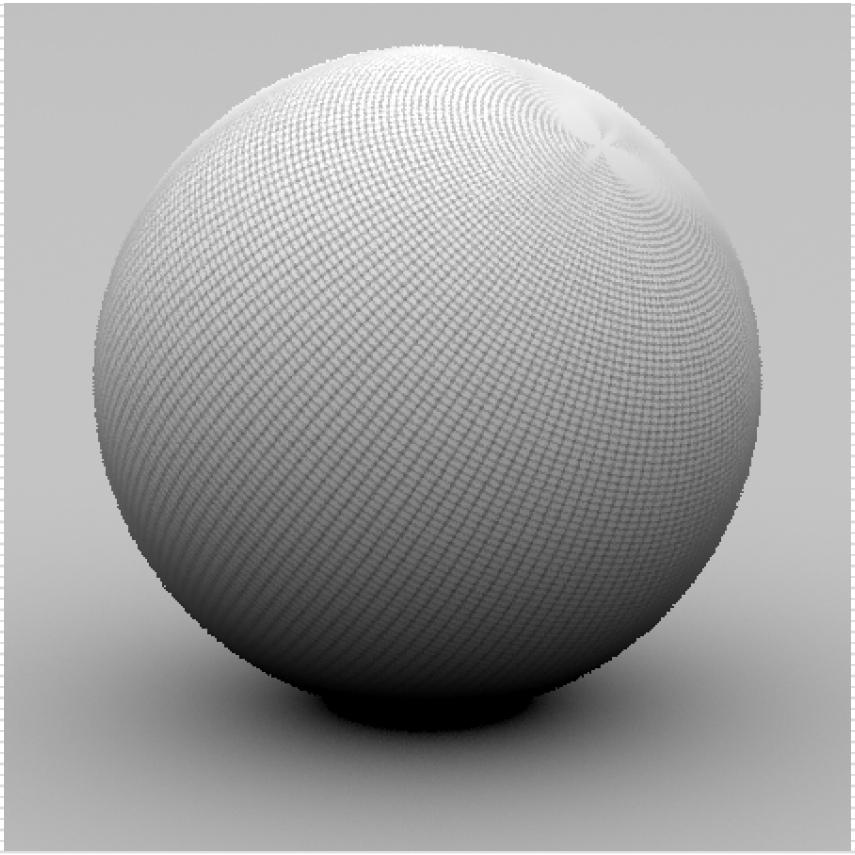


Simple Texture Mapping



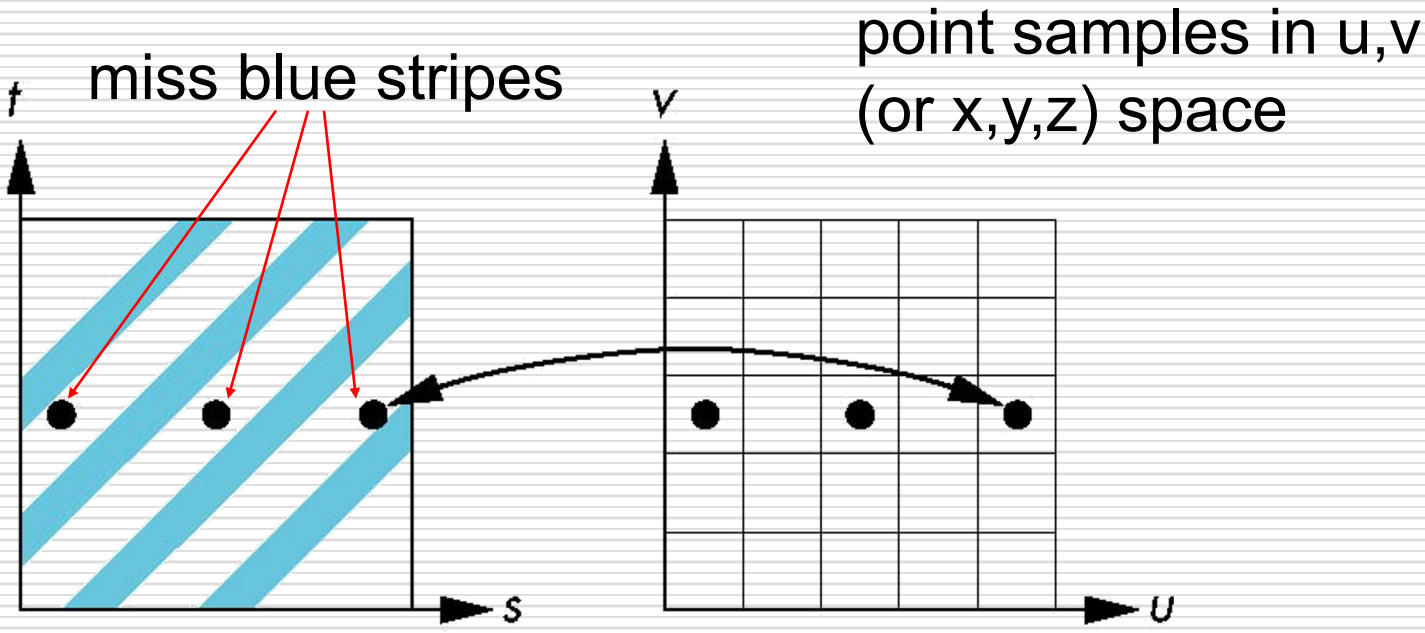
Antialiasing





Aliasing

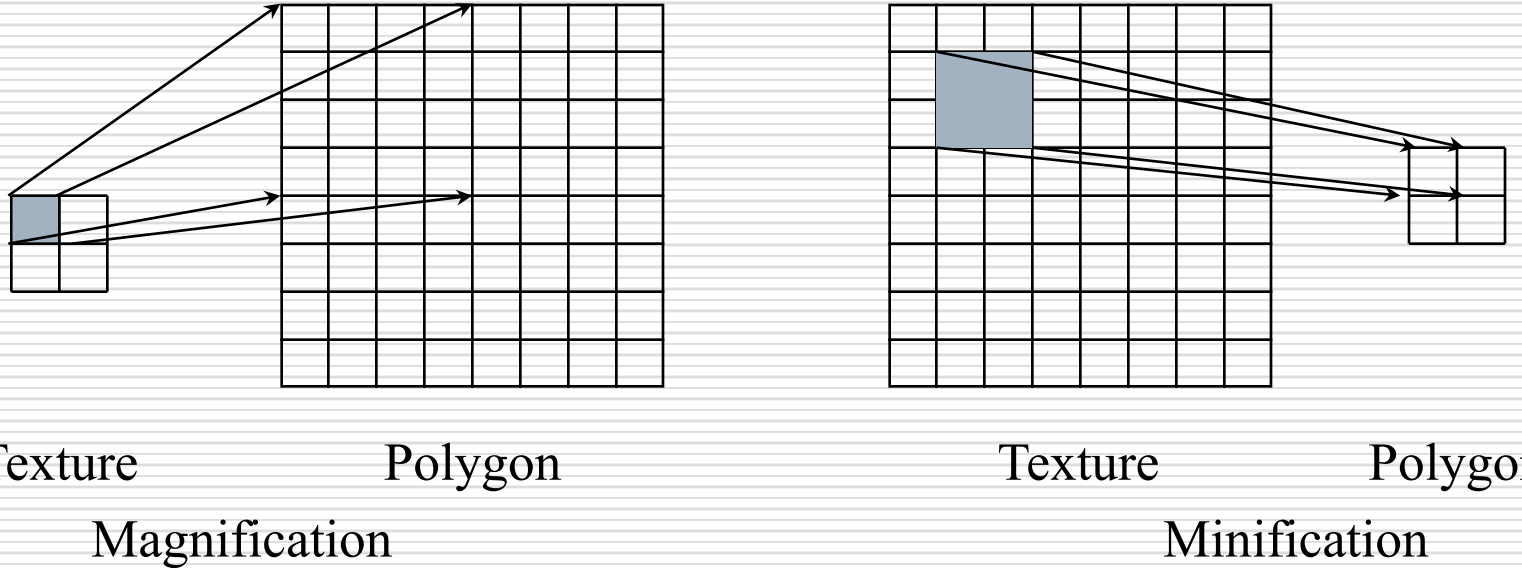
- Point sampling of the texture can lead to aliasing errors



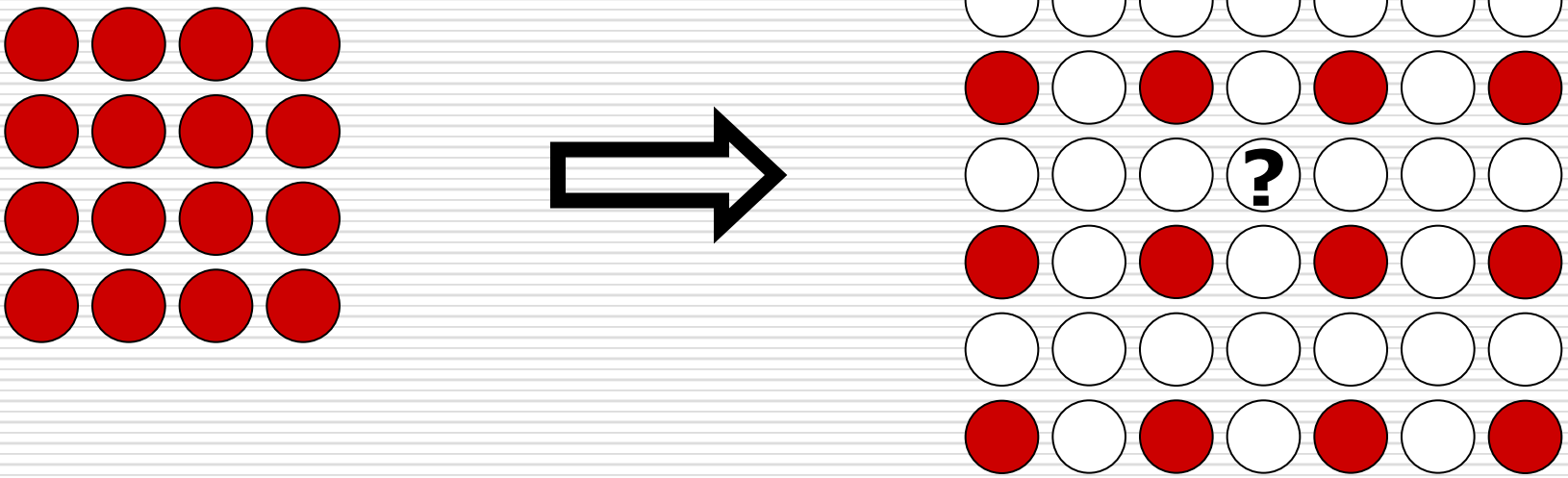
point samples in texture space

Magnification and Minification

- Example:

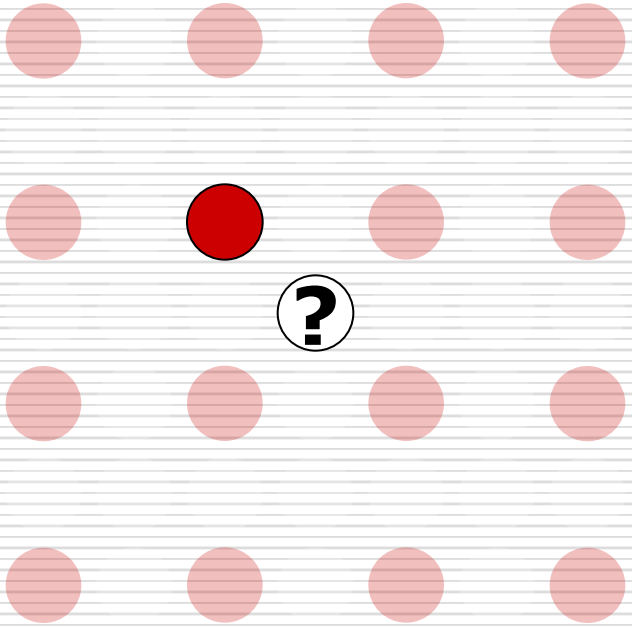


Changing Resolution



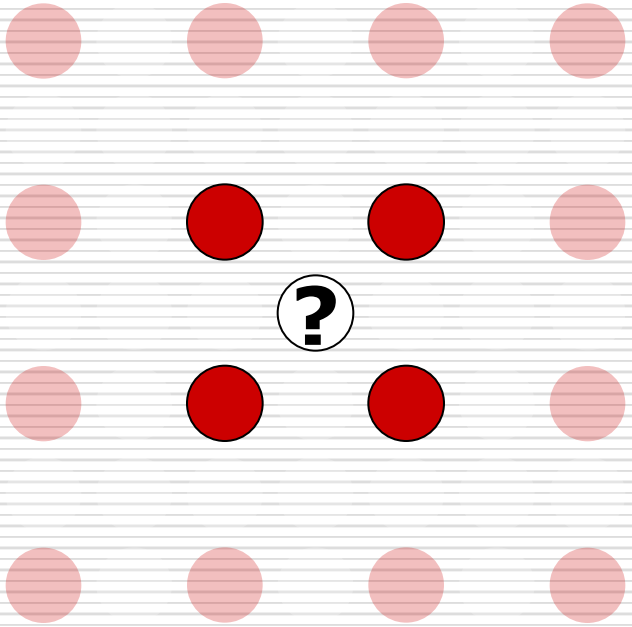
Nearest Neighbor

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



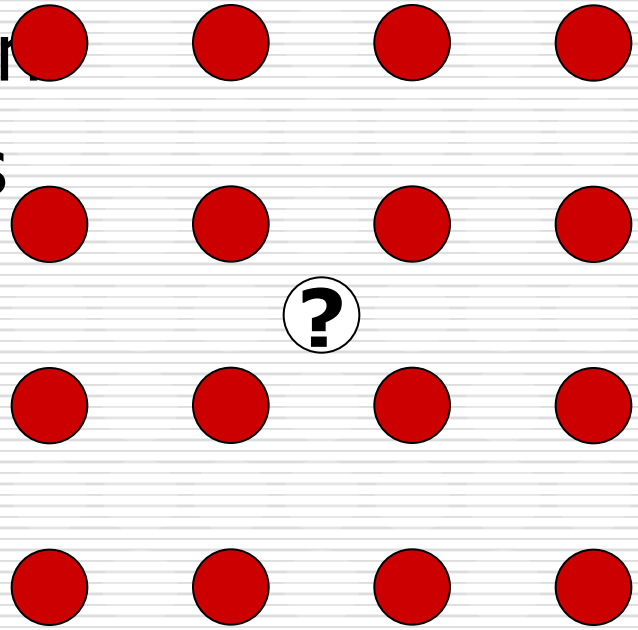
Bilinear

- 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

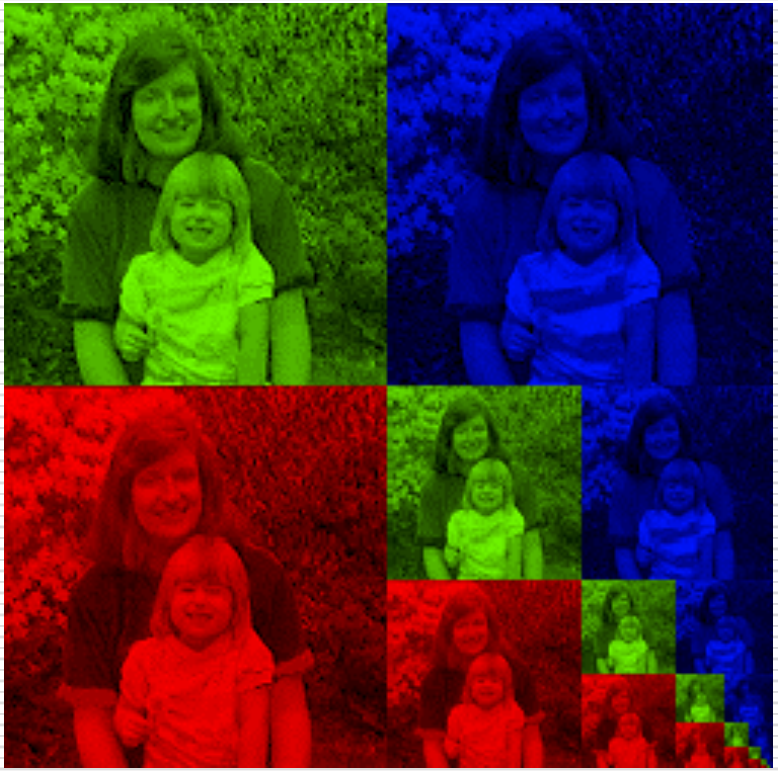

ground truth

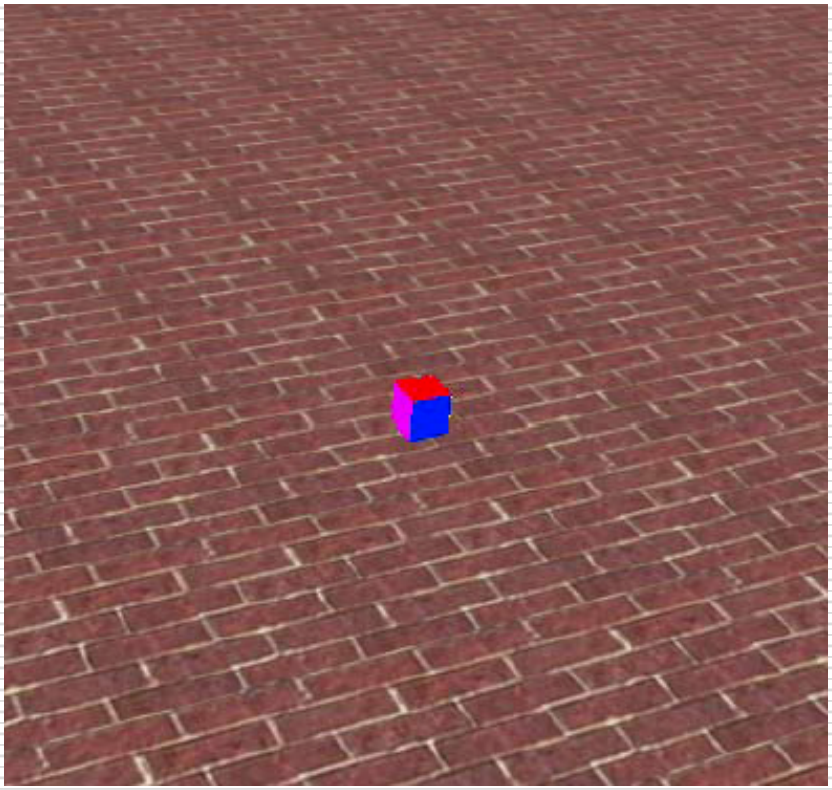
MIP Mapping

- 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

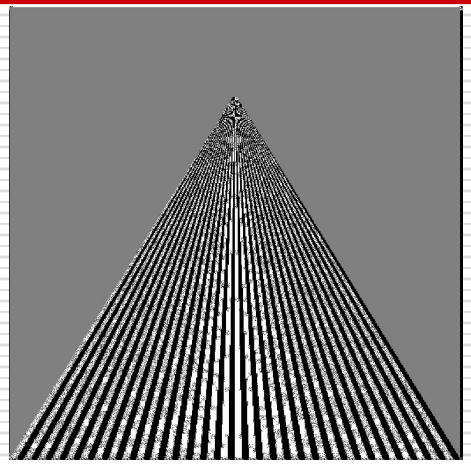
Storing MIP Maps

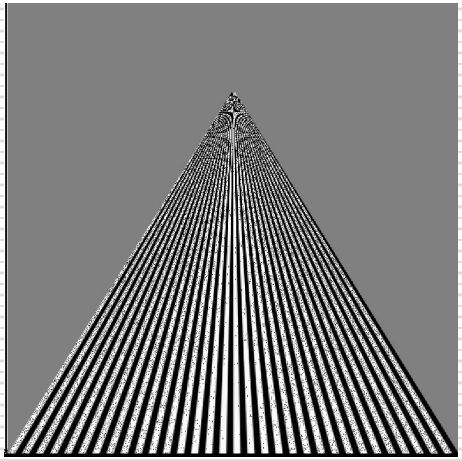




Example

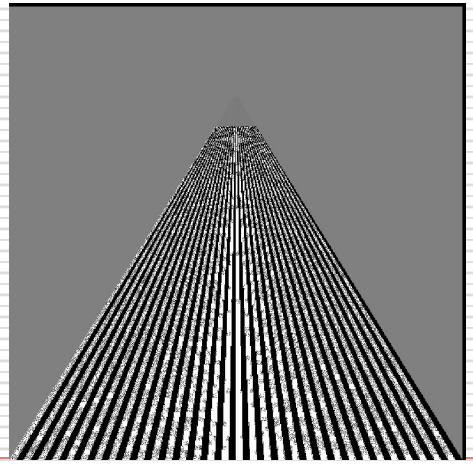
point sampling

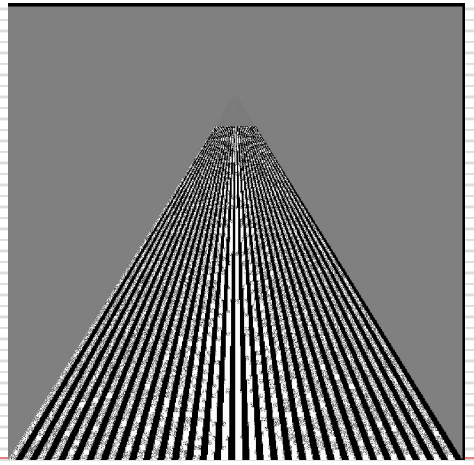




linear filtering

mipmapped point sampling





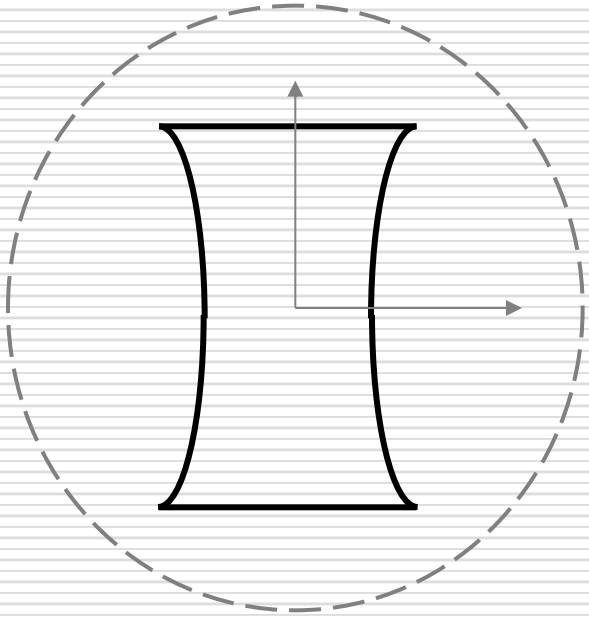
mipmapped linear filtering

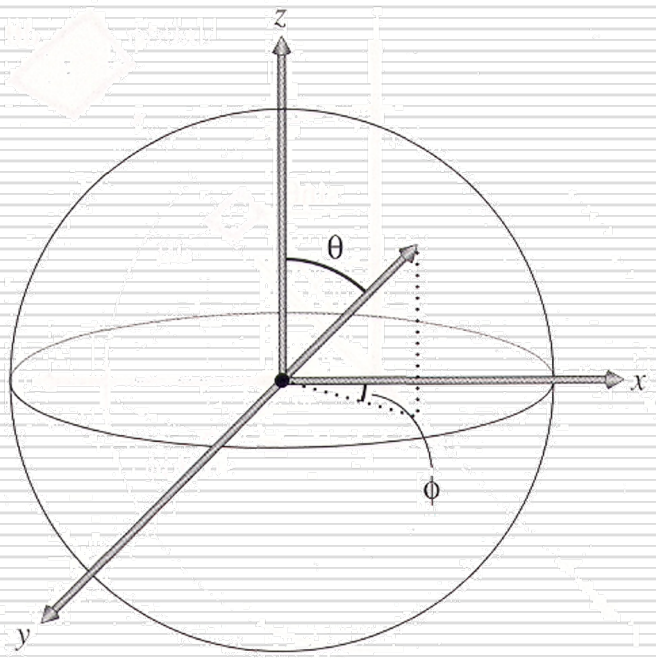
Environment Mapping

Sphere Mapping

Box Maps

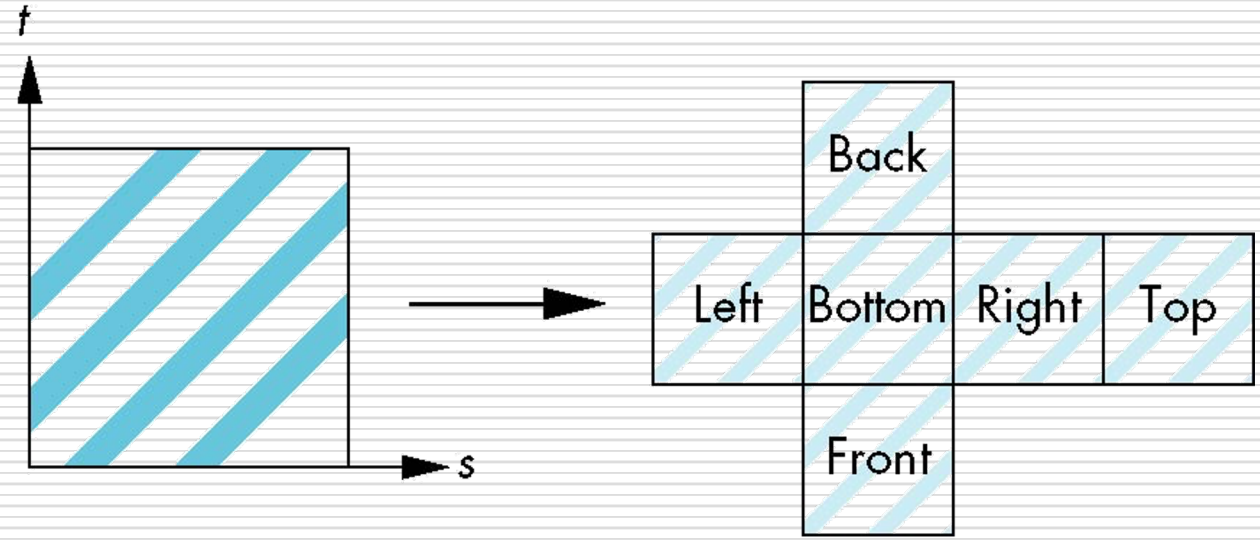
Spherical Mapping





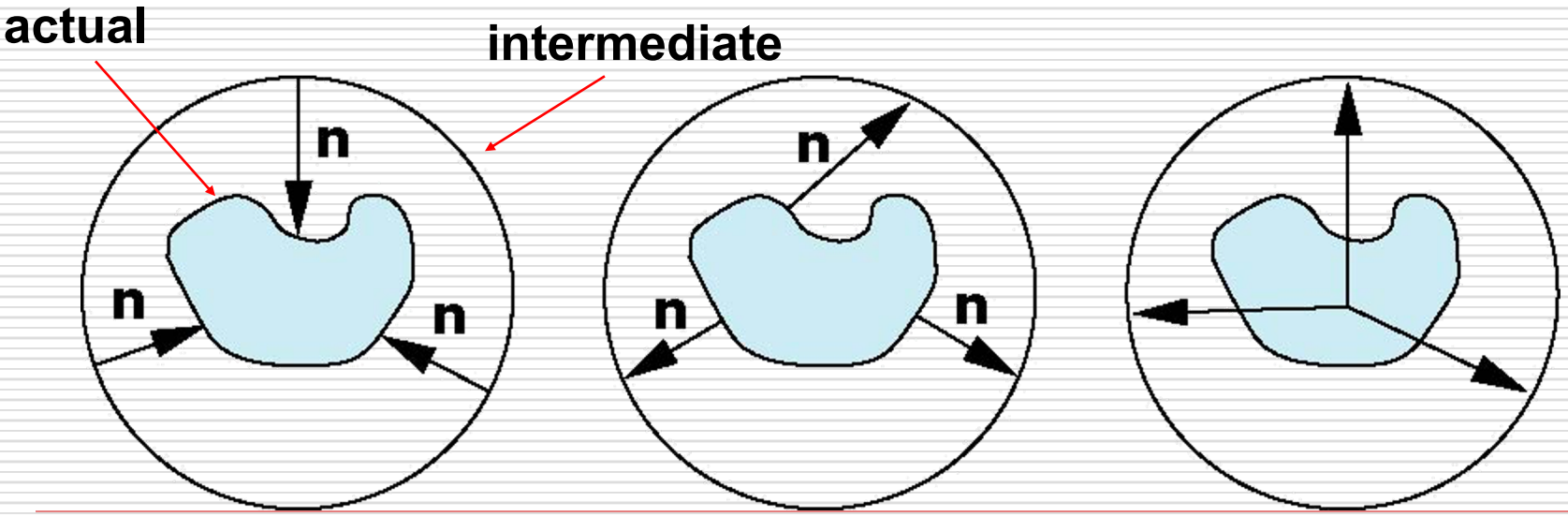
Box Mapping

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



Second Mapping

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



Environment Maps

ray traced

environment map

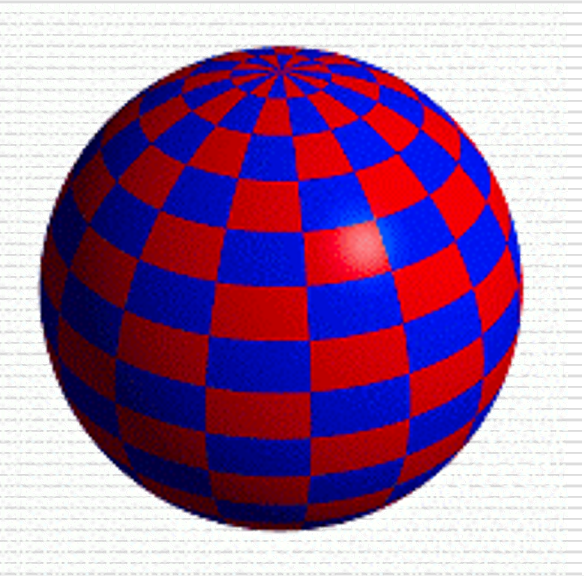
Bump Mapping

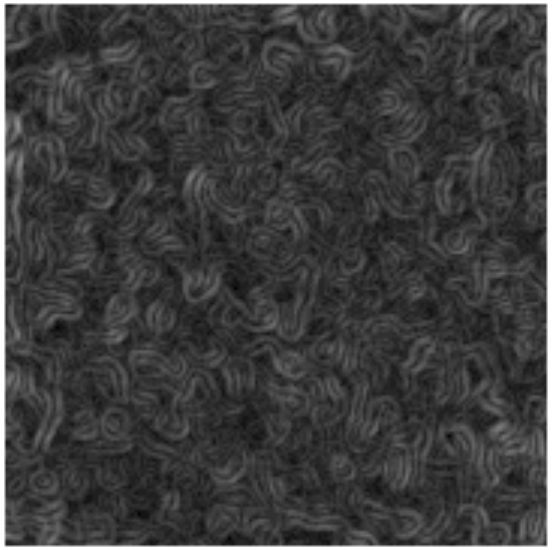
- Textures can be used for more than just color

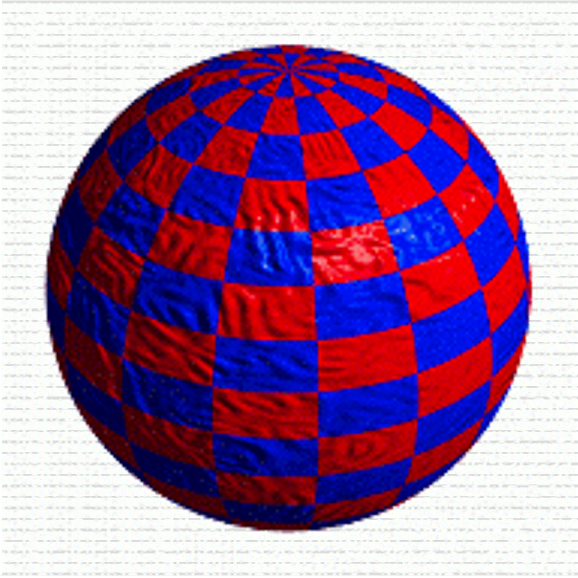
$$I = k_{\mathrm{a}} I_{\mathrm{a}} + \sum_i f_{\mathrm{att}_i} I_{\mathrm{p}_i} [k_{\mathrm{d}} (\vec{N} \bullet \vec{L}_i) + k_{\mathrm{s}} (\vec{R}_i \bullet \vec{V})^n]$$

- 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

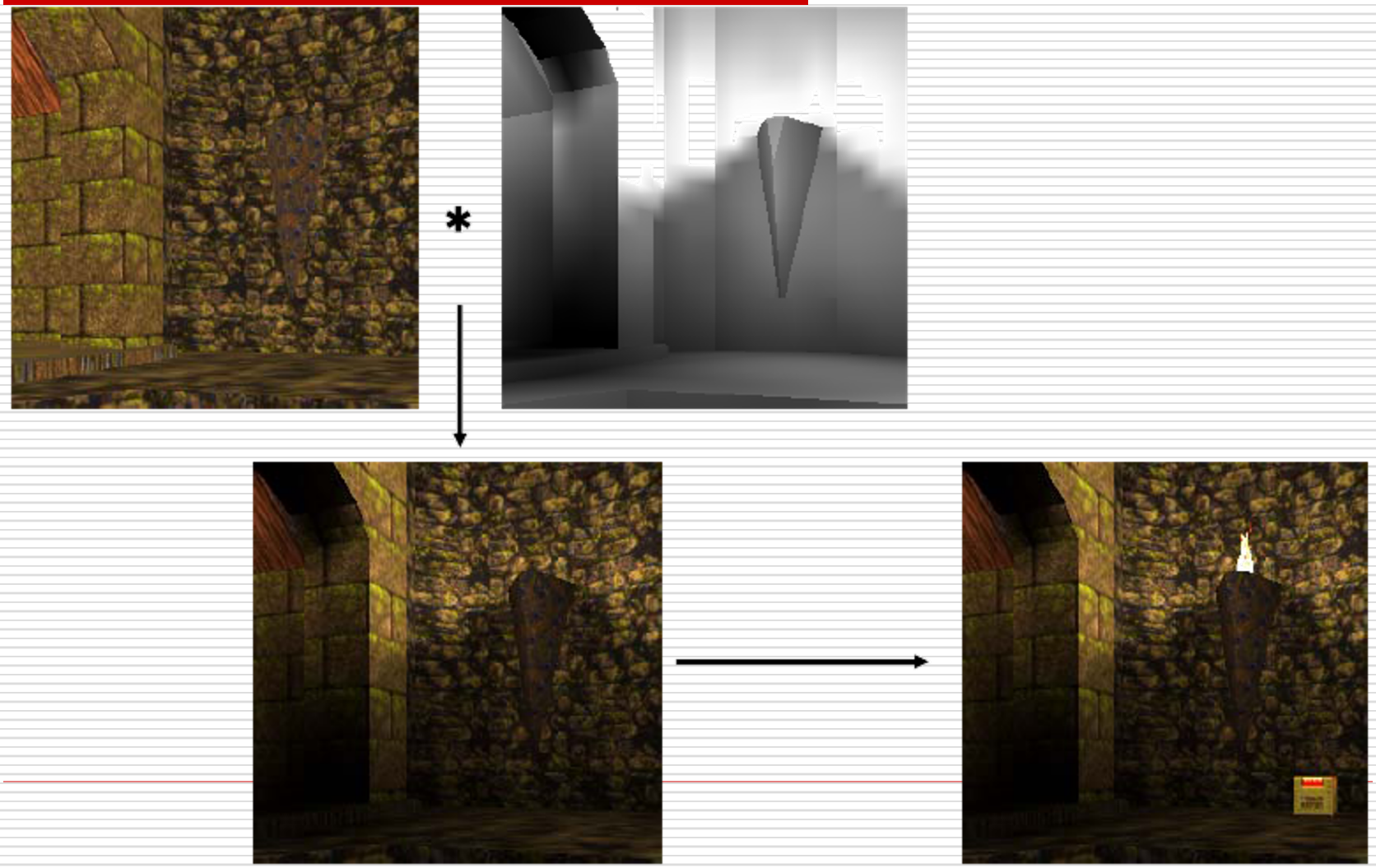




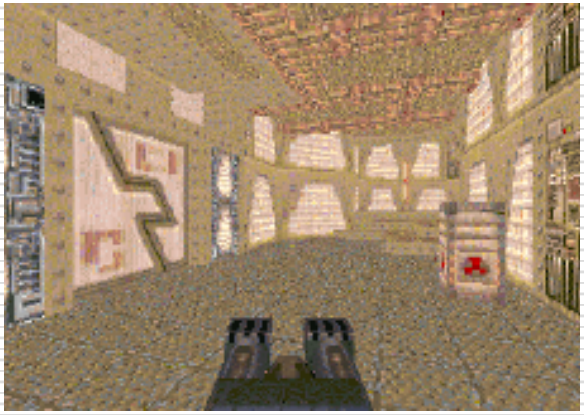


Bump Mapping

Illumination Maps



Texture Mapping in Quake

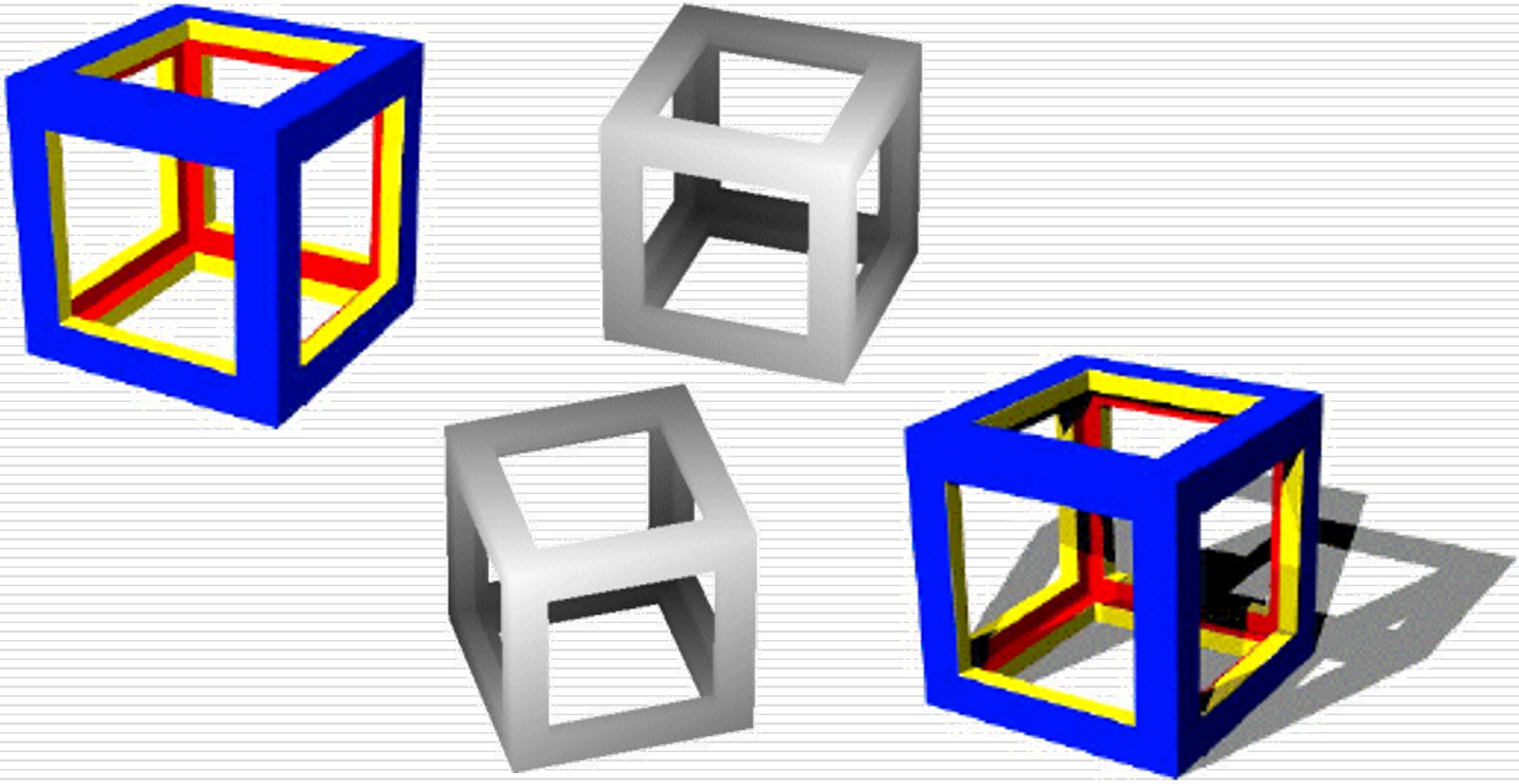


Texture Only

Texture & Light Maps

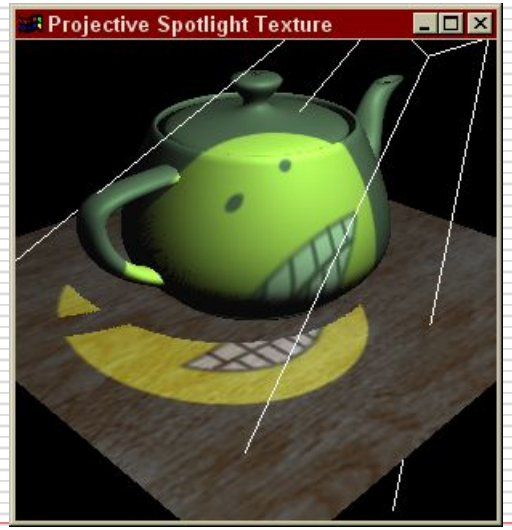
Light Map

Shadow Maps



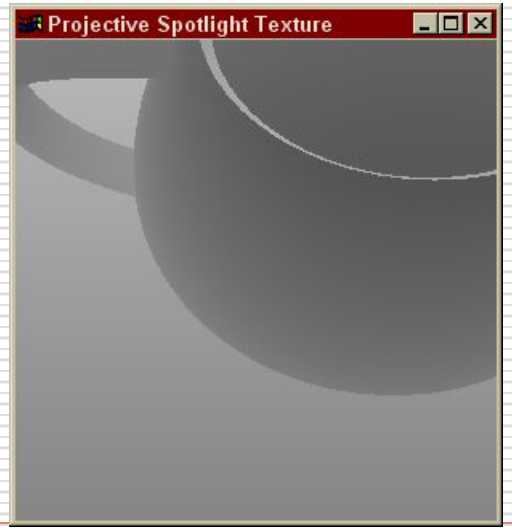
Basic Steps of Shadow Maps

- 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.



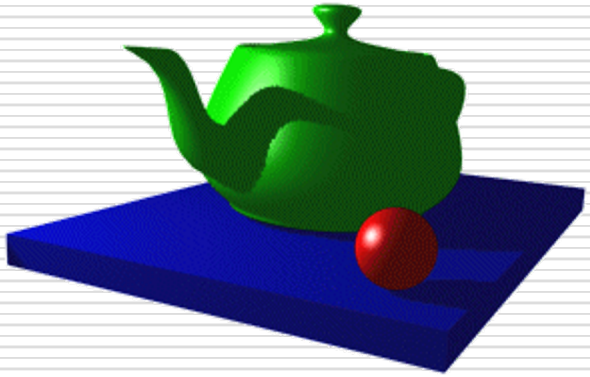
Eye's View

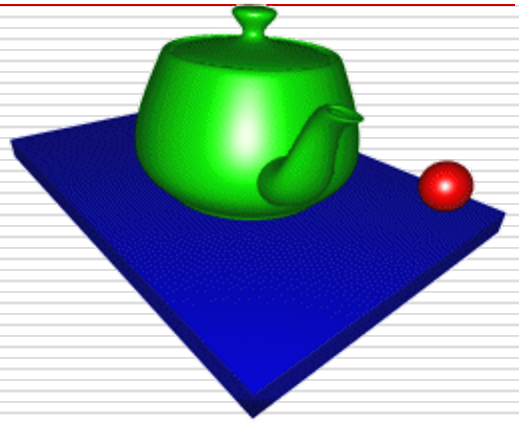
Light's View

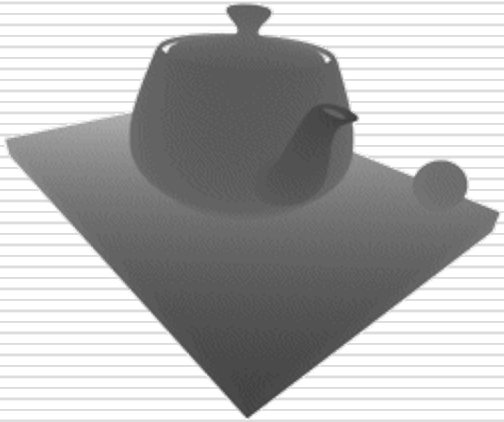


Depth/Shadow Map

Shadow Buffer







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