

# Lecture 15: Texture Mapping

# Reading

- Hearn & Baker 14.8-14.9

## **Optional**

- Paul S. Heckbert. Survey of texture mapping. *IEEE Computer Graphics and Applications* 6(11): 56-67, November 1986

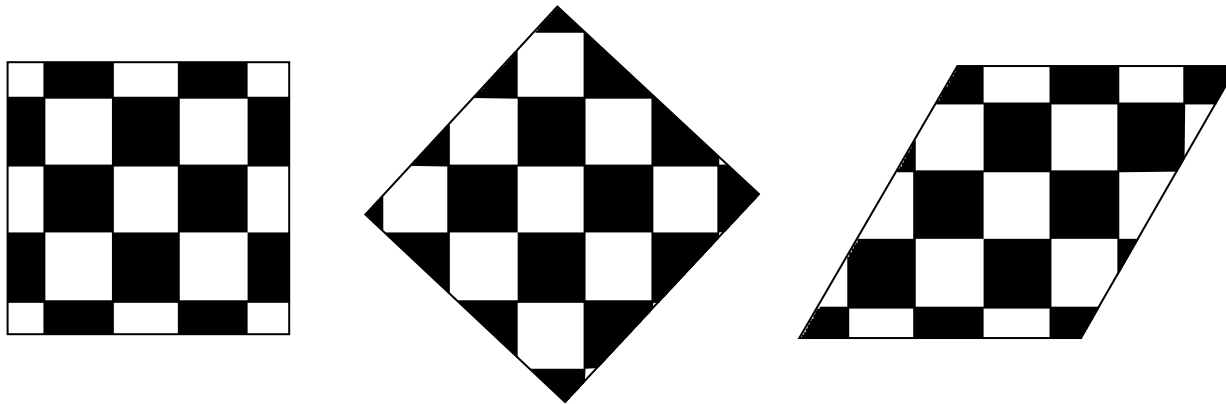
# Texture mapping



Texture mapping allows you to take a simple polygon and give it the appearance of something much more complex

- Due to Ed Catmull, PhD thesis, 1974
- ensures that “all the right things” happen as a texture polygon is transformed and rendered

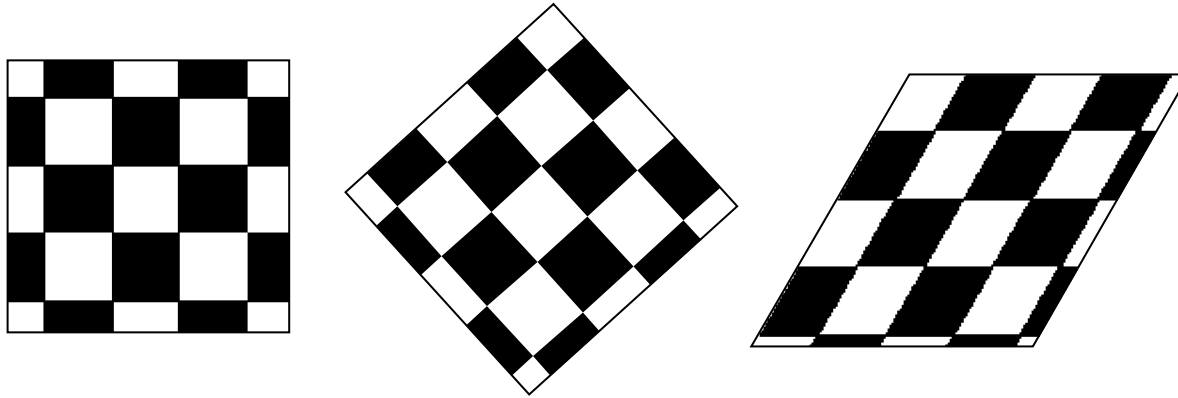
# Non-parametric texture mapping



With non parametric texture mapping:

- Texture size and orientation are fixed
- Unrelated to size and orientation of polygon
- Gives a cookie-cutter effect

# Parametric texture mapping

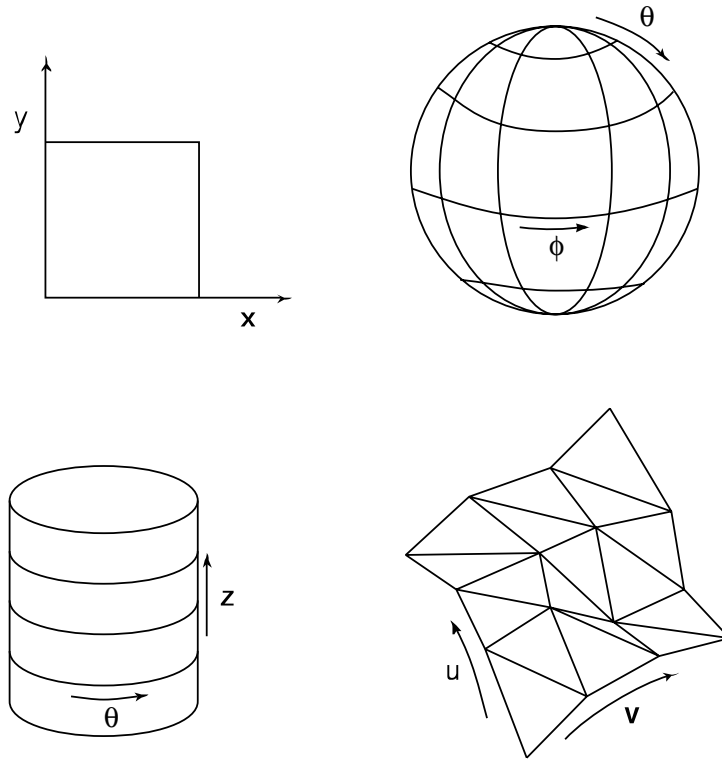


With parametric texture mapping, texture size and orientation are tied to the polygon:

- Separate texture space and screen space
- Texture the polygon as before but in texture space
- Deform (render) the textured polygon into screen space

# Implementing texture mapping

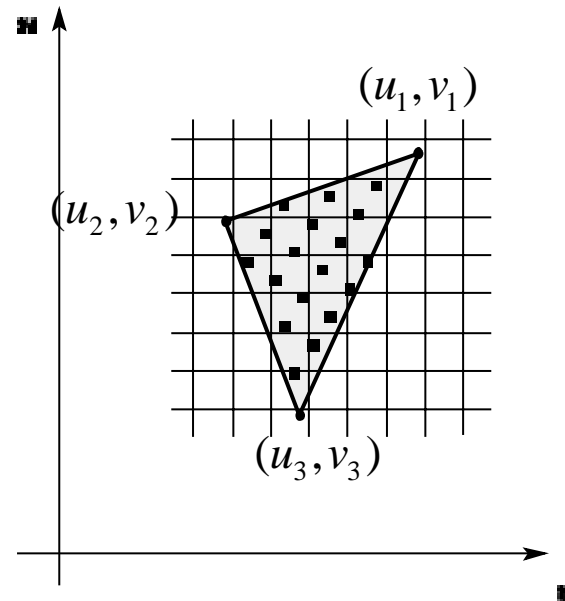
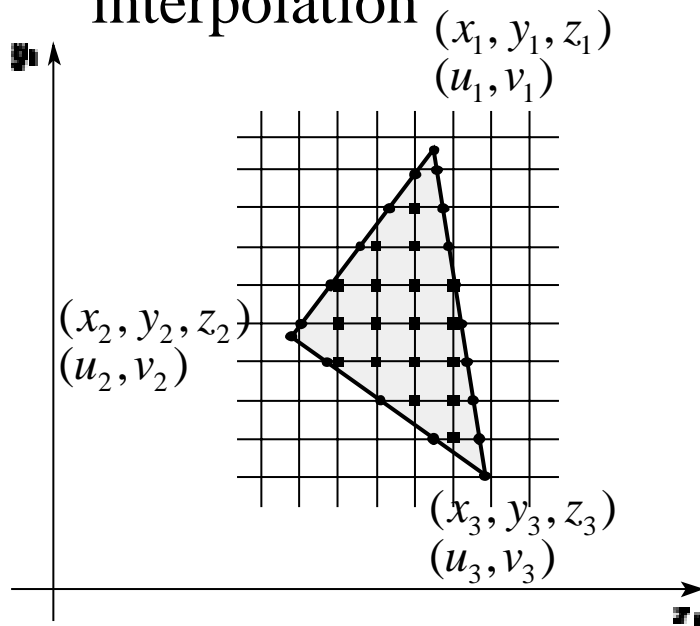
Textures can be warped around many different surfaces:



Computing  $(u,v)$  coordinates in a ray tracer is fairly straightforward

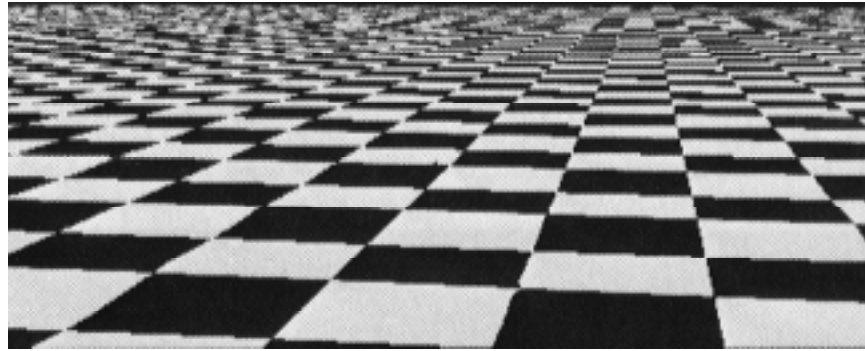
# Implementing, cont.

- Texture mapping can also be handled in z-buffer algorithms
- Scan conversion is done in screen space, as usual
- Each pixel is colored according to the texture
- Texture coordinates are found by Gouraud-style interpolation

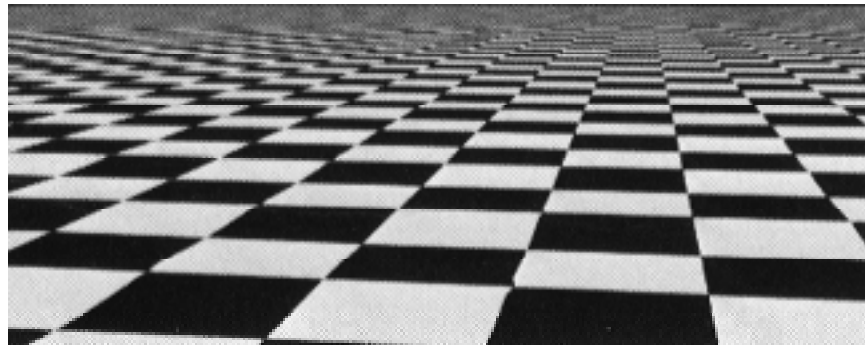


# Antialiasing

- If you point-sample the texture map, you get aliasing:



- Proper antialiasing requires area averaging in the texture:



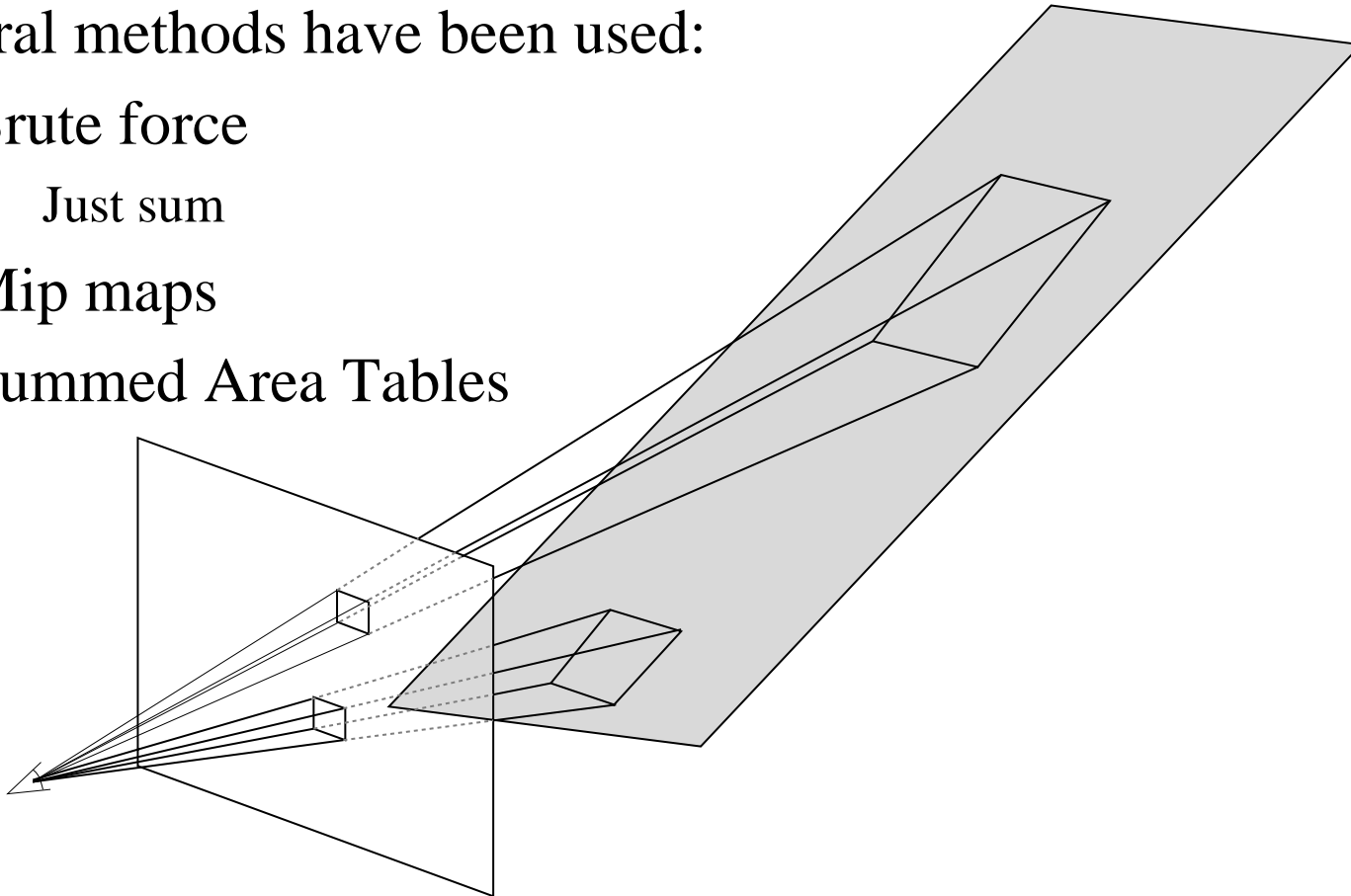


# Computing average color

Computationally difficult part is summing over the covered pixels:

Several methods have been used:

1. Brute force
  - Just sum
2. Mip maps
3. Summed Area Tables

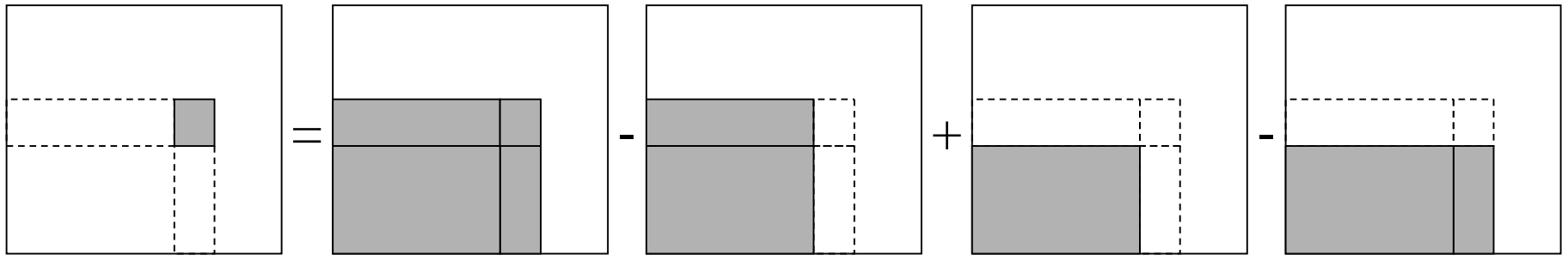


# Mip Maps



- Lance Williams, 1983
- “multum in parvo” – many things in a small place”
- Keep textures prefiltered at multiple resolutions
- Figure out two closest levels
- Linear interpolate between the two

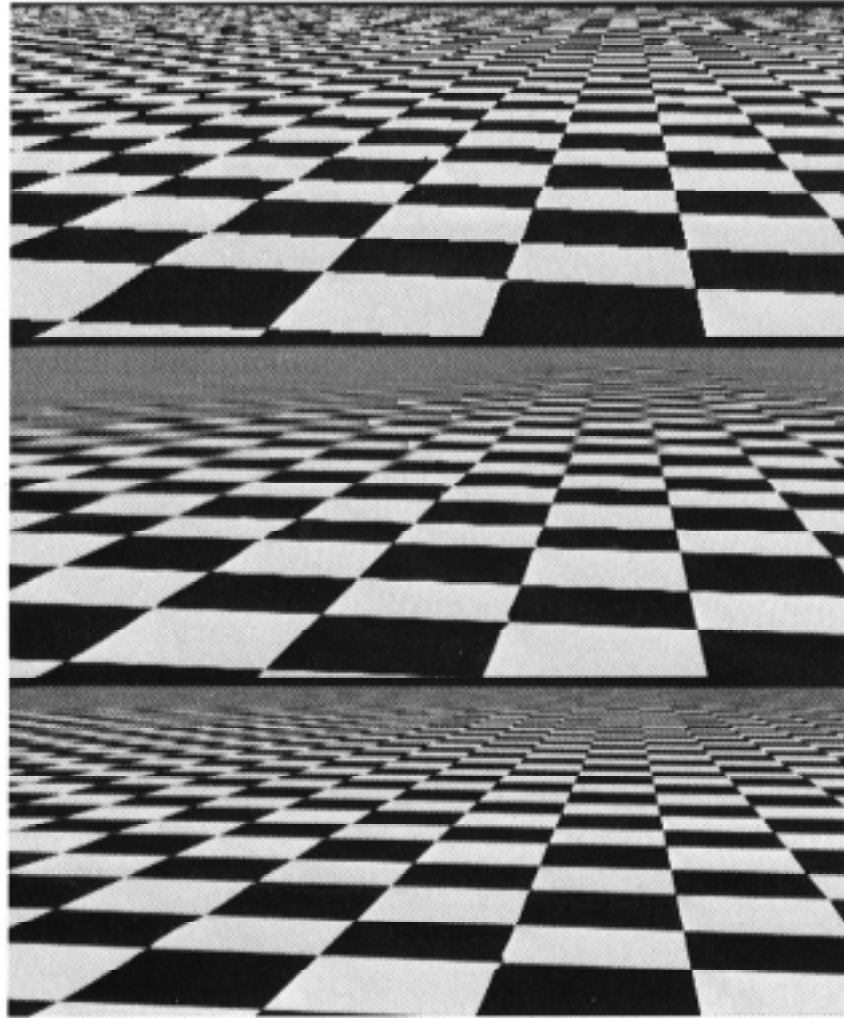
# Computing average color, cont.



## 3. Summed area tables

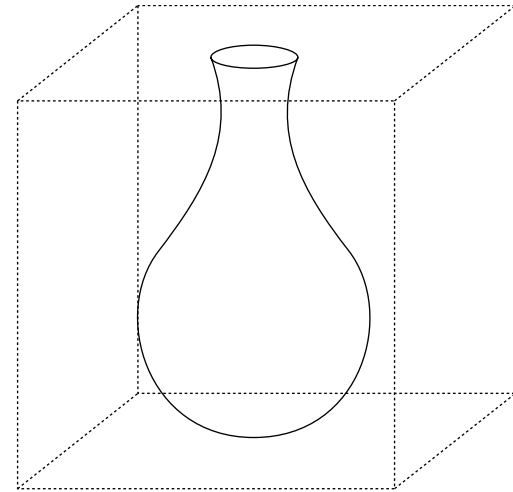
- Frank Crow, 1984
- Keep sum of everything below and to the left
- Use four table lookups
- Requires more memory (2-4 times the original image)
- Gives less blurry textures

# Comparison of techniques



# Solid textures

Q: what kinds of artifacts might you see from using a marble veneer instead of a real marble?



- One solution is to use solid textures
- Use model-space coordinates to index into a 3D texture
- Like “carving” the object from the material

One difficulty of solid texturing is coming up with the textures...

# Procedural texture mapping

Instead of using texture coordinates to index into an image, use them to compute a function that defines the texture

# Solid textures, cont.

Vase cut from a solid marble texture:



# Bump mapping

Textures can be used for more than just color

$$I = k_a I_a + \sum_i f(d_i) I_{li} \left( k_d (\mathbf{N} \cdot \mathbf{L}_i)_+ + k_s (\mathbf{V} \cdot \mathbf{R})_+^{n_s} \right)$$

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



- These bumps “animate” with the surface

**Q:** What artifacts in the images would reveal that bump mapping is fake?



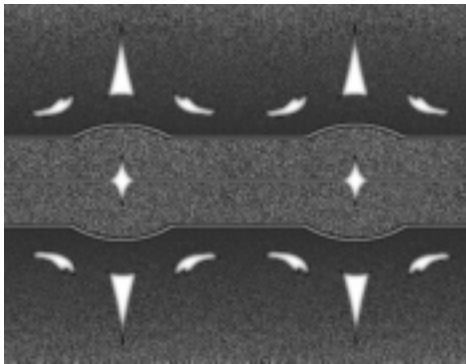
# Displacement mapping

In displacement mapping, a texture is used to perturb the surface geometry itself:

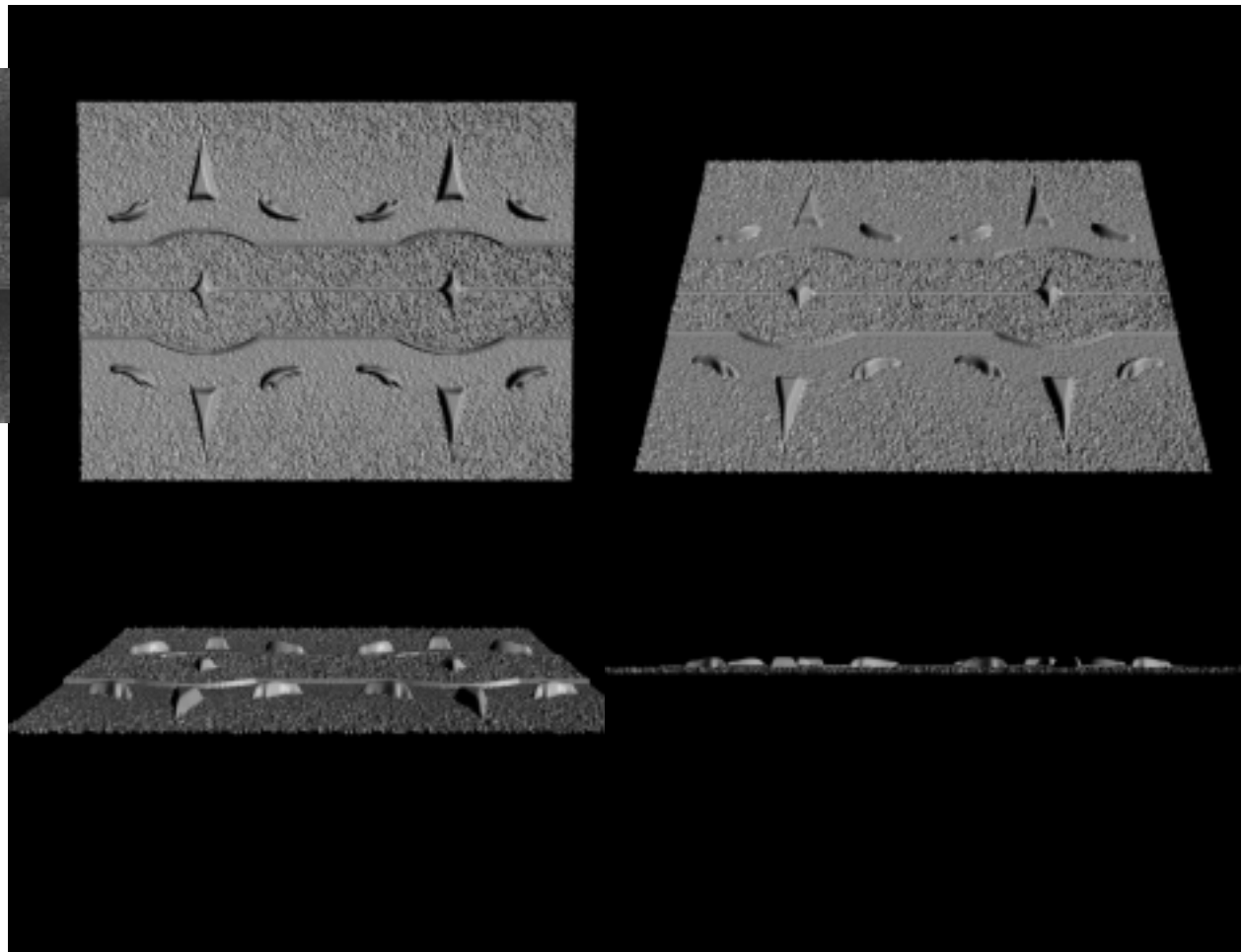
- Silhouettes are correct
- Requires doing additional hidden surface calculations

# Displacement mapping, cont.

Input texture:



Displacement map over rectangular surface:



# Bump mapping example

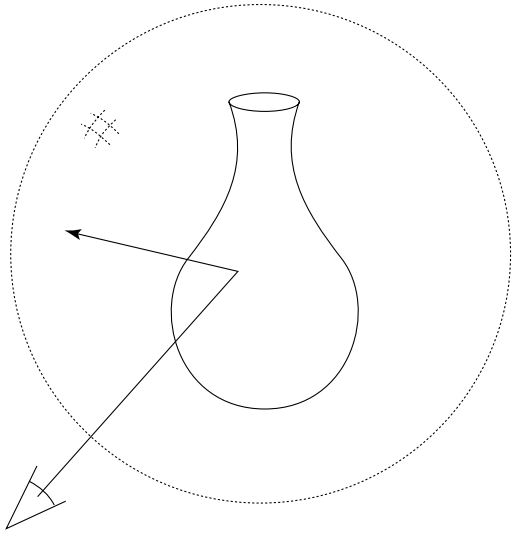


Original rendering



Rendering with bump map wrapped around a cylinder

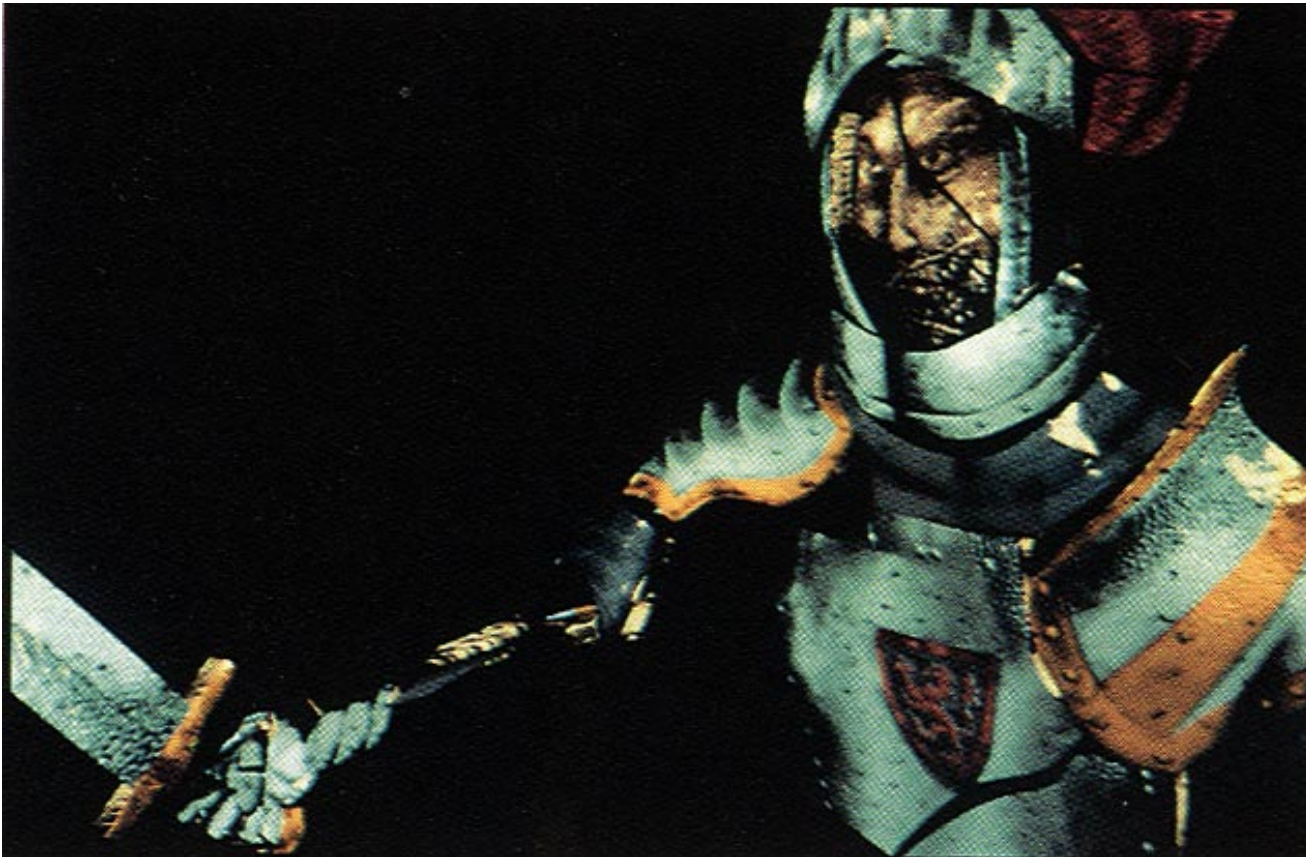
# Environment mapping



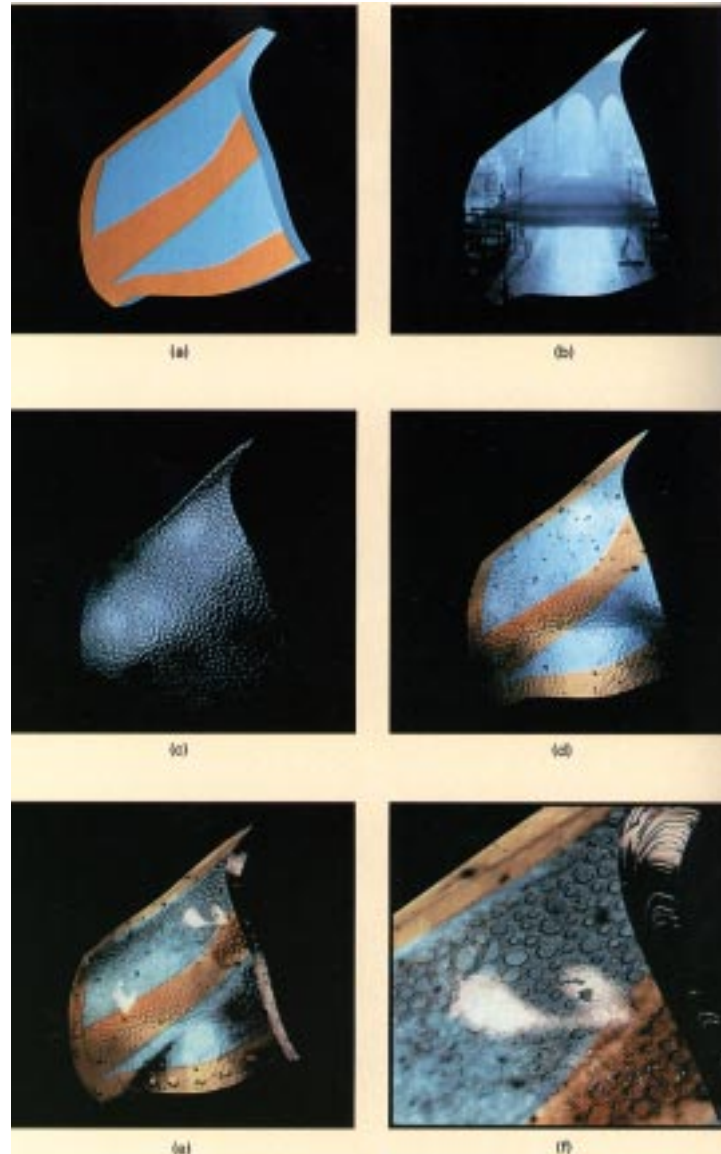
- A.k.a. reflection mapping
- Use texture to model object's environment
- Rays are bounced off objects into environment to determine color of illumination
- Works well when there is just a single object
- With some simplifications can be implemented in hardware
- Raytracer can be extended to handle refractions as well

# Combining texture maps

- Using texture maps in combination give even better effects



# Combining texture maps, cont.



# Summary

What to take from this lecture:

- What texture mapping is and what is it good for
- Understanding the various approaches to antialiased textured mapping
  - Brute force
  - Mip maps
  - Summed area tables
- Additional effect with texture mapping techniques
  - Bump mapping
  - Displacement mapping
  - Environment mapping