

---

7M836

# Animation & Rendering

Mapping, aliasing

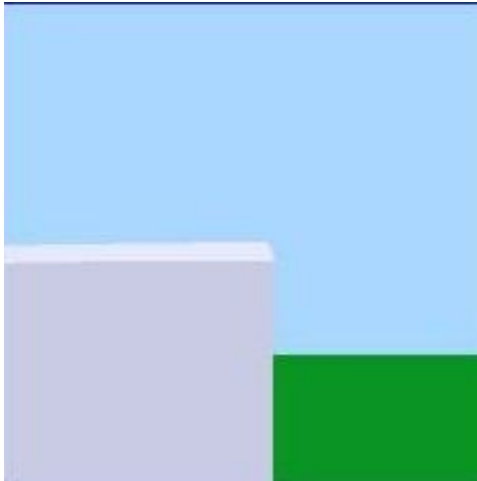
Arjan Kok, Kees Huizing, Huub van de Wetering

[h.v.d.wetering@tue.nl](mailto:h.v.d.wetering@tue.nl)

# Mapping

---

- Why?
  - More detail
  - More realism
  - Simplified modeling



# Mapping

---

- More detail by
  - Variation of diffuse and specular reflection (color), and transparency (texture mapping)
  - Variation of normals (bump mapping)
  - Variation of heights (displacement mapping)
- Mapping techniques can also be used for
  - Simulation mirroring (environment mapping)
  - Shadows (shadow mapping)

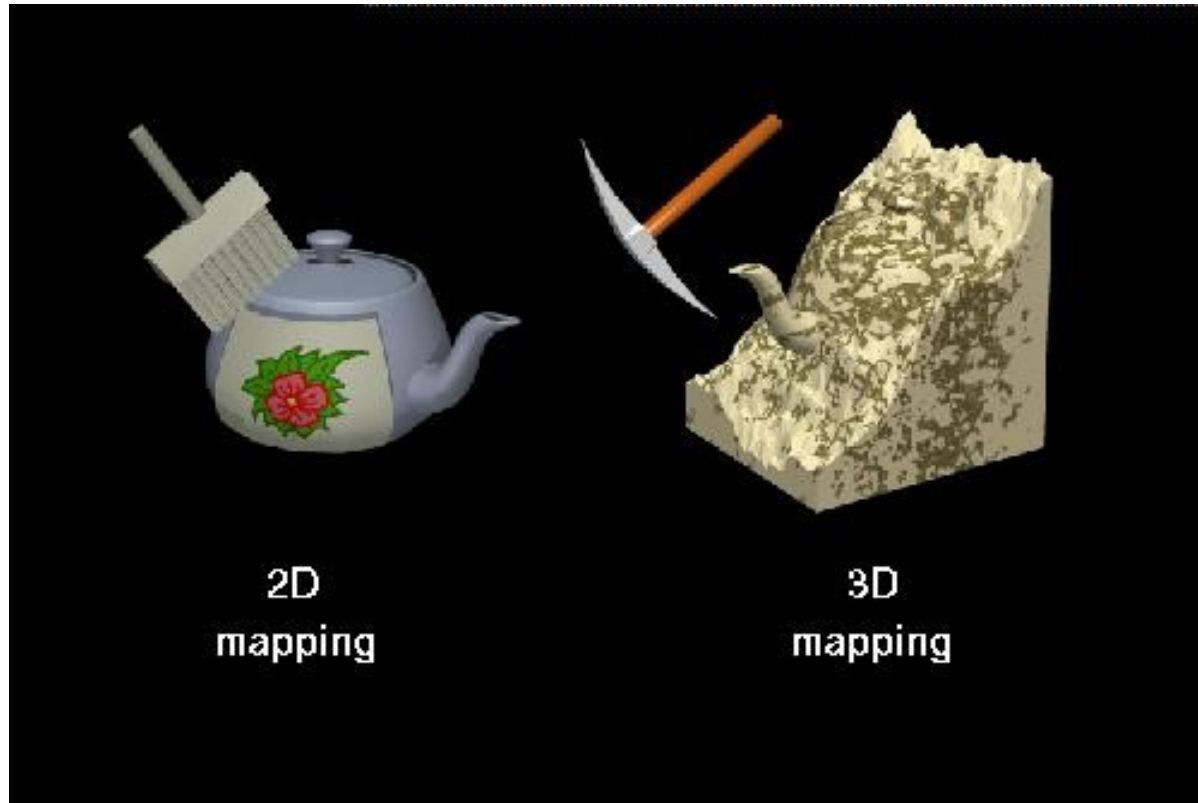
# Texture mapping

---

- Texture mapping is the process that modifies appearance of a point on a surface
- Appearance of point determined by
  - Image
  - Function
  - Dataset

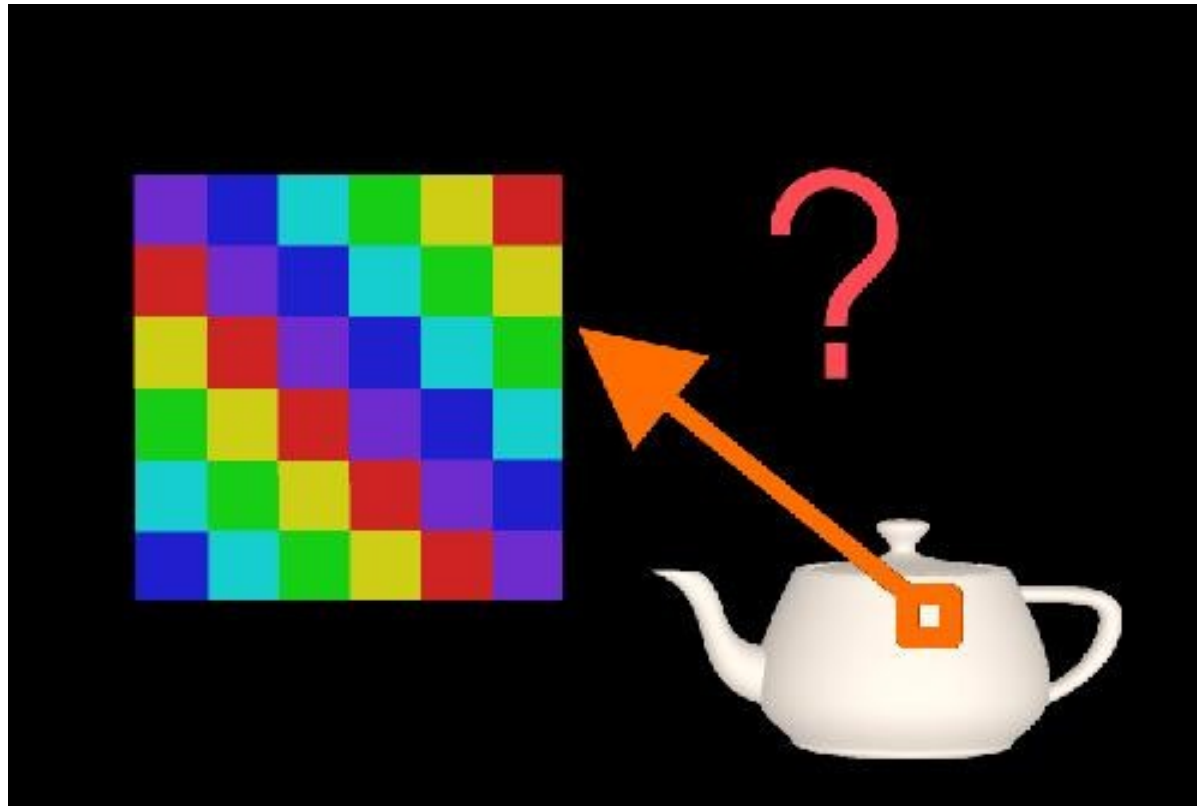
# 2D and 3D mapping

---



# Texture mapping

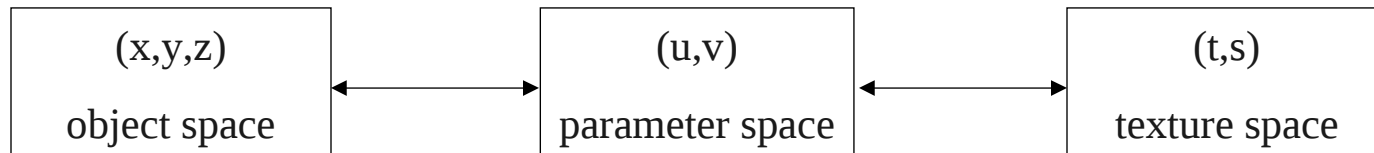
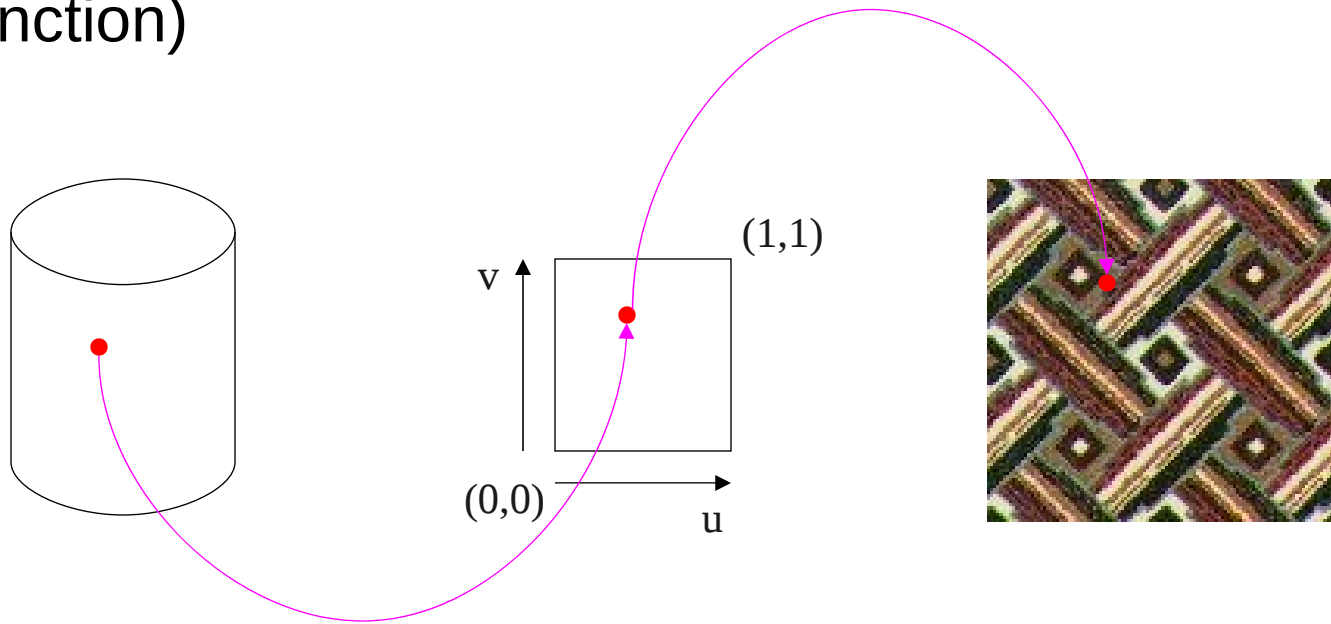
---



# 2D mapping function

---

- Links a 3D object with a 2D texture (image or function)



# $(x,y,z)$ to $(u,v)$

---

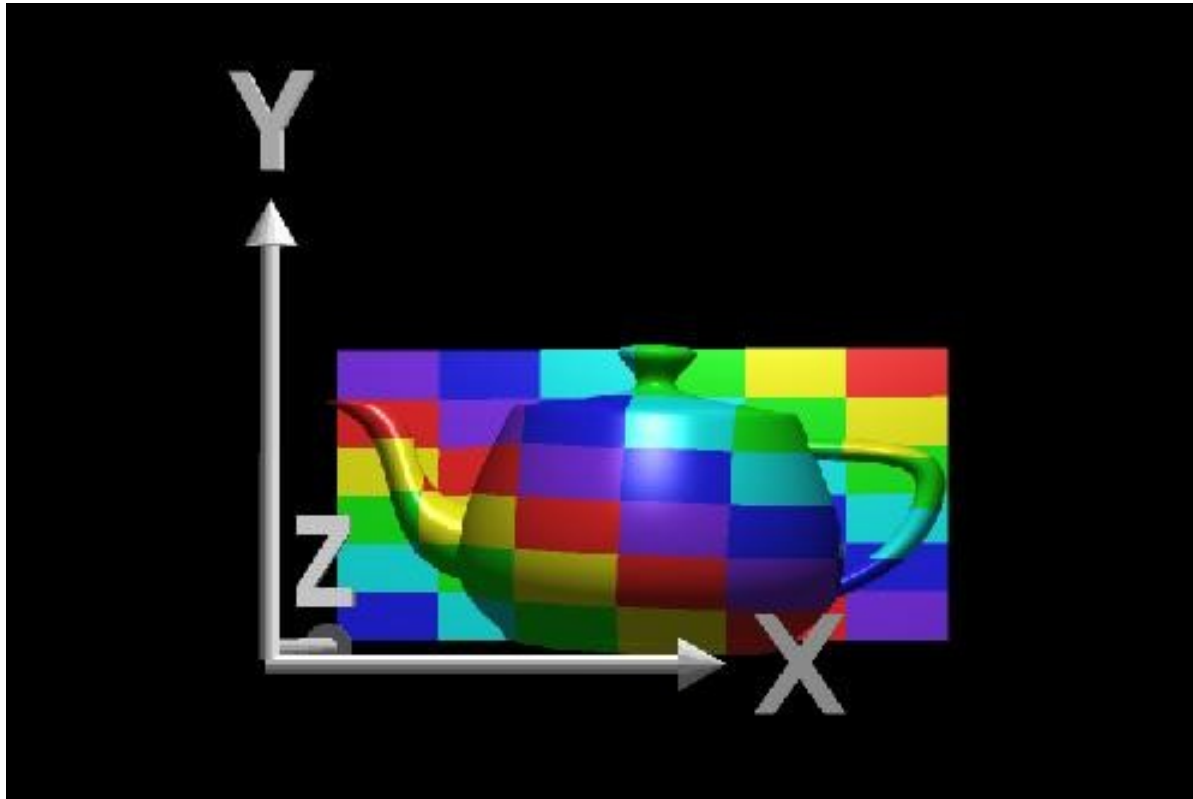
- Use “standard” mappings
  - Planar, cylindrical, sphere, box, ...
- Some geometric descriptions contain “natural “ uv-values E.g. spline surfaces
- Define your own uv-parameterization over object



# Planar mapping

---

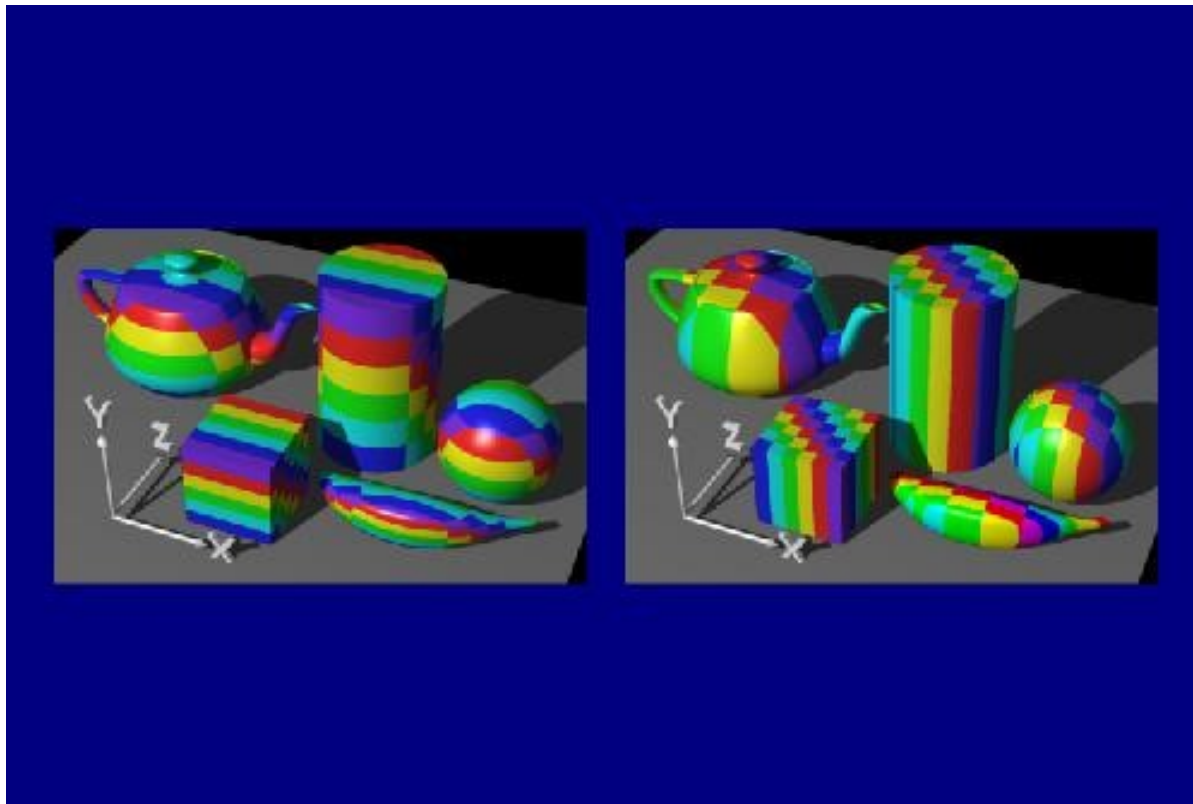
- $(x, y, z)$  to  $(u, v)$



# Planar mapping

---

- $(x, y, z)$  to  $(u, v)$



# Cylindrical mapping

---

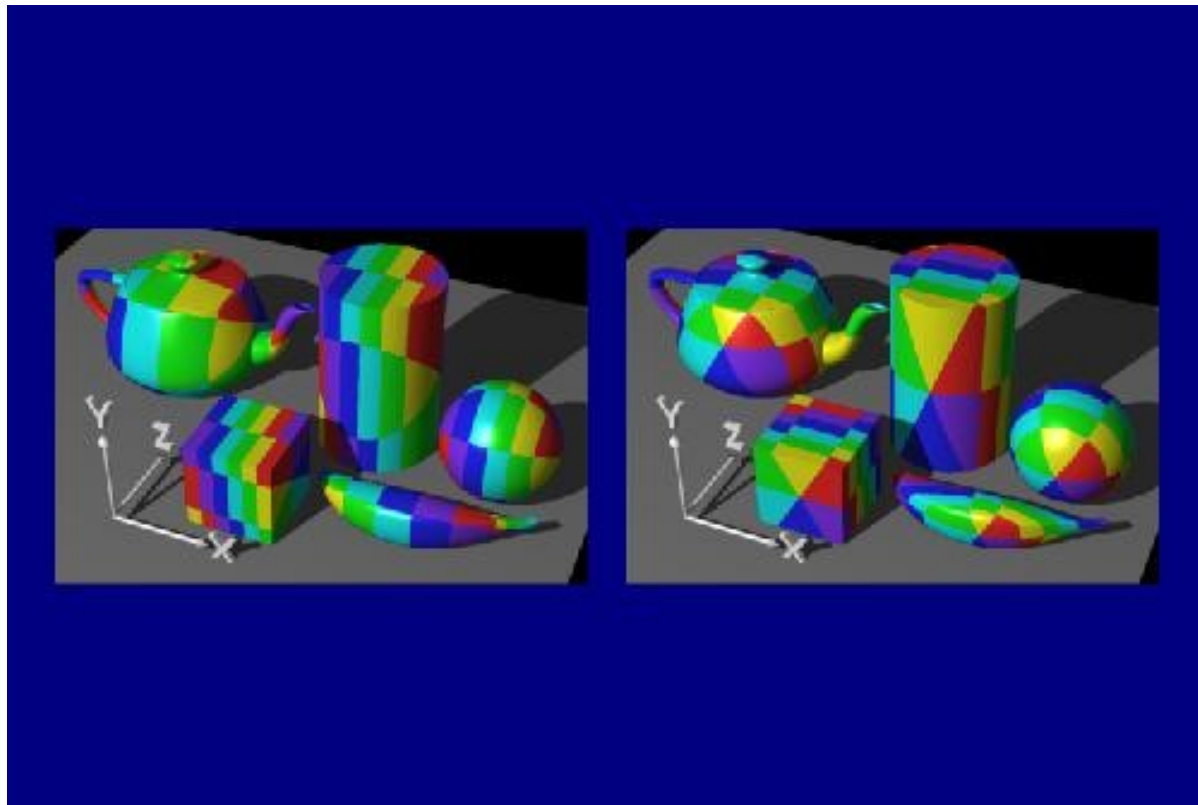
- $(x, y, z)$  to  $(r, \theta, h)$  to  $(u, v)$



# Cylindrical mapping

---

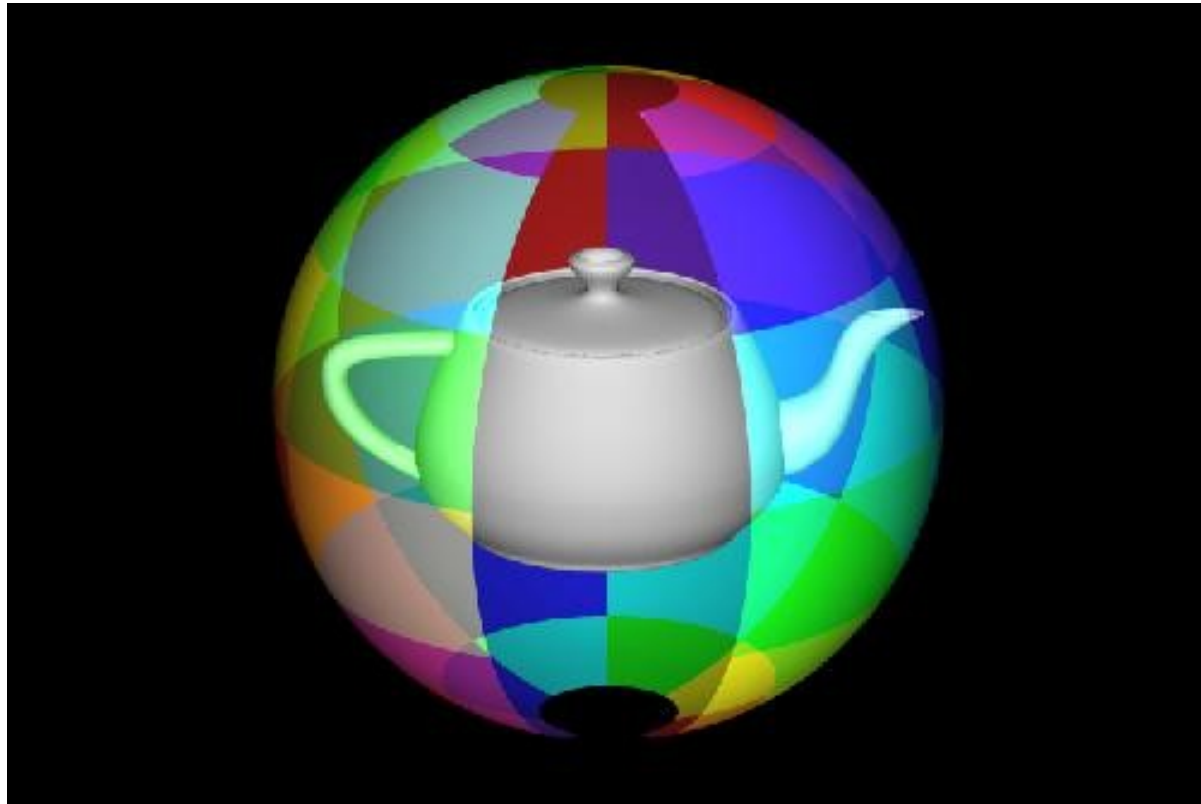
- $(x, y, z)$  to  $(r, \theta, h)$  to  $(u, v)$



# Spherical mapping

---

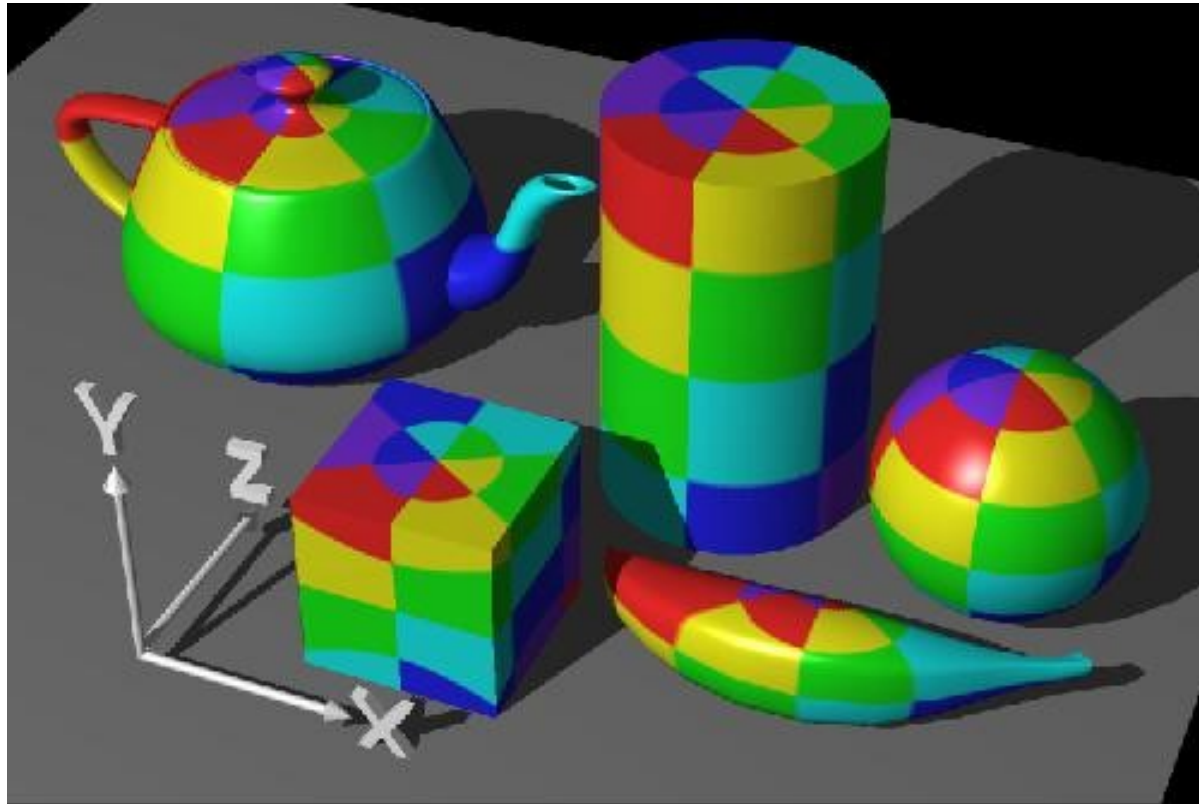
- $(x, y, z)$  to  $(r, \theta, \varphi)$  to  $(u, v)$



# Spherical mapping

---

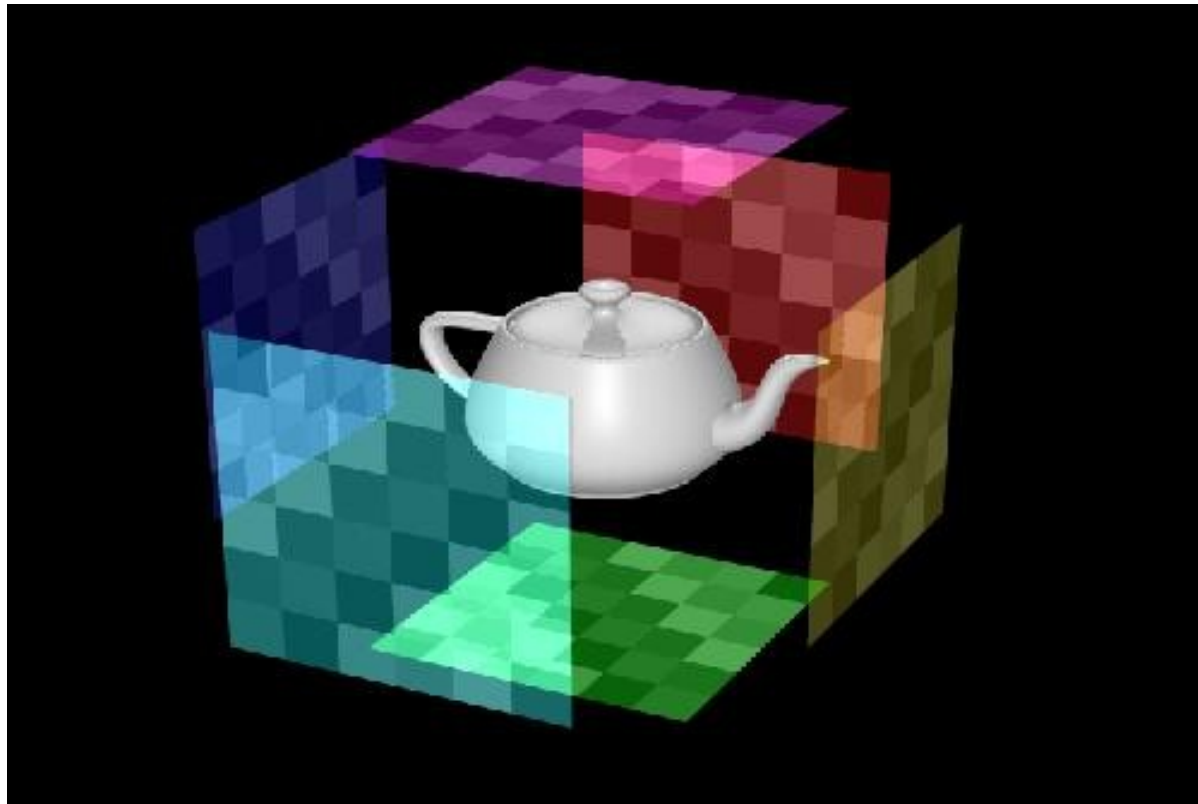
- $(x, y, z)$  to  $(r, \theta, \phi)$  to  $(u, v)$



# Box mapping

---

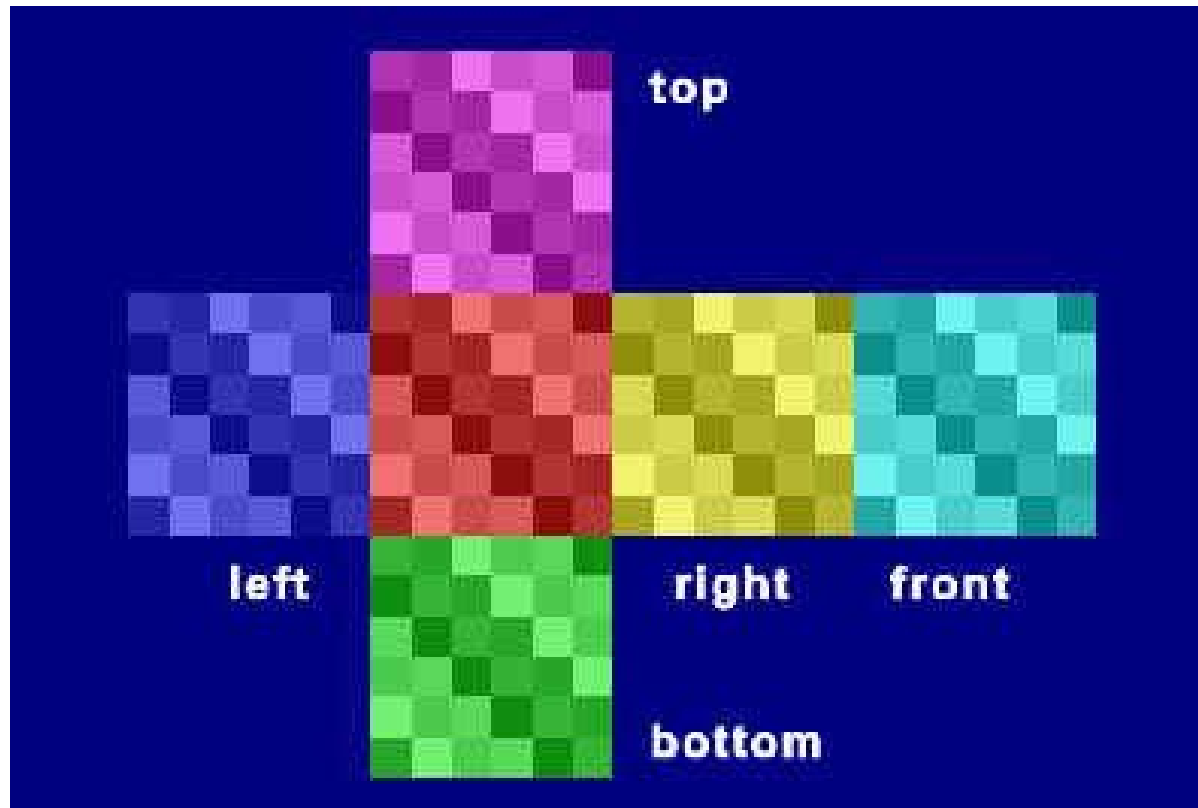
- Determine relevant side of box
- $(x, y, z)$  to  $(u, v)$



# Box mapping

---

- Box texture

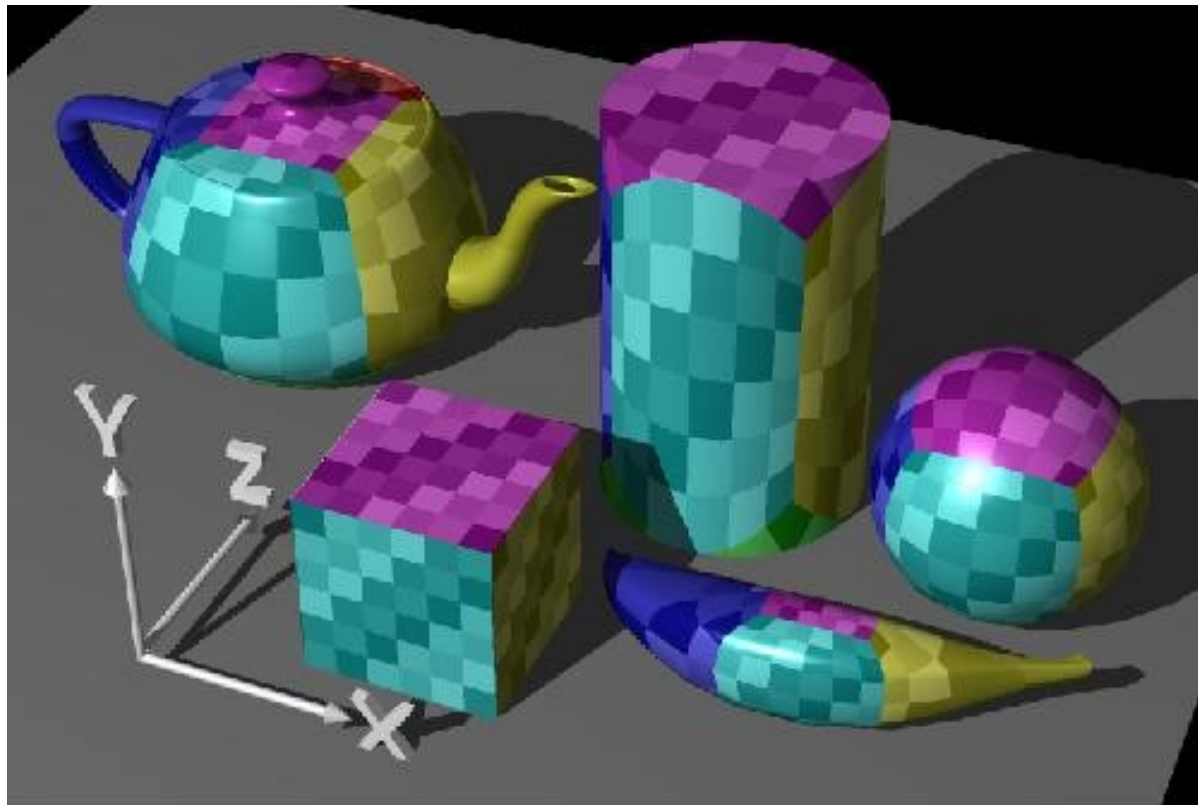




# Box mapping

---

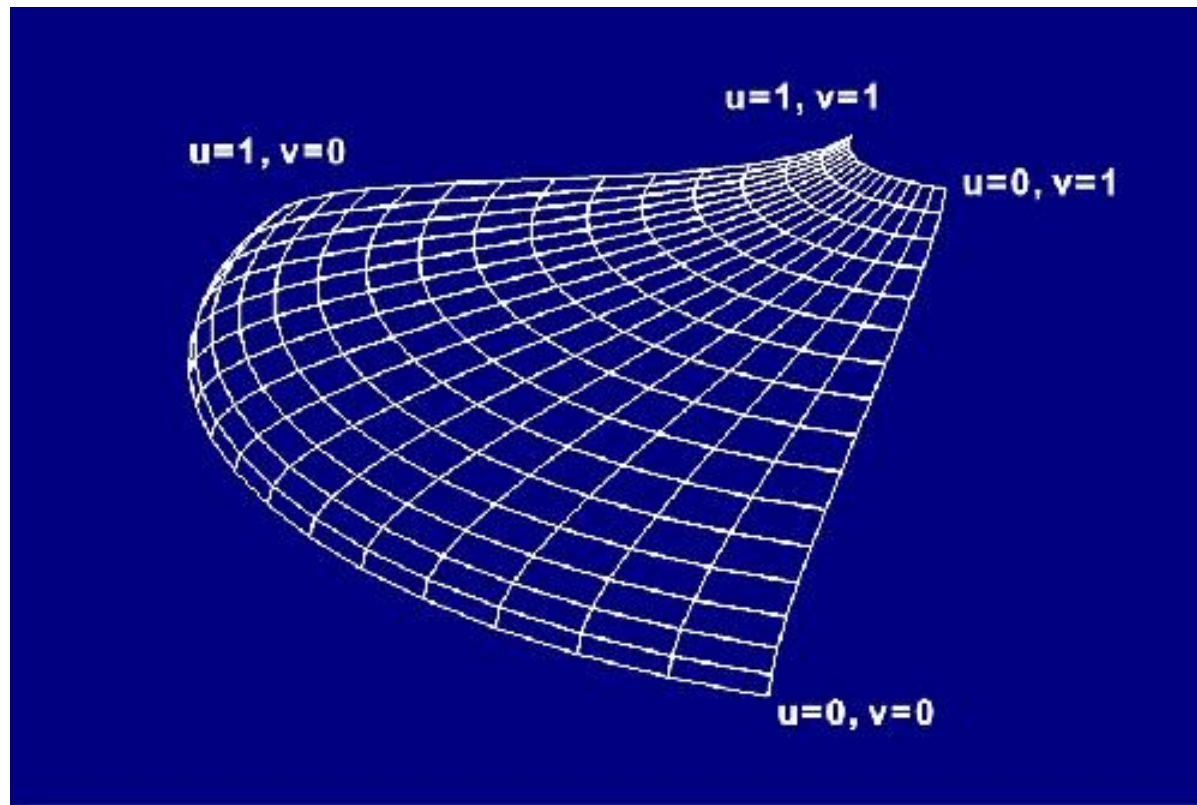
- Determine relevant side of box
- $(x, y, z)$  to  $(u, v)$



# UV-mapping of geometry

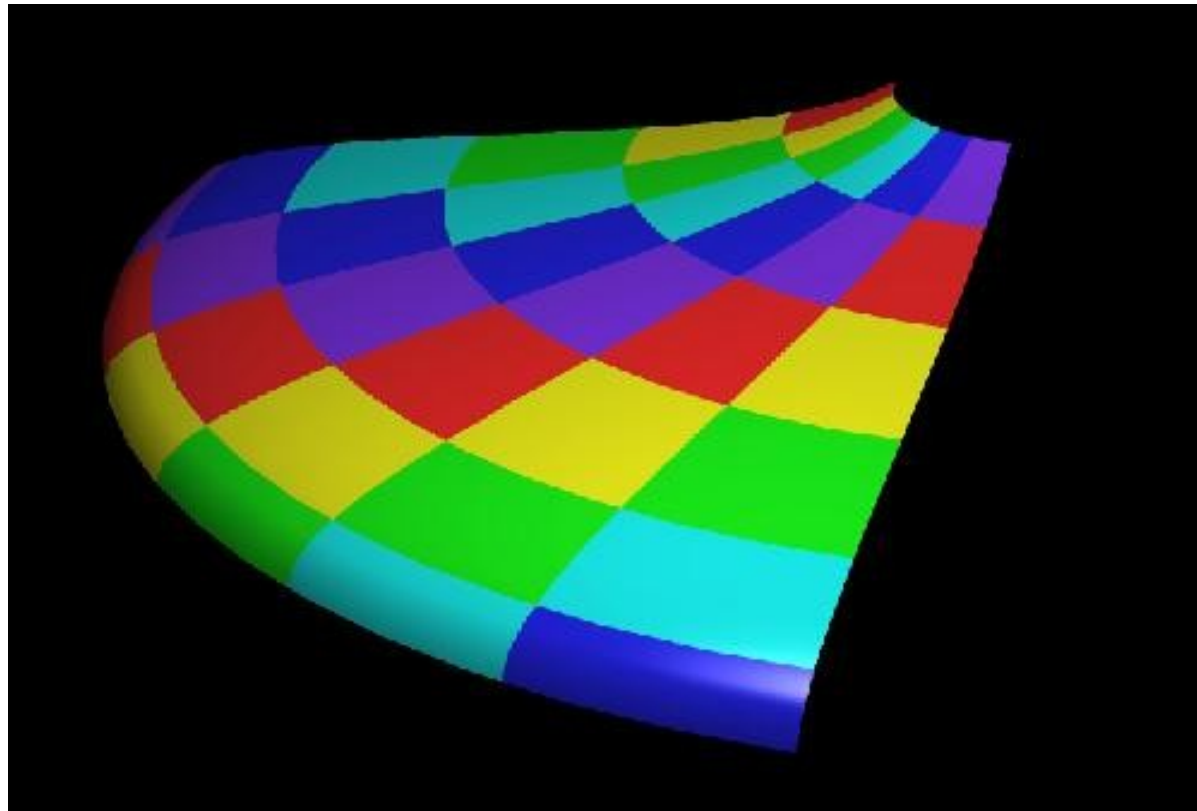
---

- Parametric surfaces



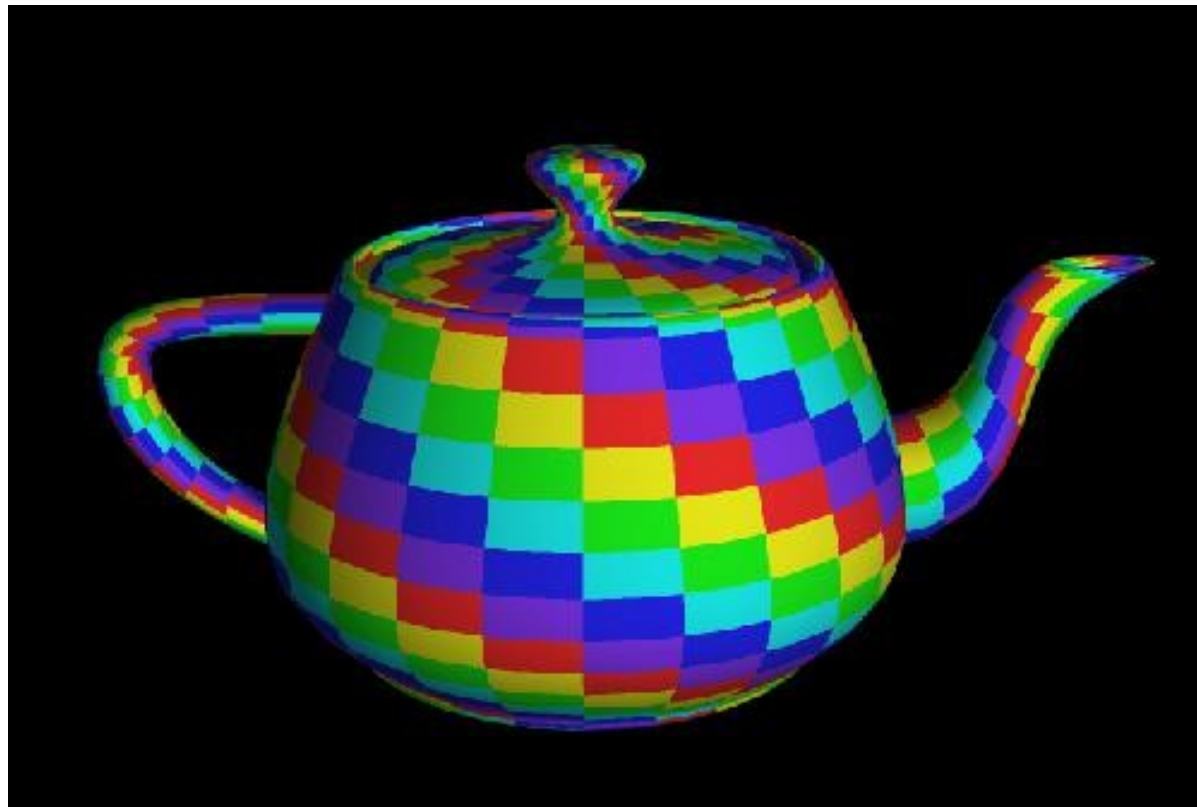
# Parametric surface

---



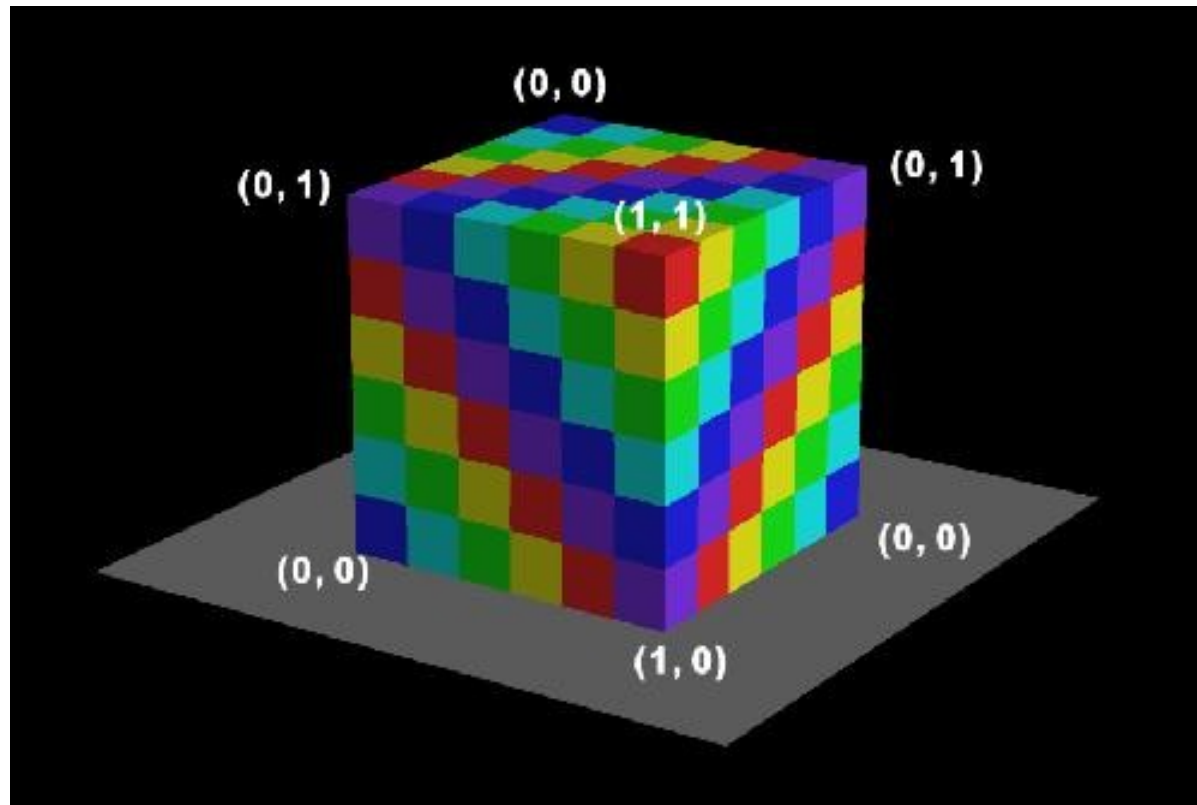
# Parametric surface

---



# Define uv-mapping

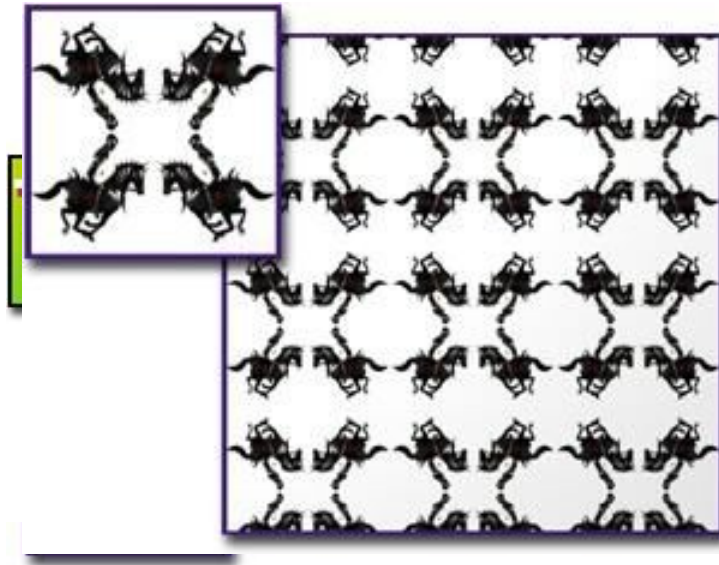
---



# Mapping

---

- When  $(u,v)$  outside range  $[0, 1]$ 
  - Decal ←
  - Repeat/Tile ←
  - Mirror ←
  - Clamp
  - Border



# Texture combination functions

---

- Texture is combined with material (reflection properties) of object
- Methods

- Replace

$$C = C_T$$

- Modulate

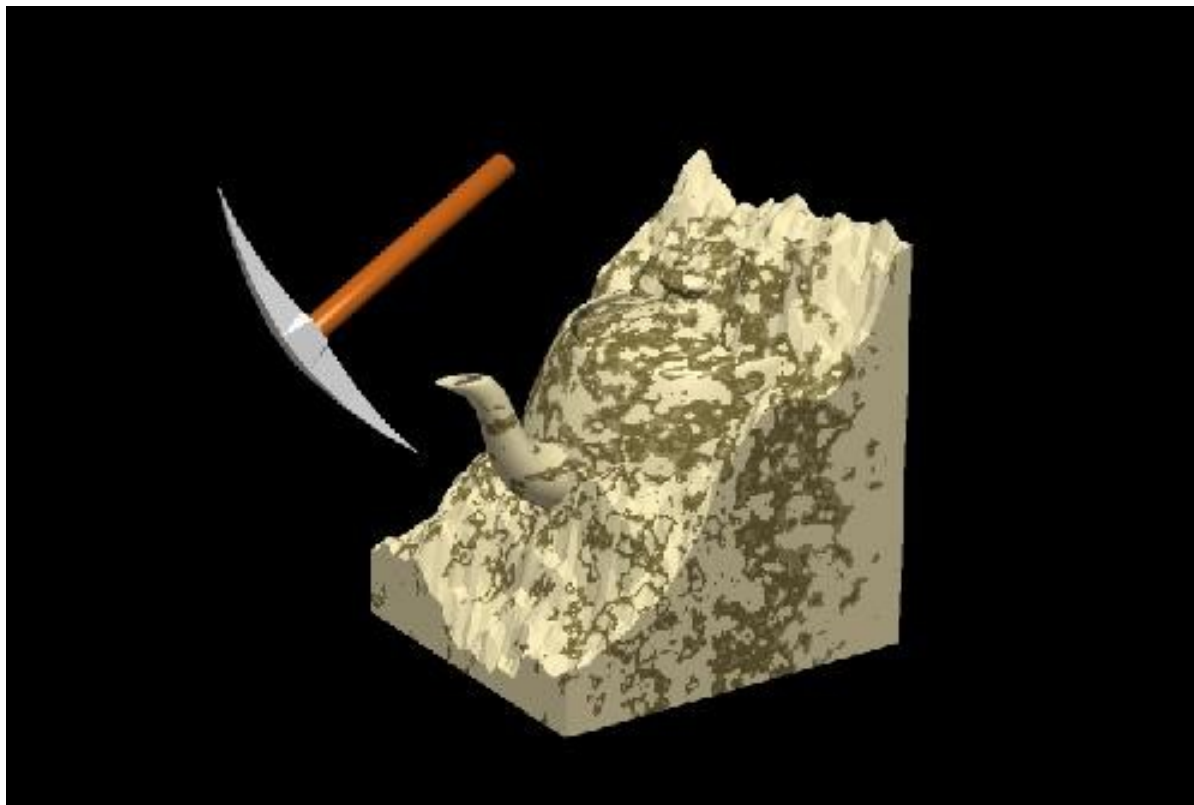
$$C = C_I C_T$$

- Blend

$$C = C_I (1 - C_T) + C_B C_T$$

# 3D mapping

---

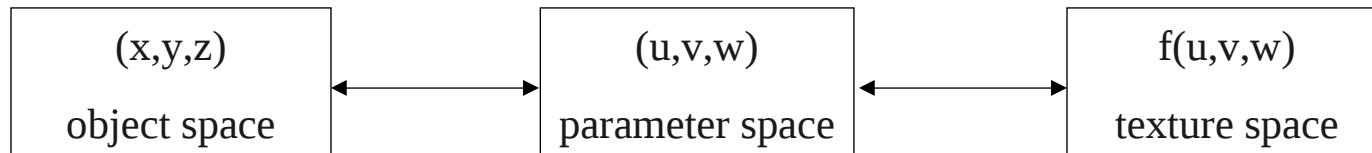
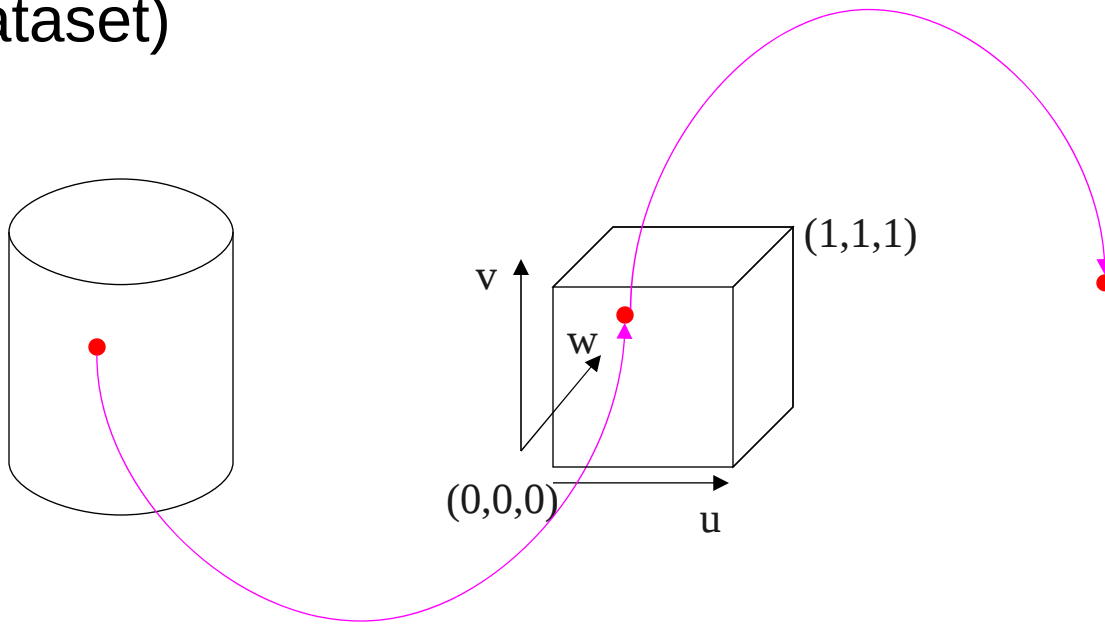




# 3D mapping function

---

- Link a 3D object with a 3D texture (function of dataset)



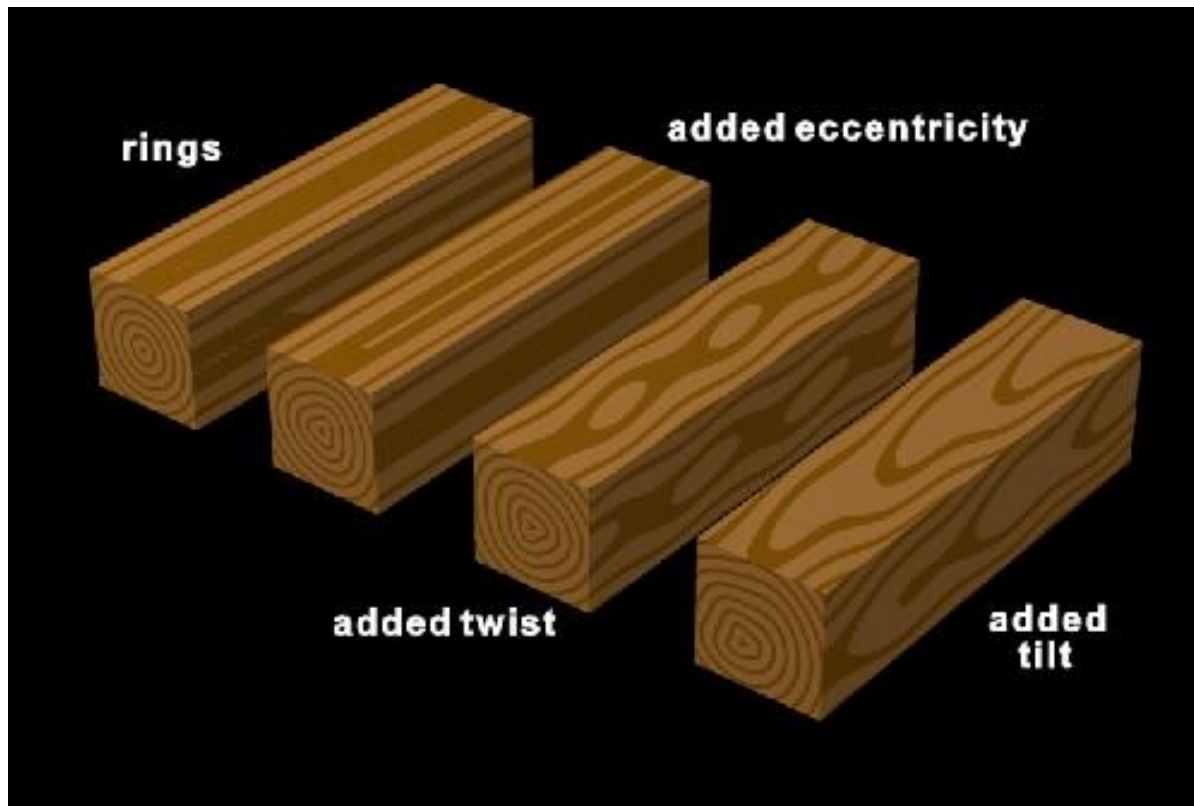
# 3D mapping example

---



# 3D mapping example

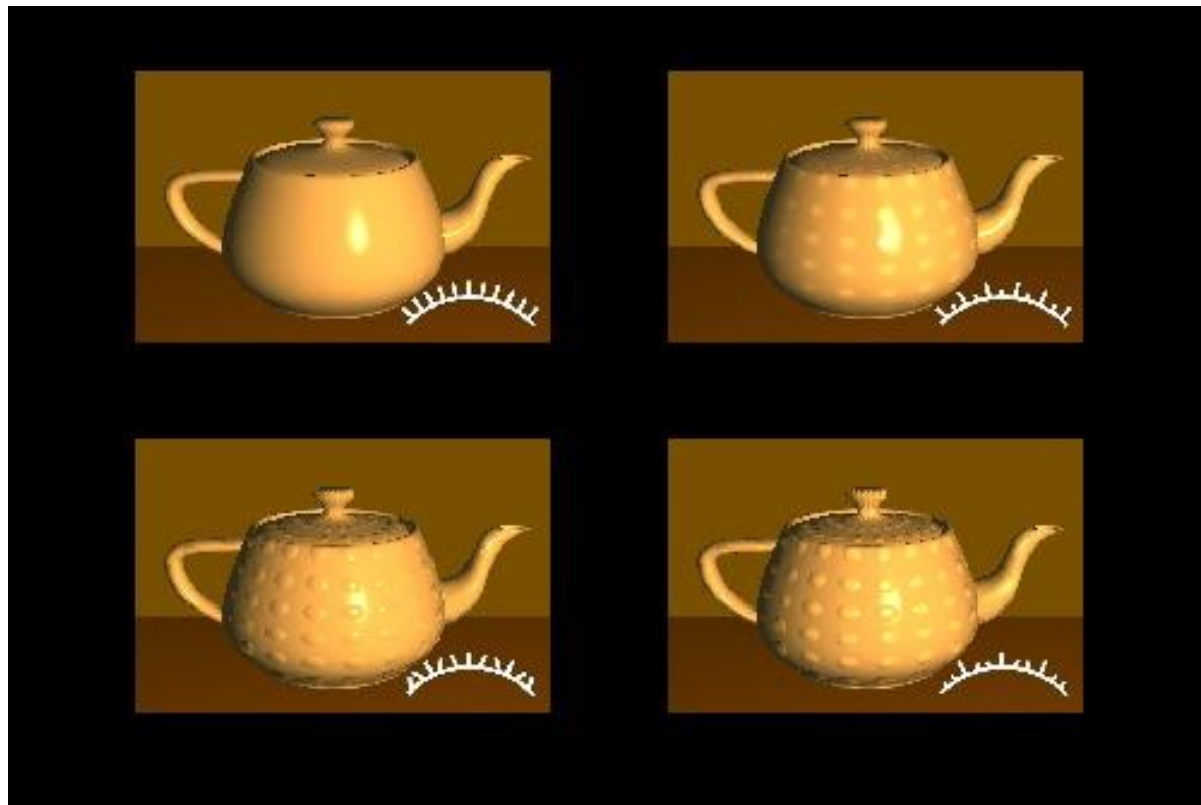
---



# Bump mapping

---

- Use texture to modify normal of point on surface
- Illumination uses modified normal



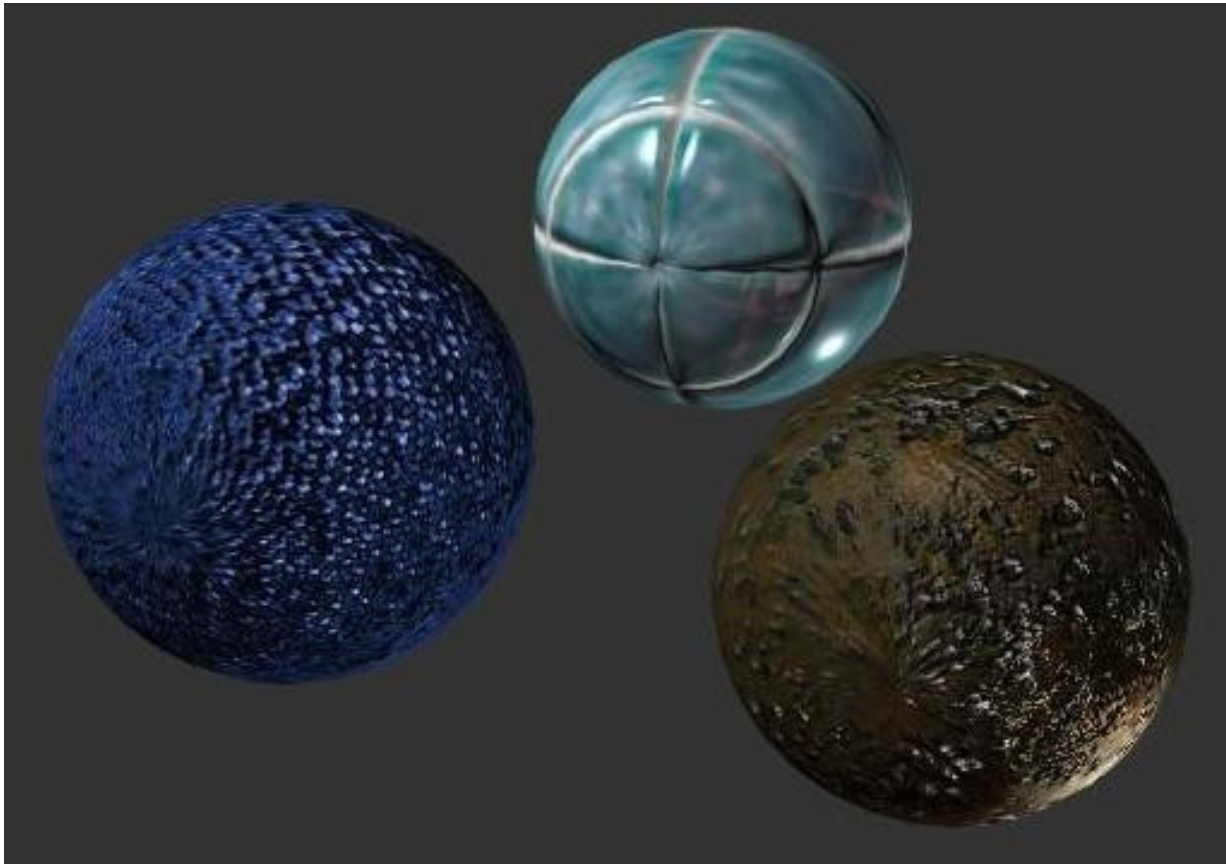
# Bump mapping

---



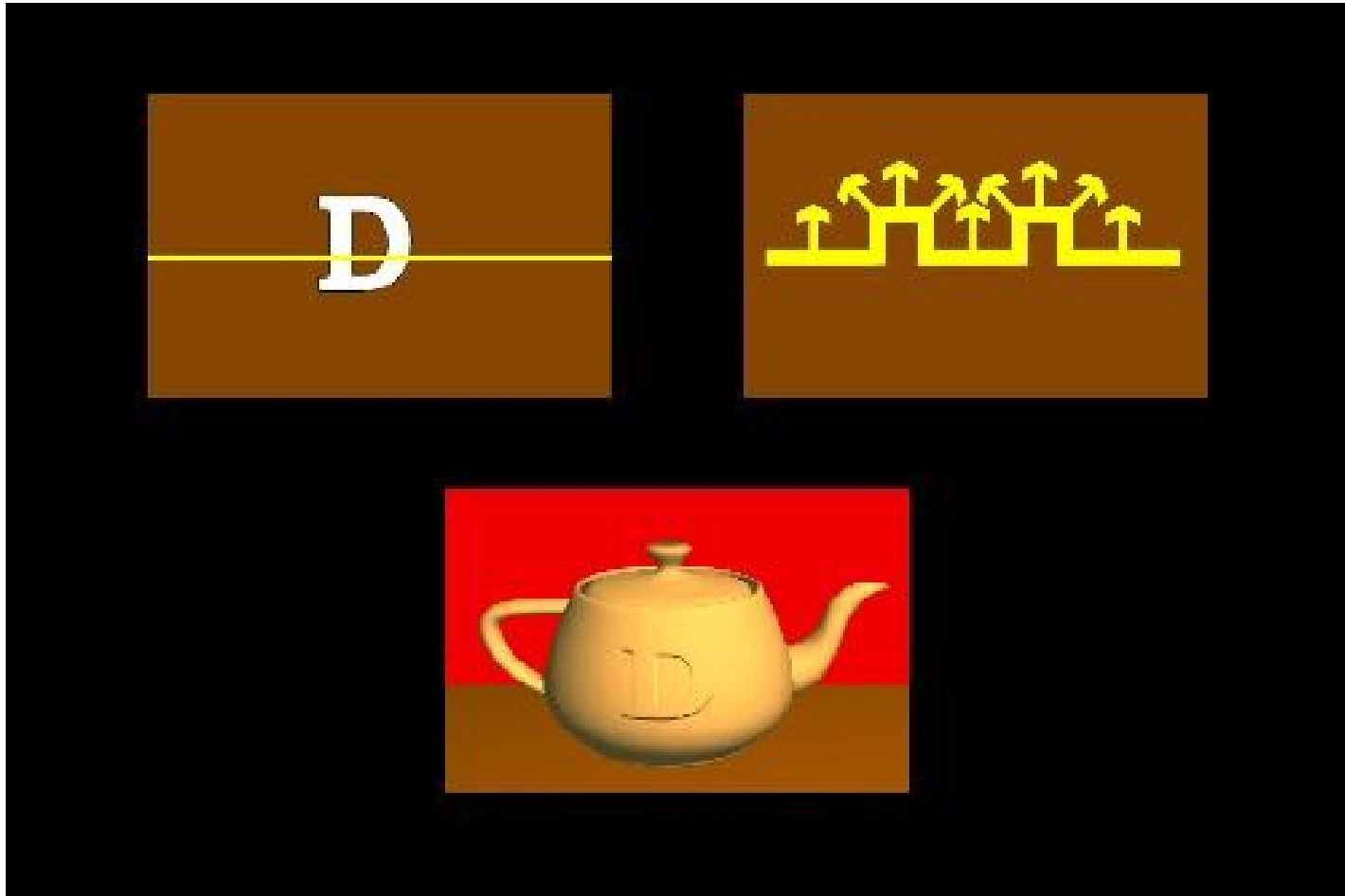
# Bump mapping examples

---



# Bump mapping

---



# Environment mapping

---

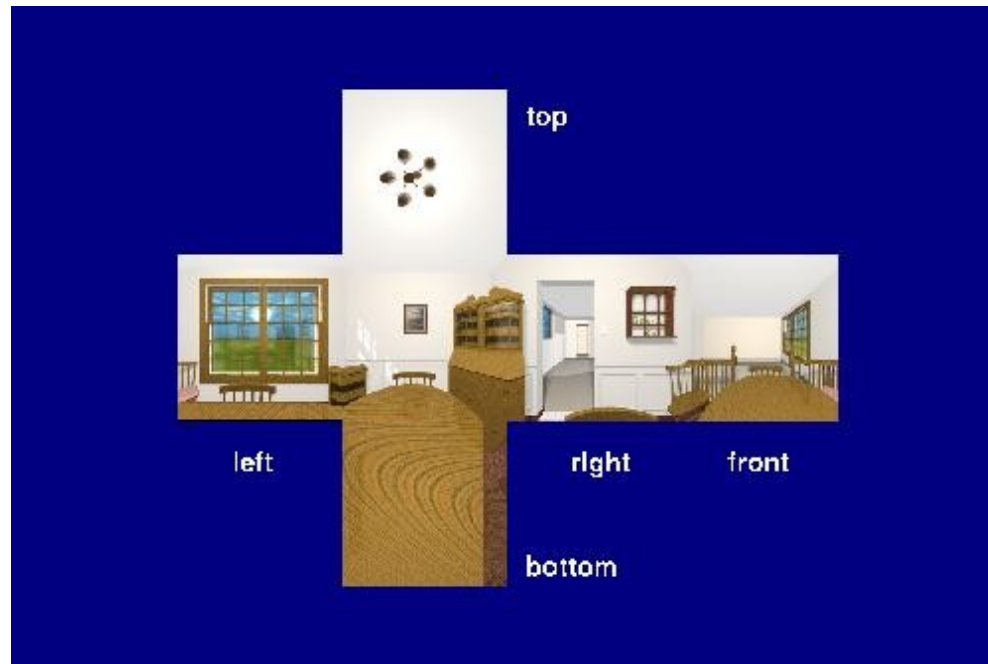
- Use texture mapping to simulate reflections
  - Less accurate compared to ray tracing
  - Much faster
  - No specular-specular reflections



# Environment mapping

---

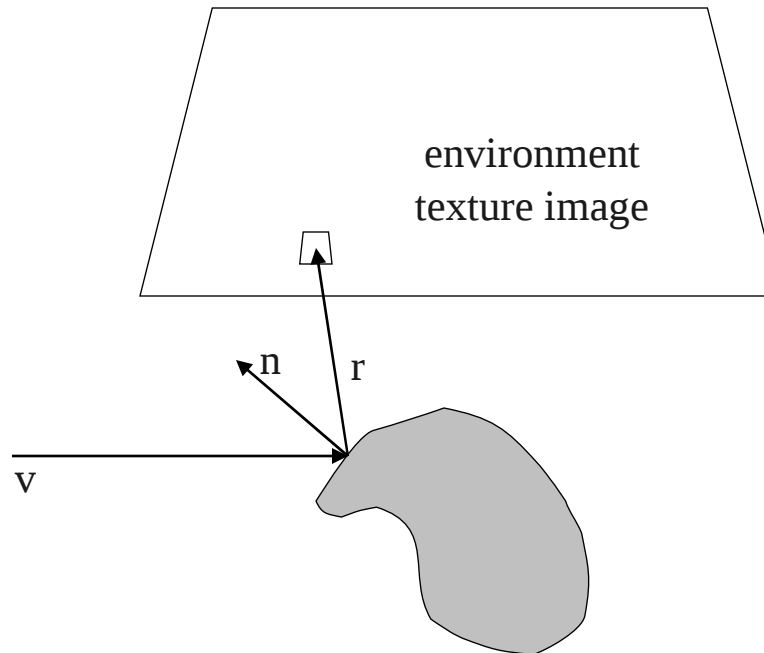
- Render scene 6 times from center of reflective object
- Use resulting mirrors as texture



# Environment mapping

---

- At visible point
  - Compute reflection direction:  $r = 2(n \cdot v)n - v$
  - Use this direction to determine side in box and  $(u,v)$  values within texture



# Environment mapping

---



# Environment mapping & ray tracing

---



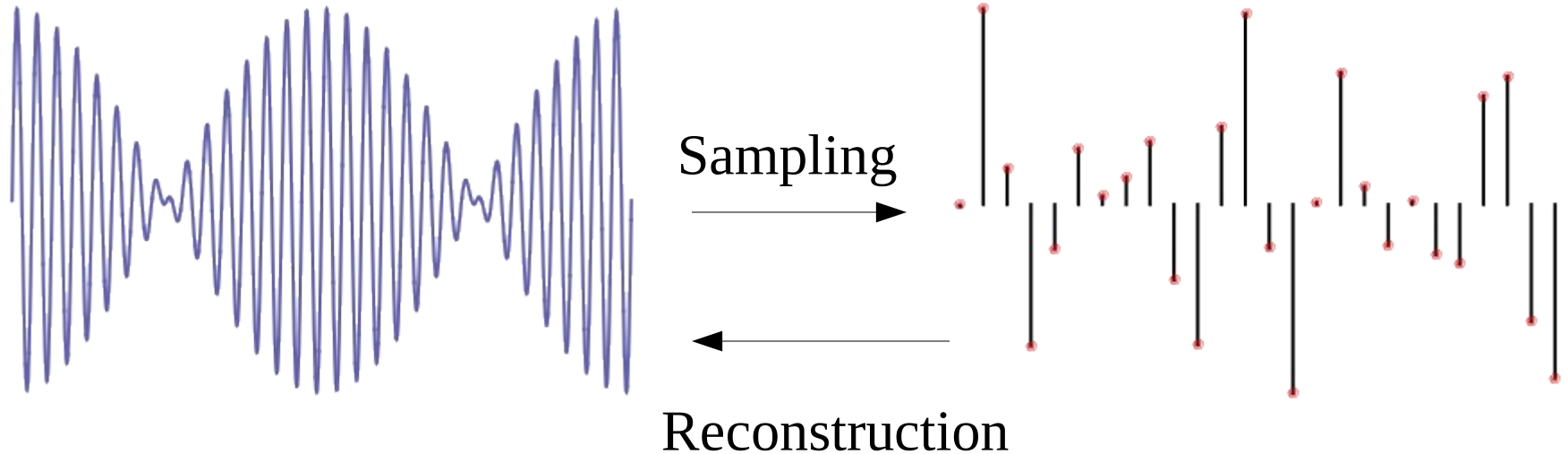
# Aliasing

---

- Aliasing is error in signal caused by limitations in system that generates or processes signal
  - Occurs when there is not enough information to represent something

# Why called aliasing?

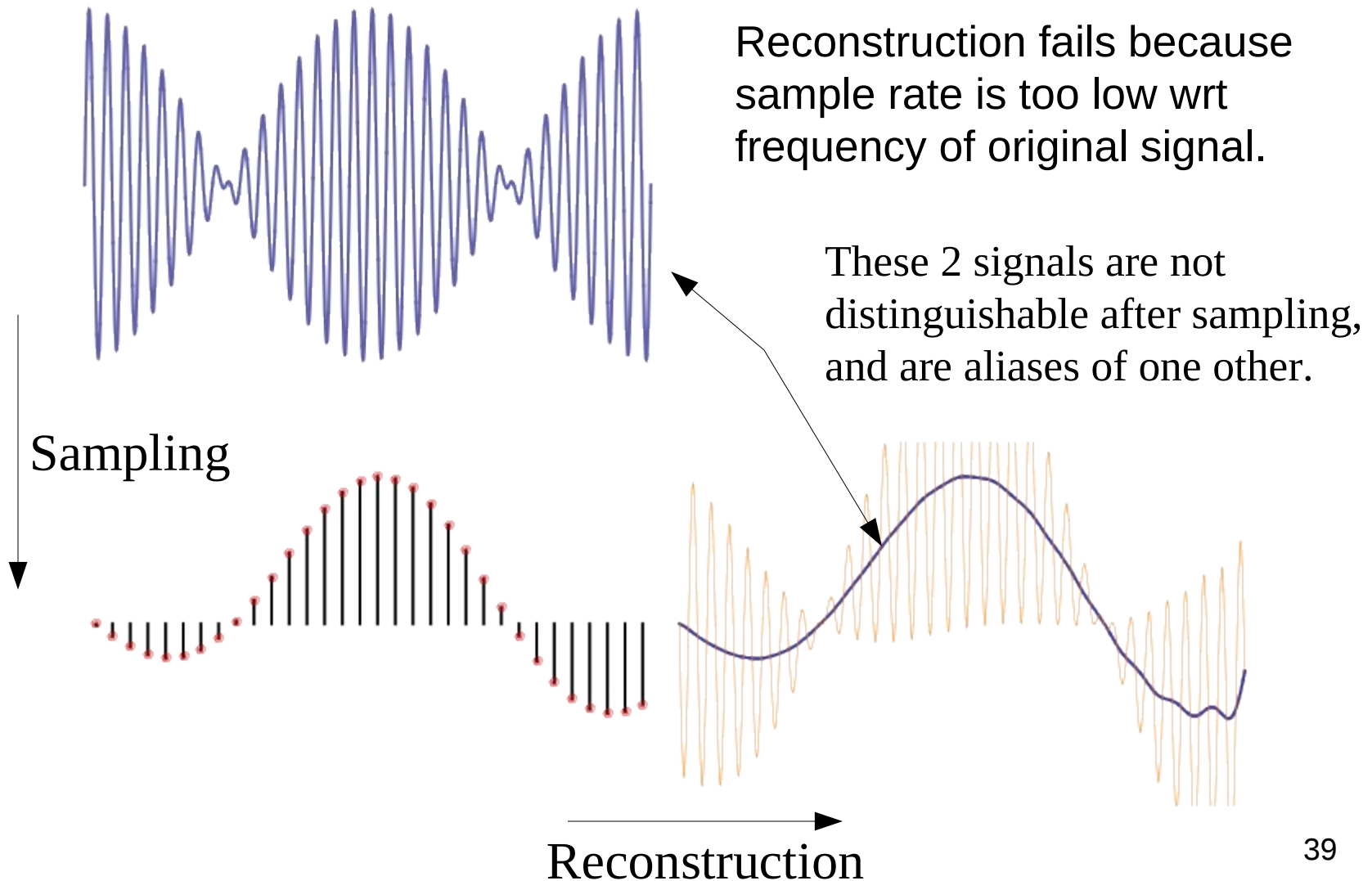
---



However, if sampling is too inaccurate, reconstruction is impossible.

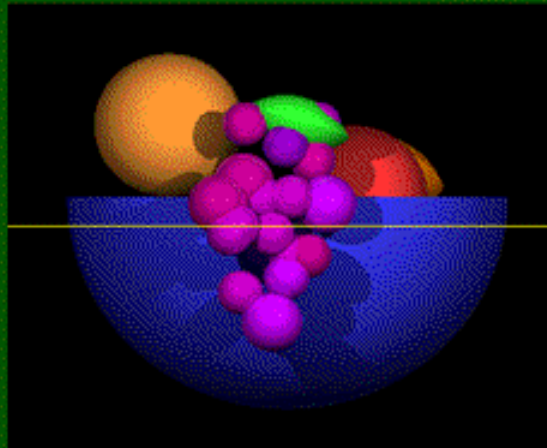
# Why called aliasing?

---

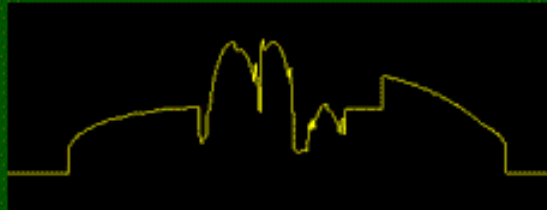


# Aliasing

---



**Original  
scene**

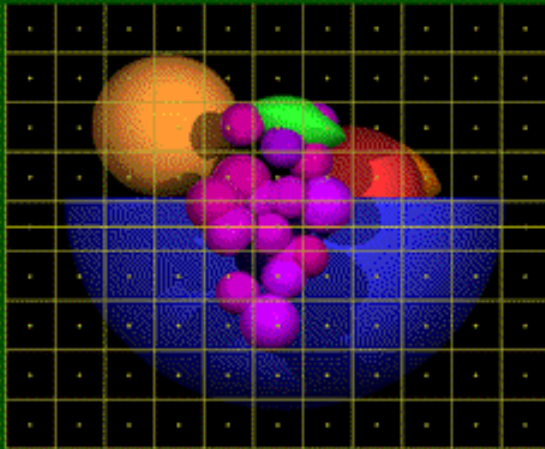


**Luminosity  
signal**



# Aliasing

---



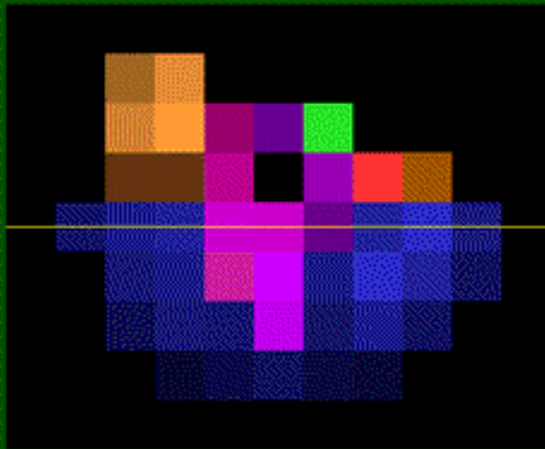
**Sampling at  
pixel centers**



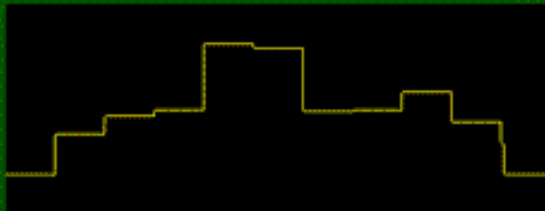
**Sampled  
signal**

# Aliasing

---



**Rendered  
image**



**Luminosity  
signal**

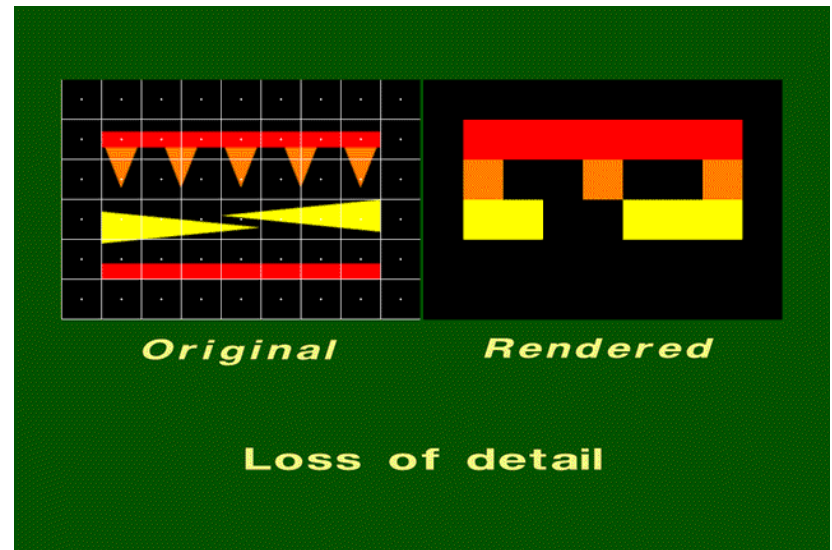
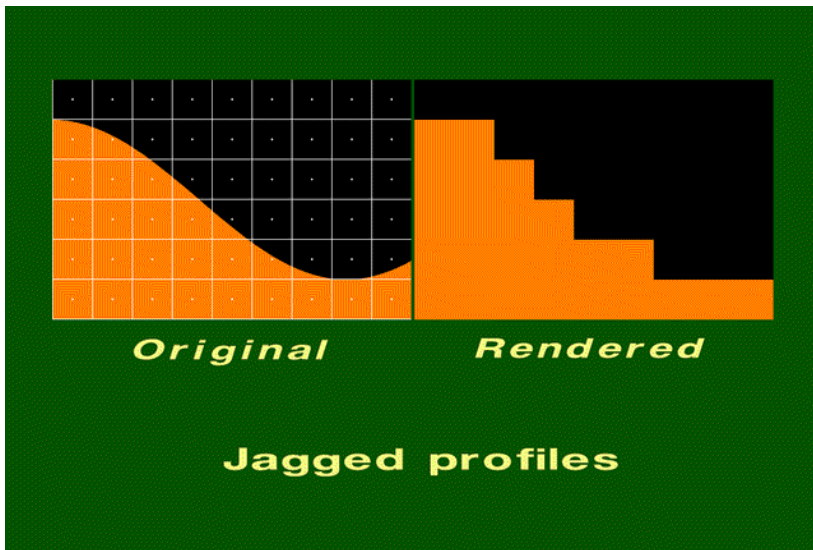
# Examples of aliasing

---

- Jagged edges (jaggies)
- Loss of detail
- Temporal aliasing

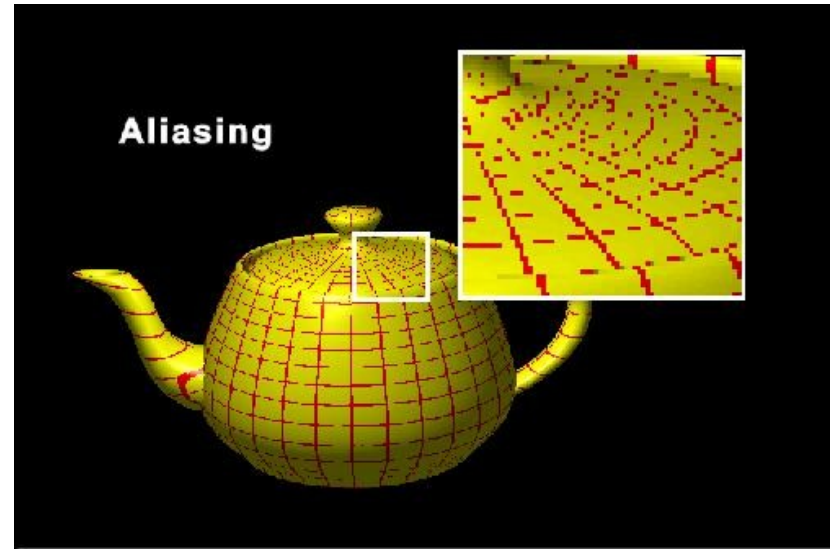
# Aliasing

---



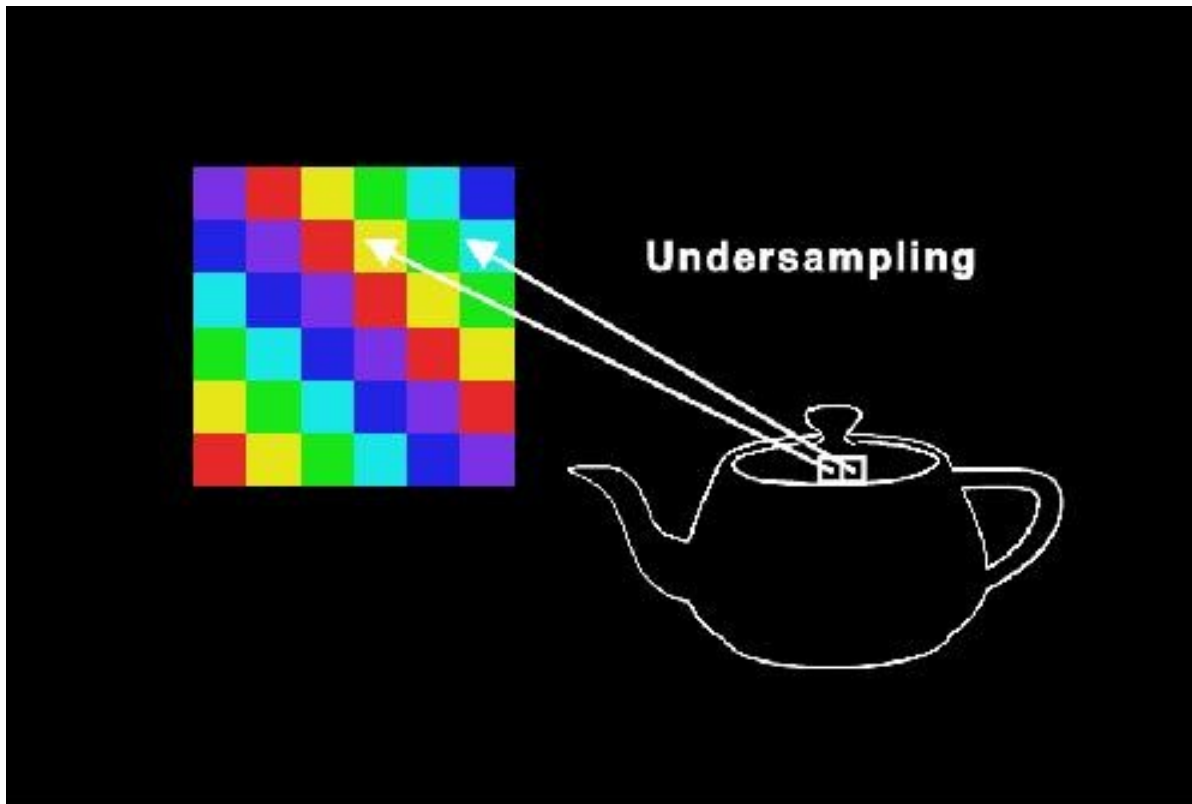
# Aliasing in textures

---



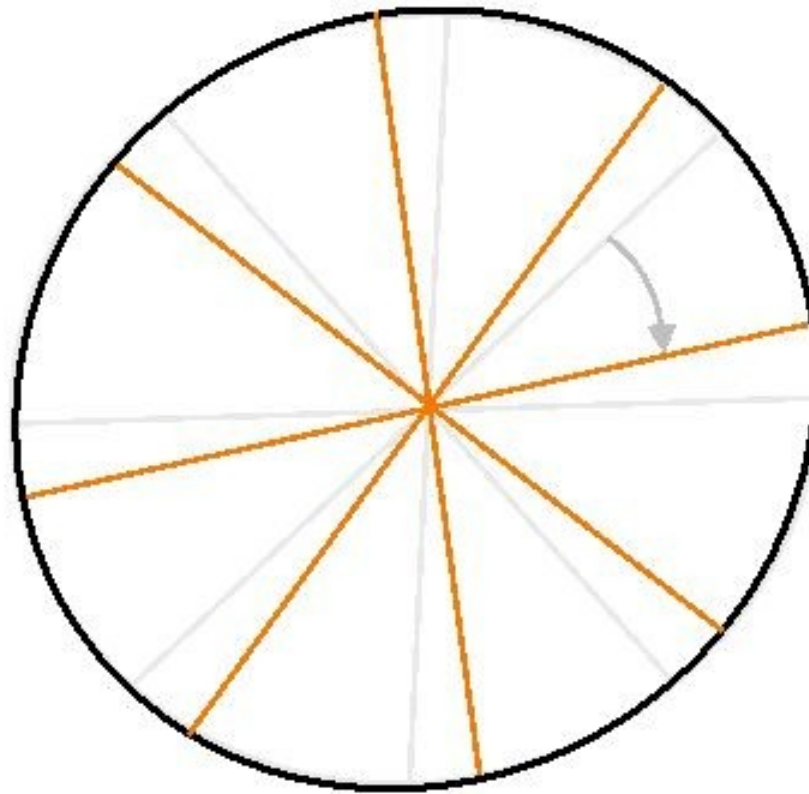
# Aliasing in textures

---



# Temporal aliasing

---



# Anti-aliasing

---

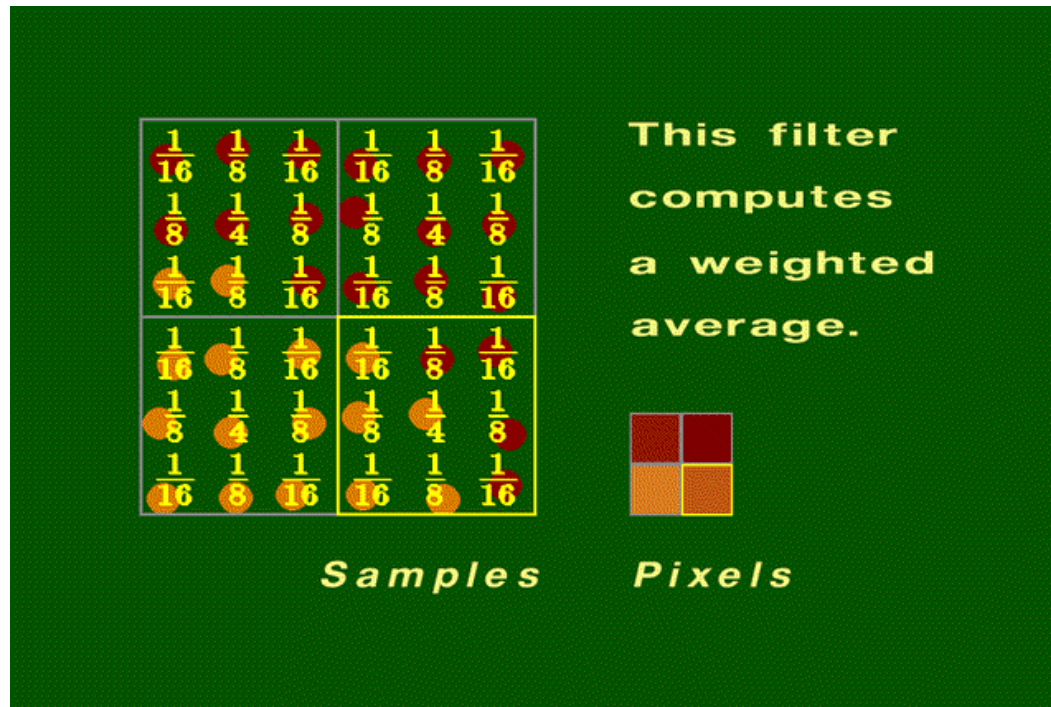
- Minimization of aliasing effects
- Methods
  - Pre-filtering
  - Post-filtering



# Postfiltering / supersampling

---

- More samples per pixel
- Pixel color is average of samples



# Postfiltering

---

