Filtering and Mipmaps

- Texture Interpolation

This photo is too small.

Filtering and Mipmaps

- □ Zooming

Consider a black and white image:

Task:

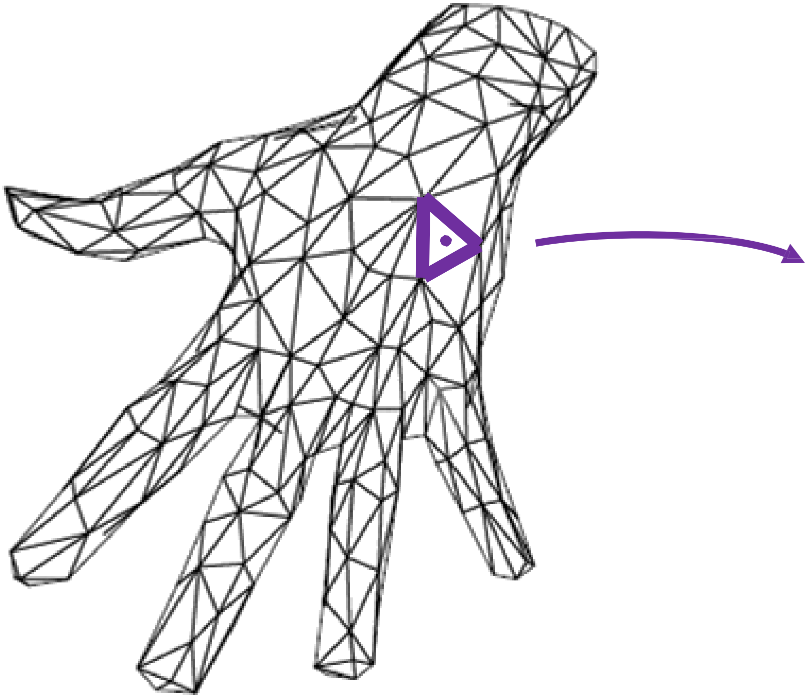
Blow up to poster size (zoom by a factor of 16)

Filtering and Mipmaps

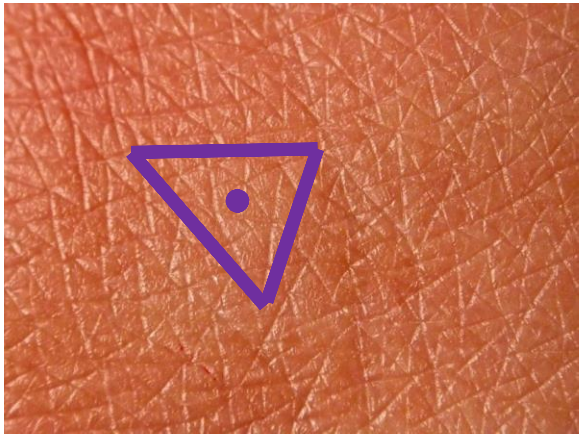
- Interpolation
  - Given: The values of a function f at a few locations. f(1), f(2), f(3), …
  - Compute: The values else where What is f(1.5)?
  - The challenge: Modeling how the function “should” behave.

Filtering and Mipmaps

- When Does Interpolation Happen?

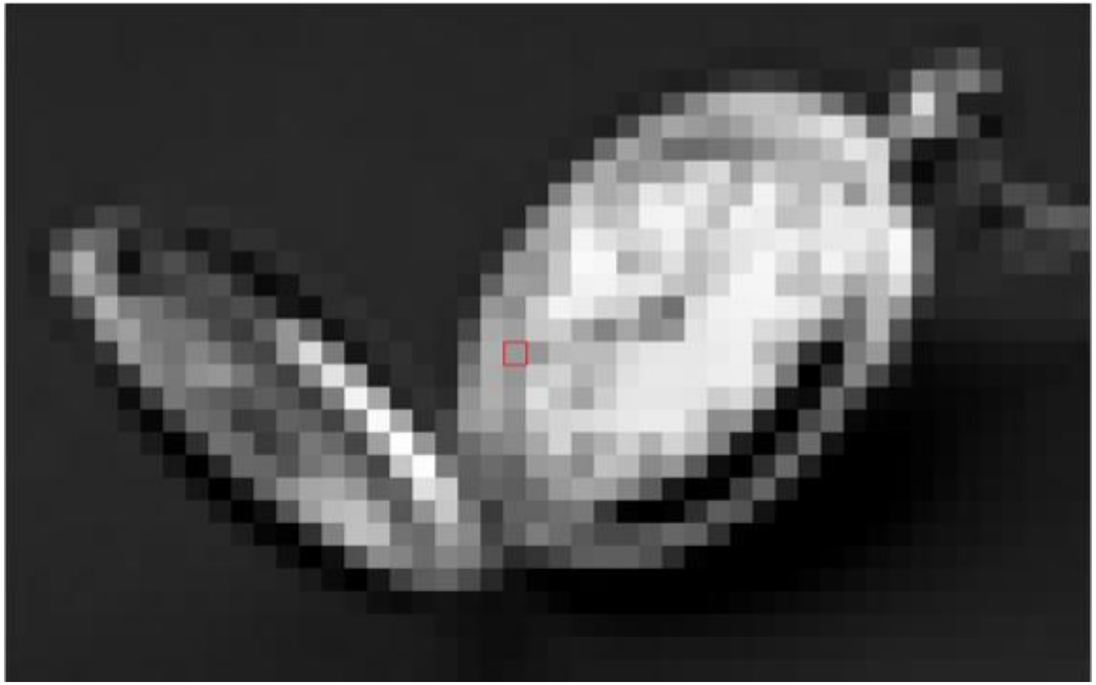


(s,t) may not land
at an integer location!



Filtering and Mipmaps

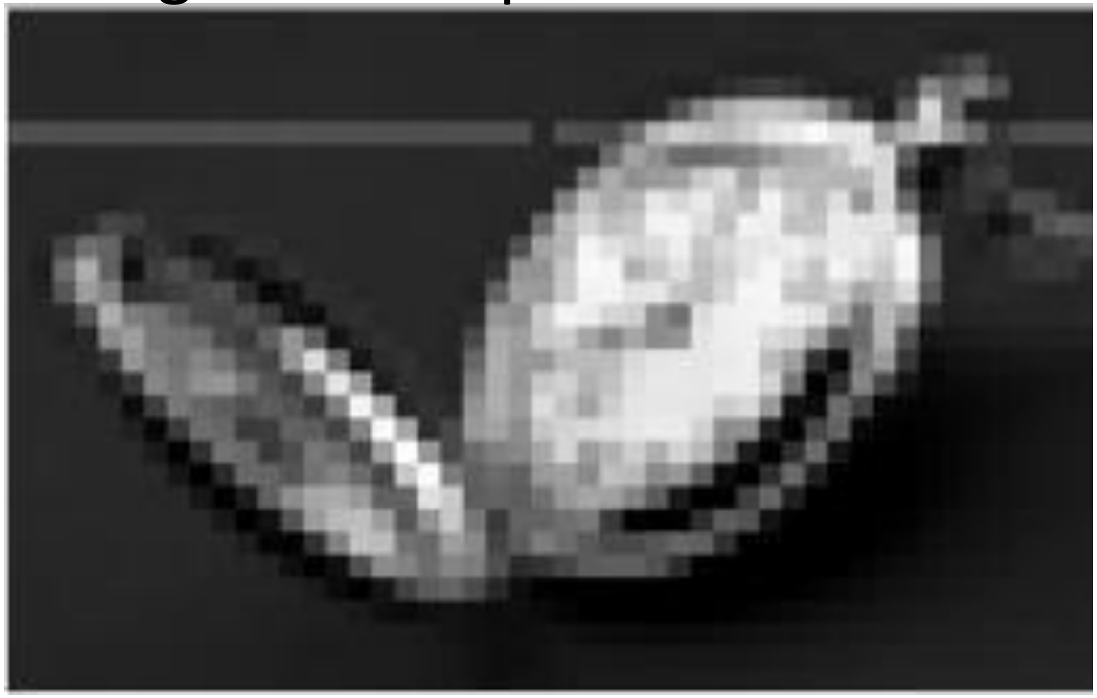
- □ Nearest Neighbor Interpolation



First try: Repeat each row 16 times, then each column 16 times

Filtering and Mipmaps

□ Nearest Neighbor Interpolation



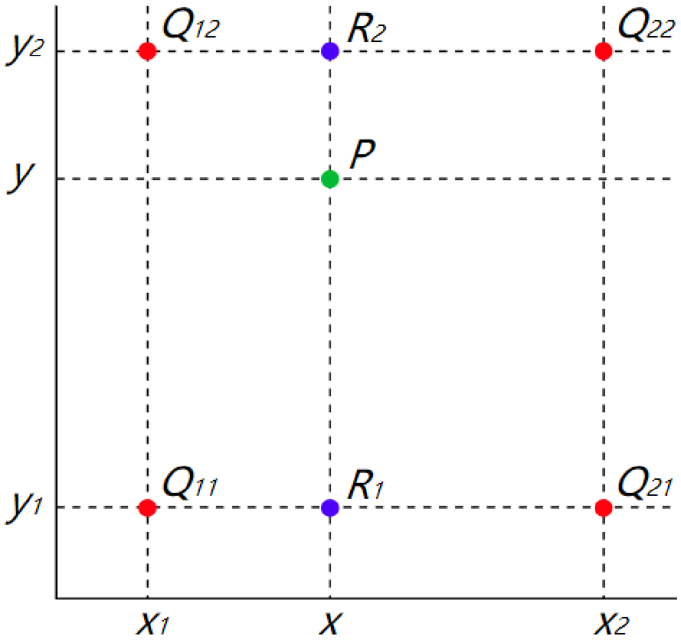
Discontinuous!

- We need a better way to find in-between values

Filtering and Mipmaps

□ Bi-Linear Interpolation

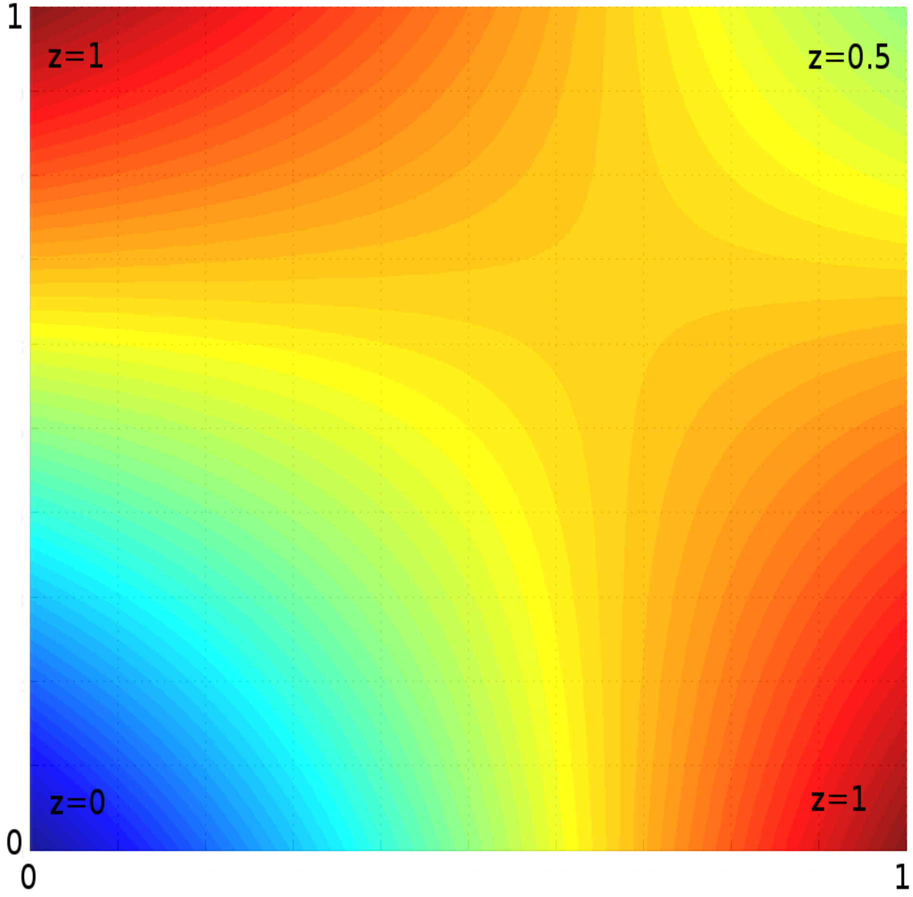
$$
\begin{aligned}
f(x,y) \approx\ & \frac{f(Q_{11})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y) \\
&+ \frac{f(Q_{21})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y) \\
&+ \frac{f(Q_{12})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1) \\
&+ \frac{f(Q_{22})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1).
\end{aligned}
$$



Interpolate in x then in y (or vice versa!)

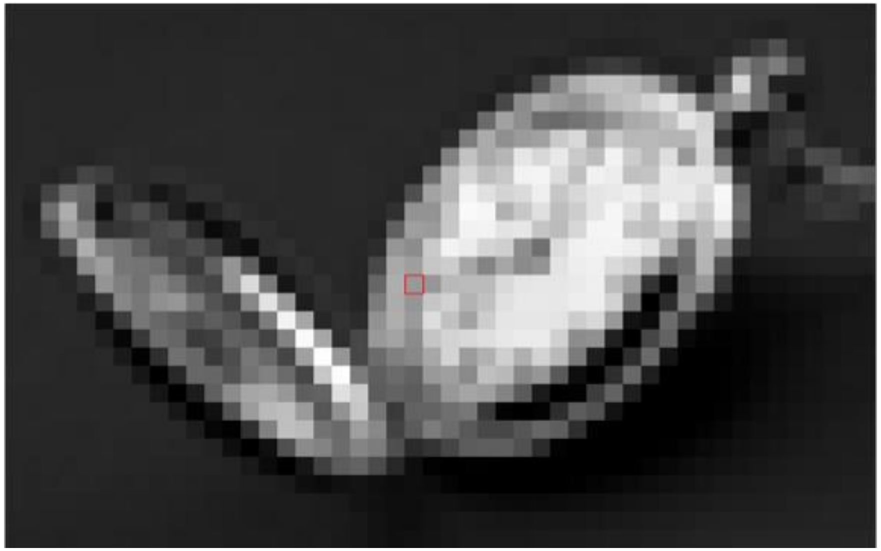
Filtering and Mipmaps

- □ Bi-Linear Interpolation

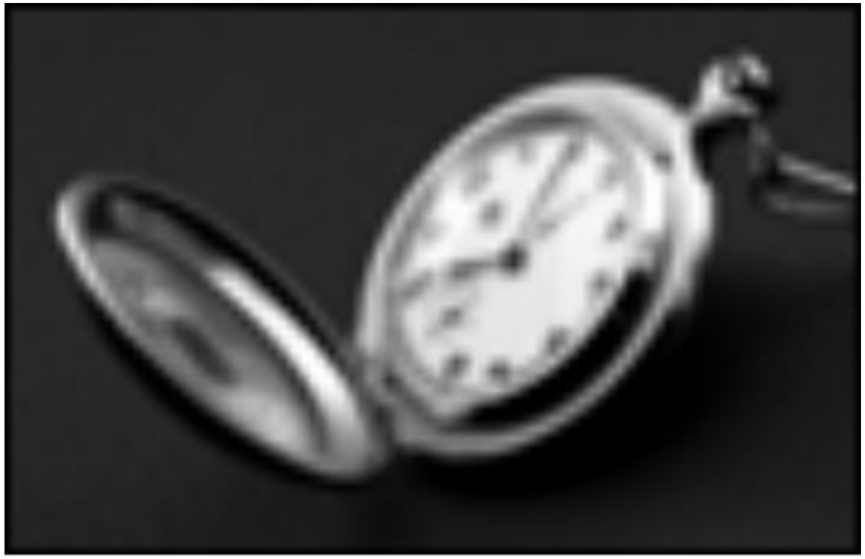


Filtering and Mipmaps

- □ Comparison



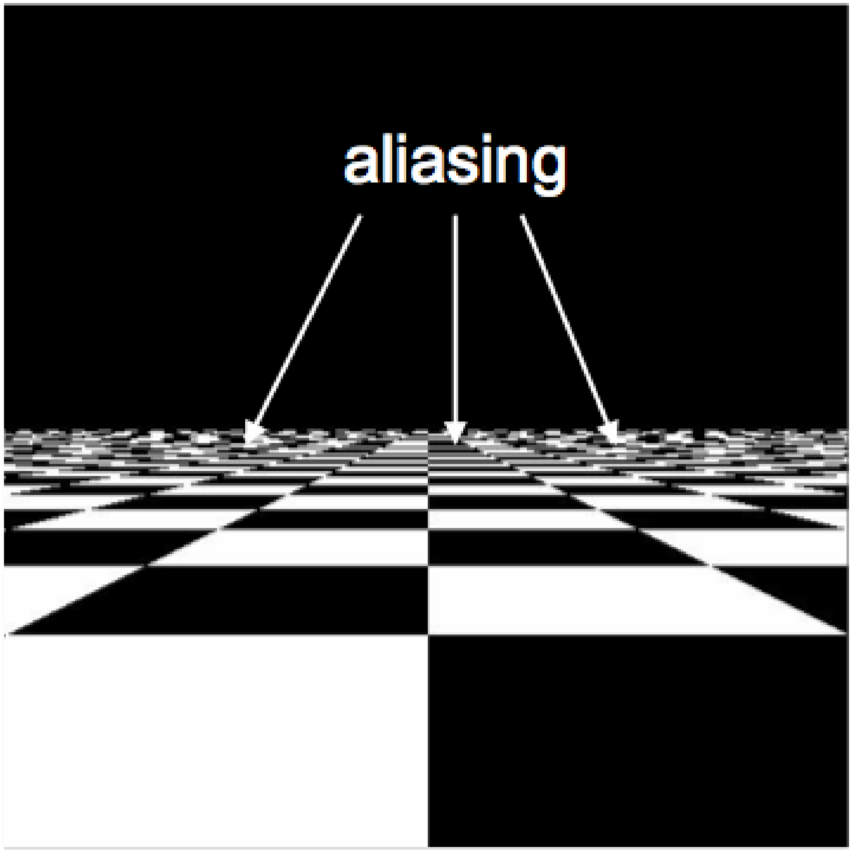
Nearest Neighbor



Bilinear

Why Do We Need to Filter?

- □ Texture image is shrunk in distant parts of the image
- □ This leads to aliasing
- □ Can be fixed with filtering
  - bilinear in space
  - trilinear in space and level of detail (mipmapping)

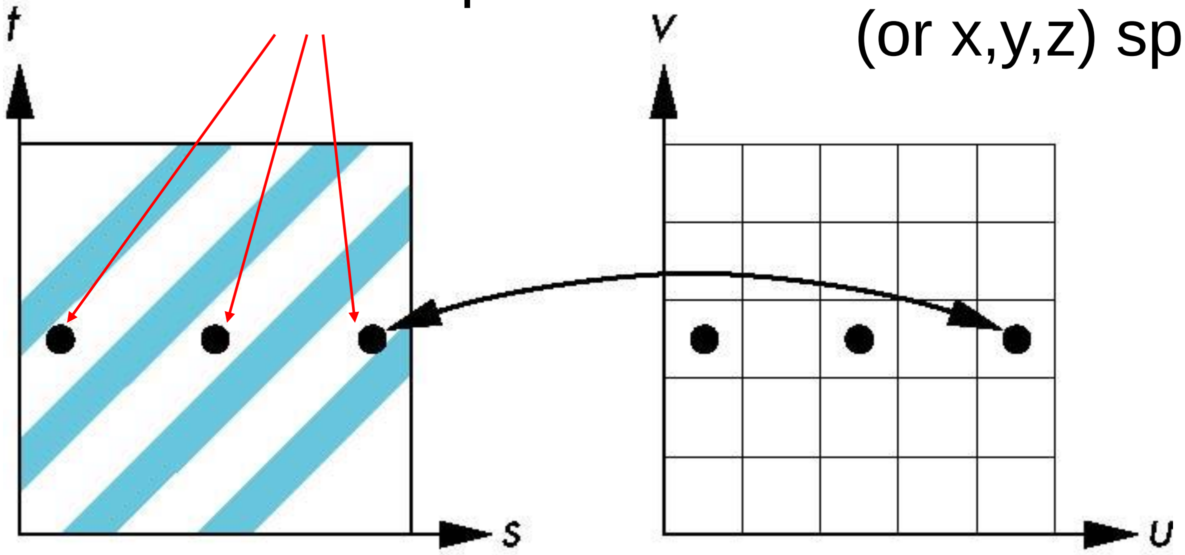


Aliasing

- □ Point sampling of the texture can lead to aliasing errors

miss blue stripes

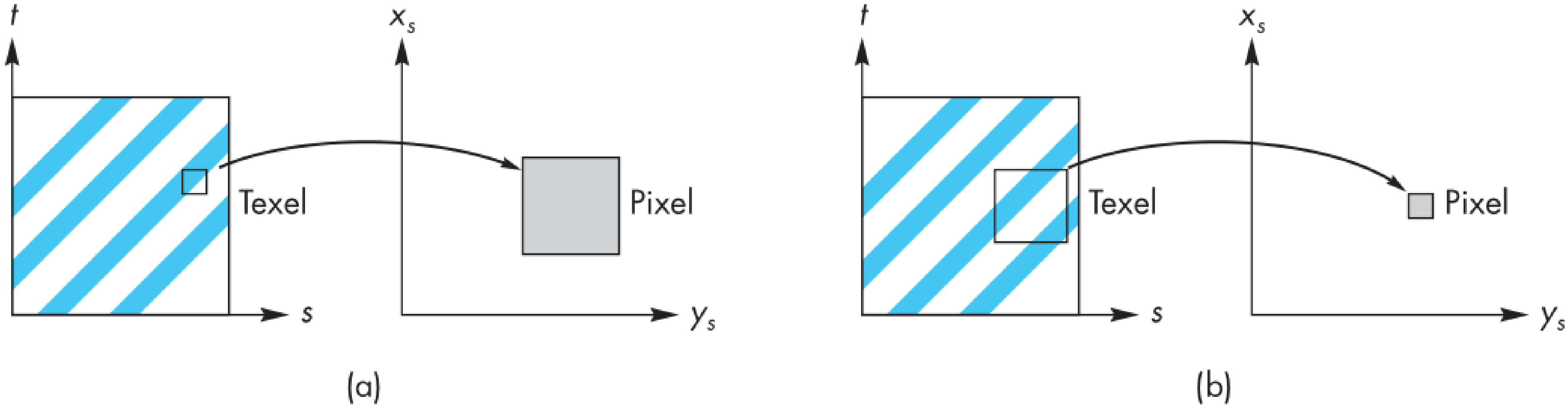
point samples in u,v (or x,y,z) space



point samples in texture space

Aliasing

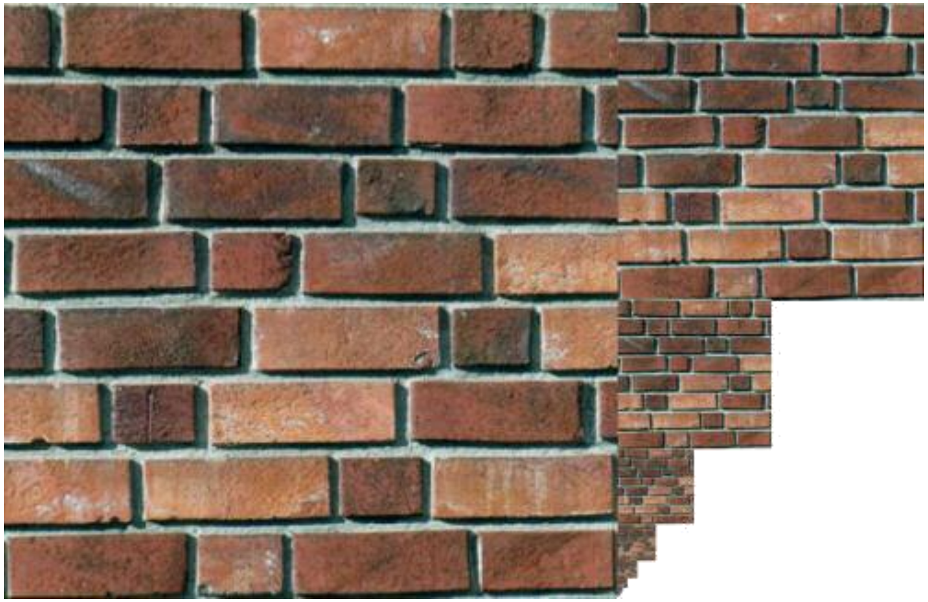
- The size of the pixel that we are trying to color on the screen may be smaller or larger than one pixel (texel) of the texture



Mapping texels to pixels. (a) Minification. (b) Magnification

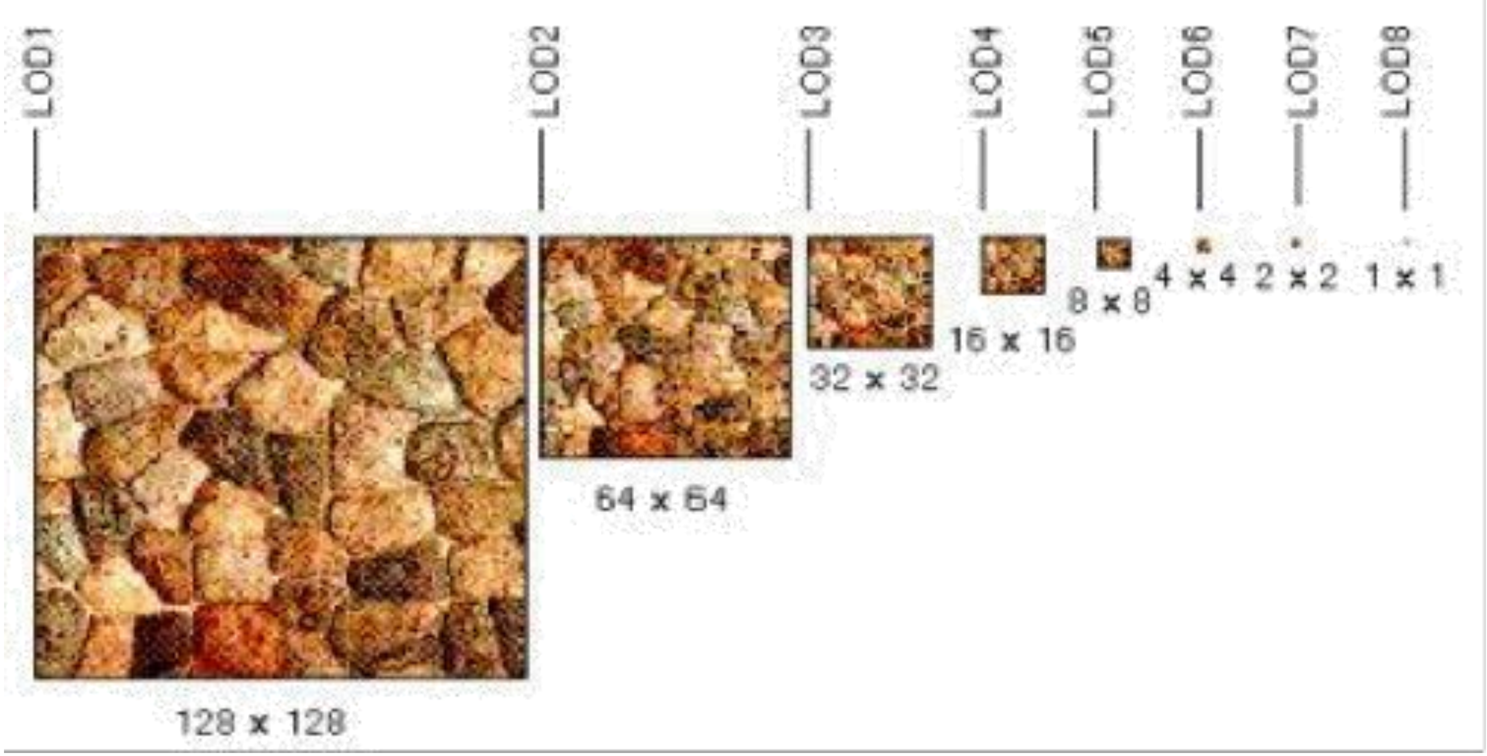
Filtering and Mipmaps

- Mipmapping
  - Precompute texture at different scales and use the appropriate texture at each distance
  - When rendering, choose scale to avoid having to minify on the fly



Filtering and Mipmaps

- Mipmapping
  - Each piece represents one level of detail(LOD)
  - Simplified by using powers of two



2

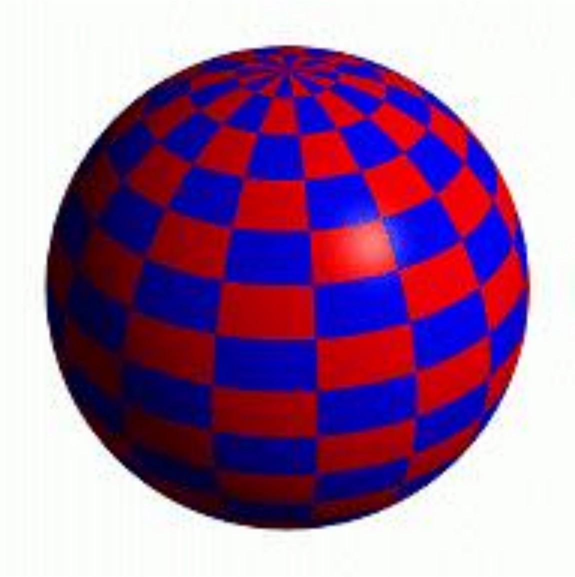
Applications

Texture Mapping Applications

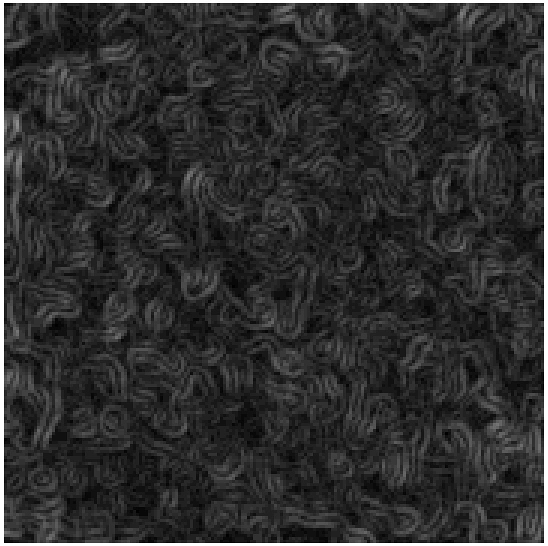
- Light maps
- Bump mapping
- Displacement mapping
- Environment Mapping
- Procedural texturing
- And many more

Bump Mapping

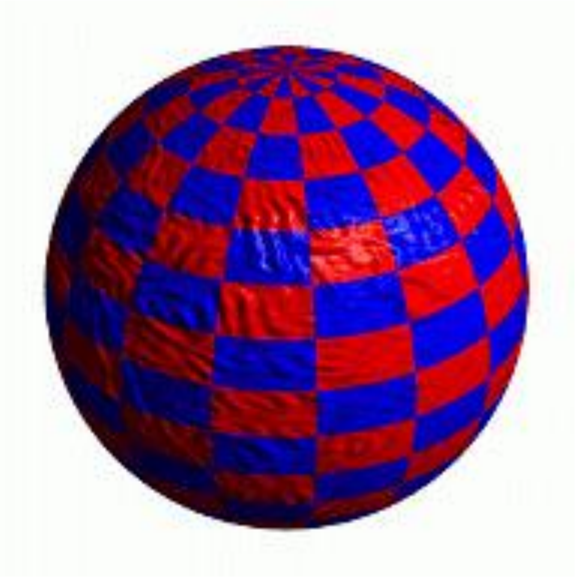
- Texture = change in surface normal!



Sphere w/ texture



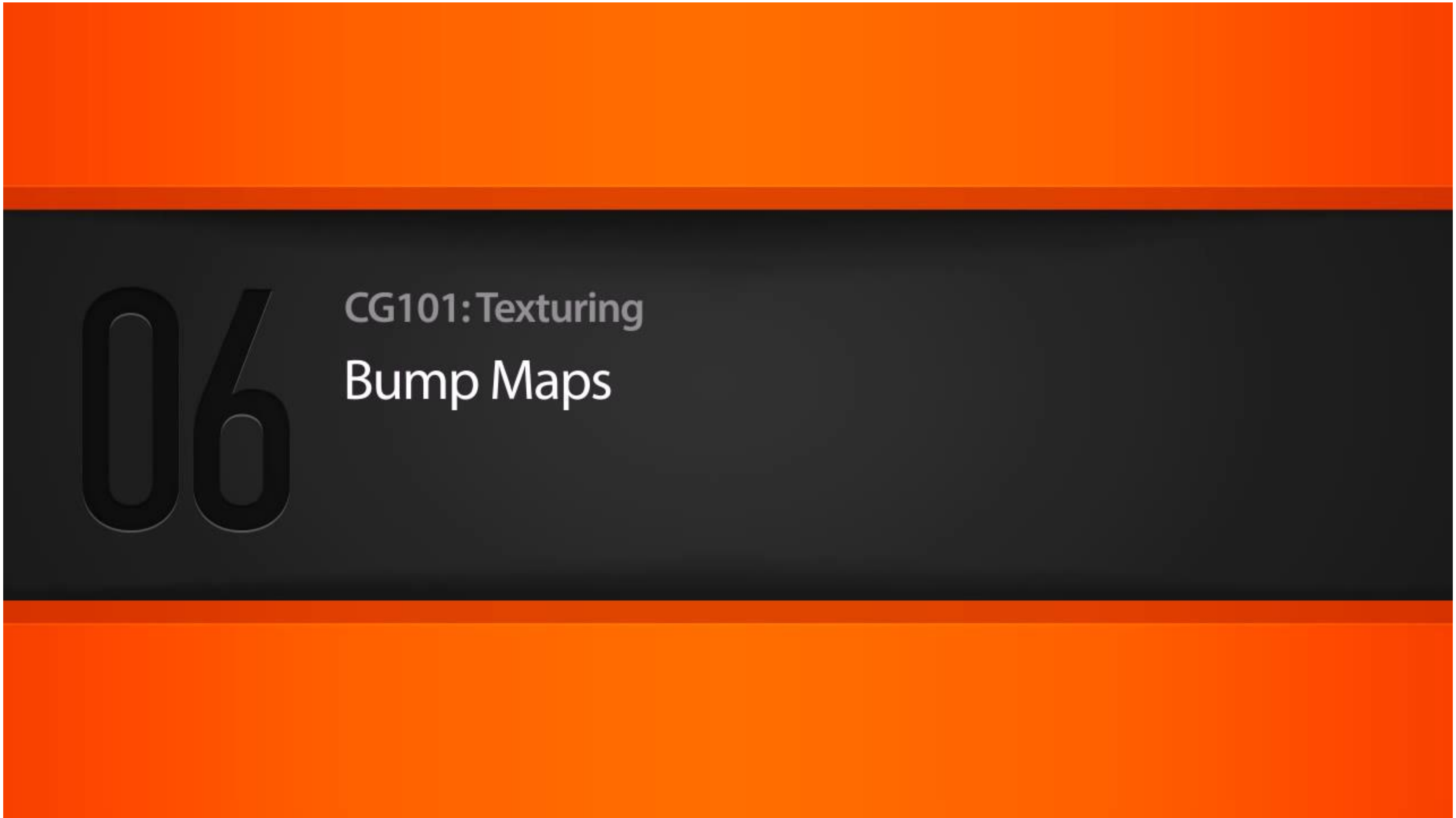
Swirly bump map



Sphere w/ texture and swirly bump map

Bump Mapping

- Texture = change in surface normal!



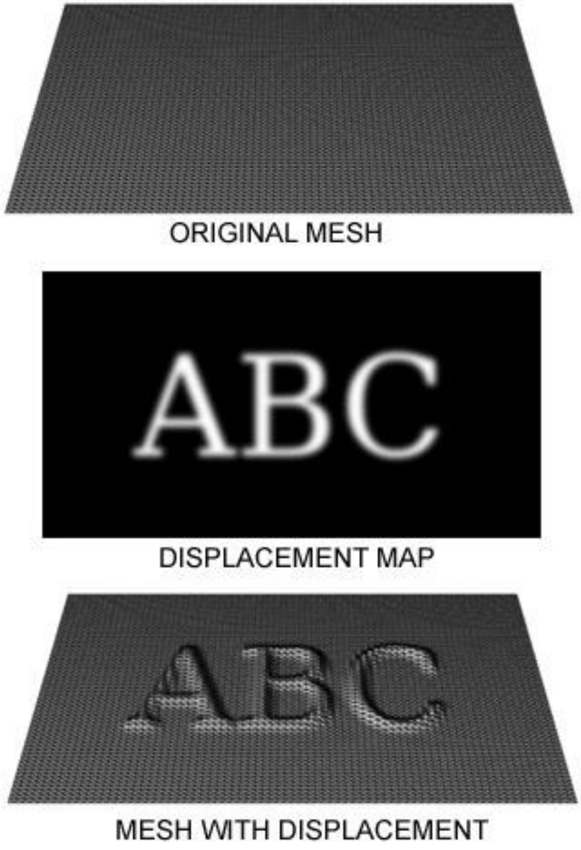
06

CG101: Texturing

Bump Maps

Displacement Mapping

- □ Texture = displacement to the surface!



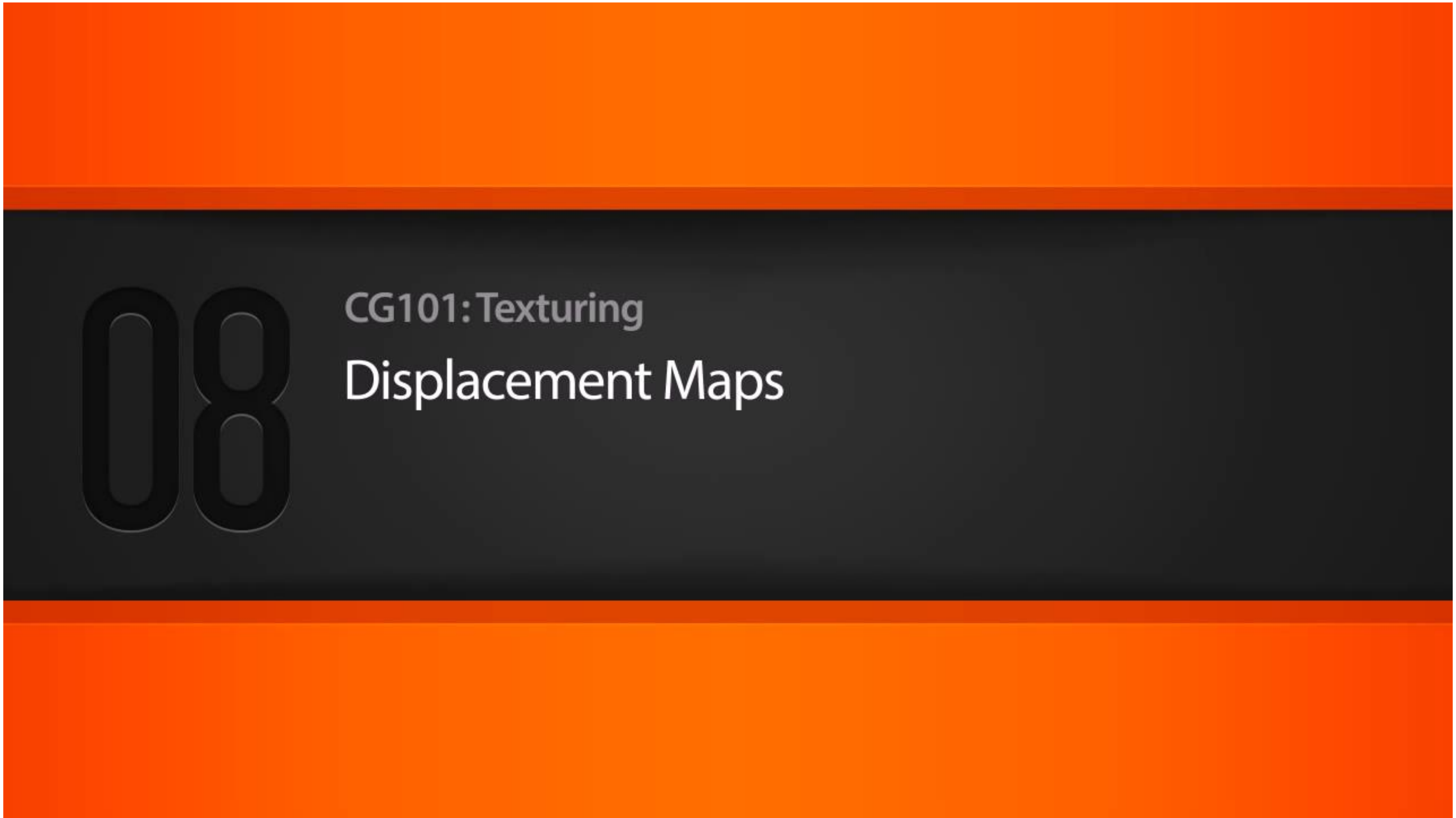
ORIGINAL MESH

DISPLACEMENT MAP

MESH WITH DISPLACEMENT

Displacement Mapping

- □ Texture = displacement to the surface!



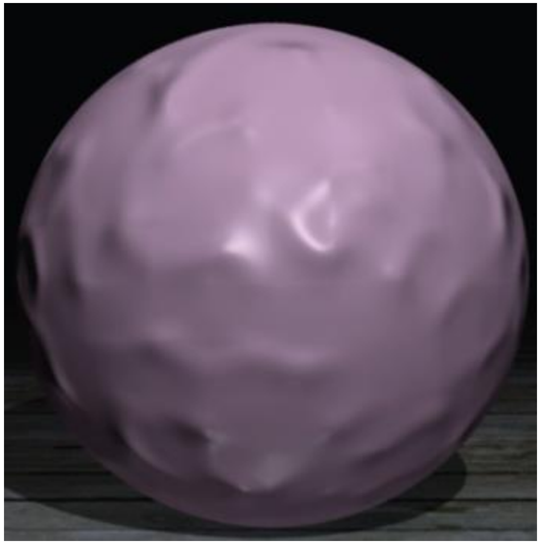
08

CG101: Texturing

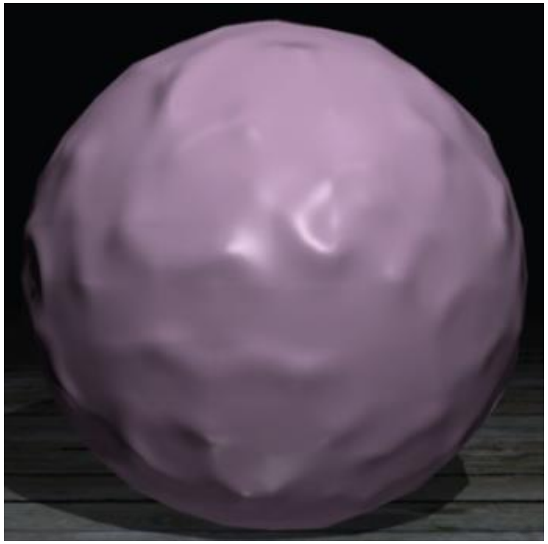
Displacement Maps

Comparison

- Displacement mapping actually changes geometry based on texture map



Bump



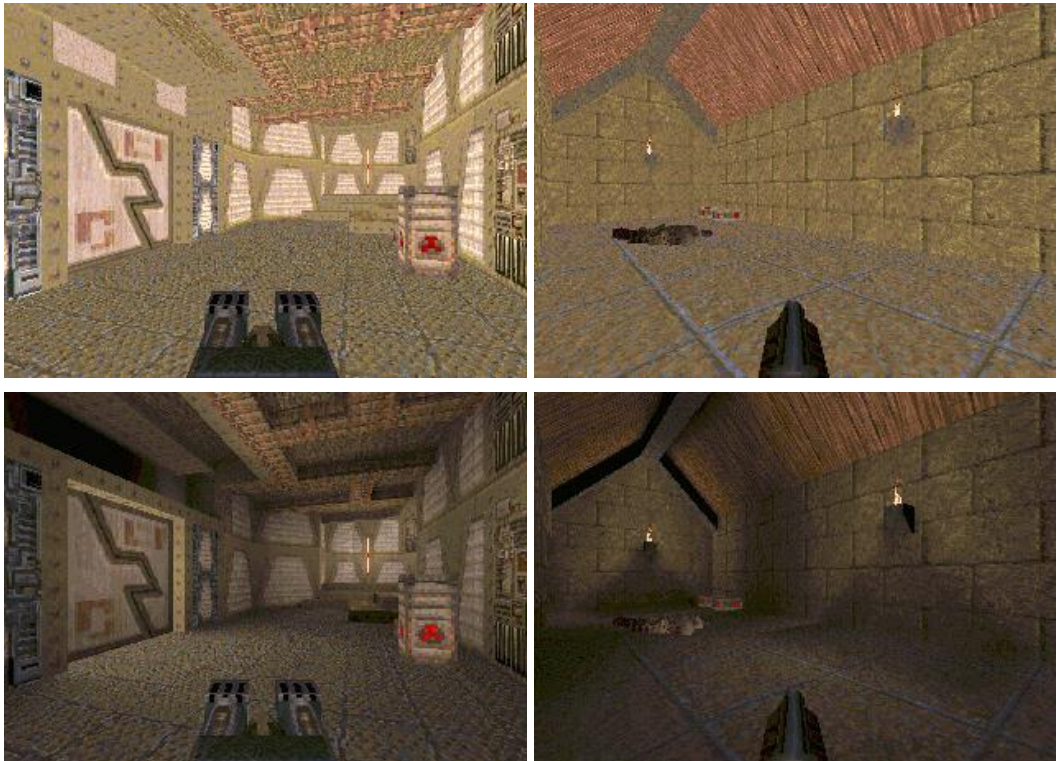
Displacement

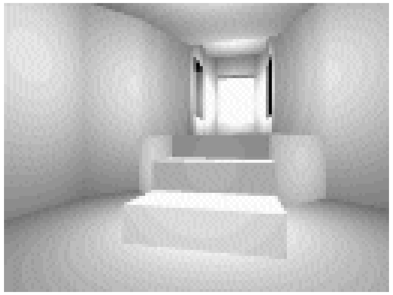
Illumination Maps

- Quake introduced *illumination maps* or *light maps* to capture lighting effects in video games

Texture map:

Texture map +light map

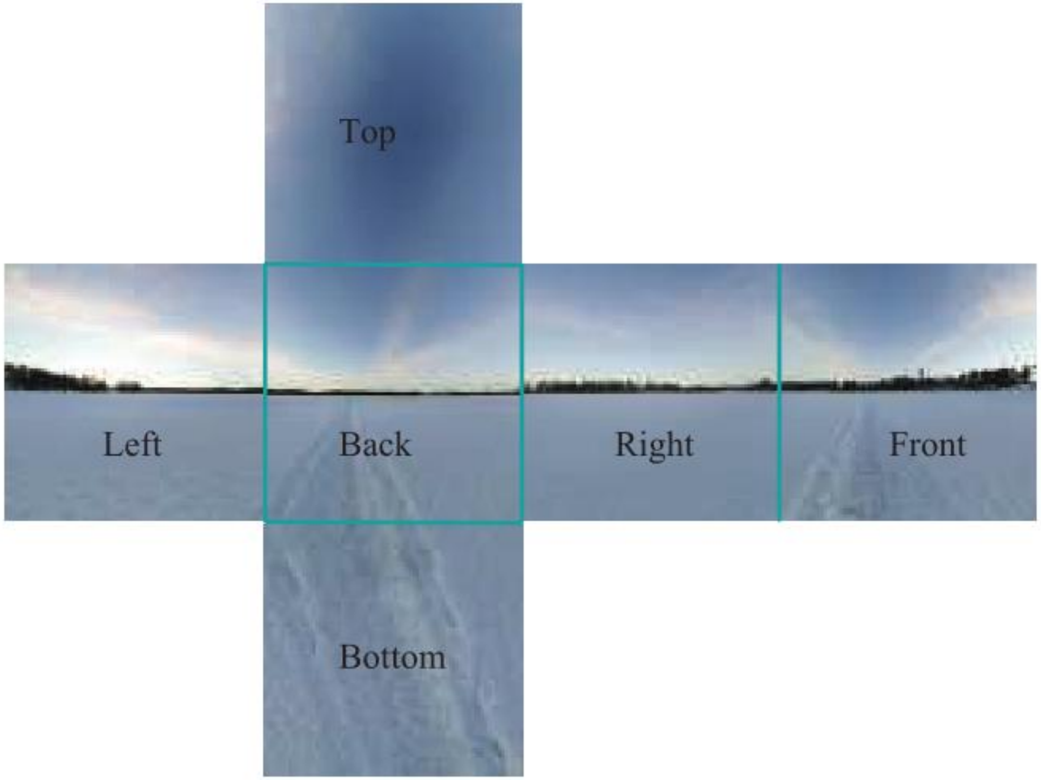




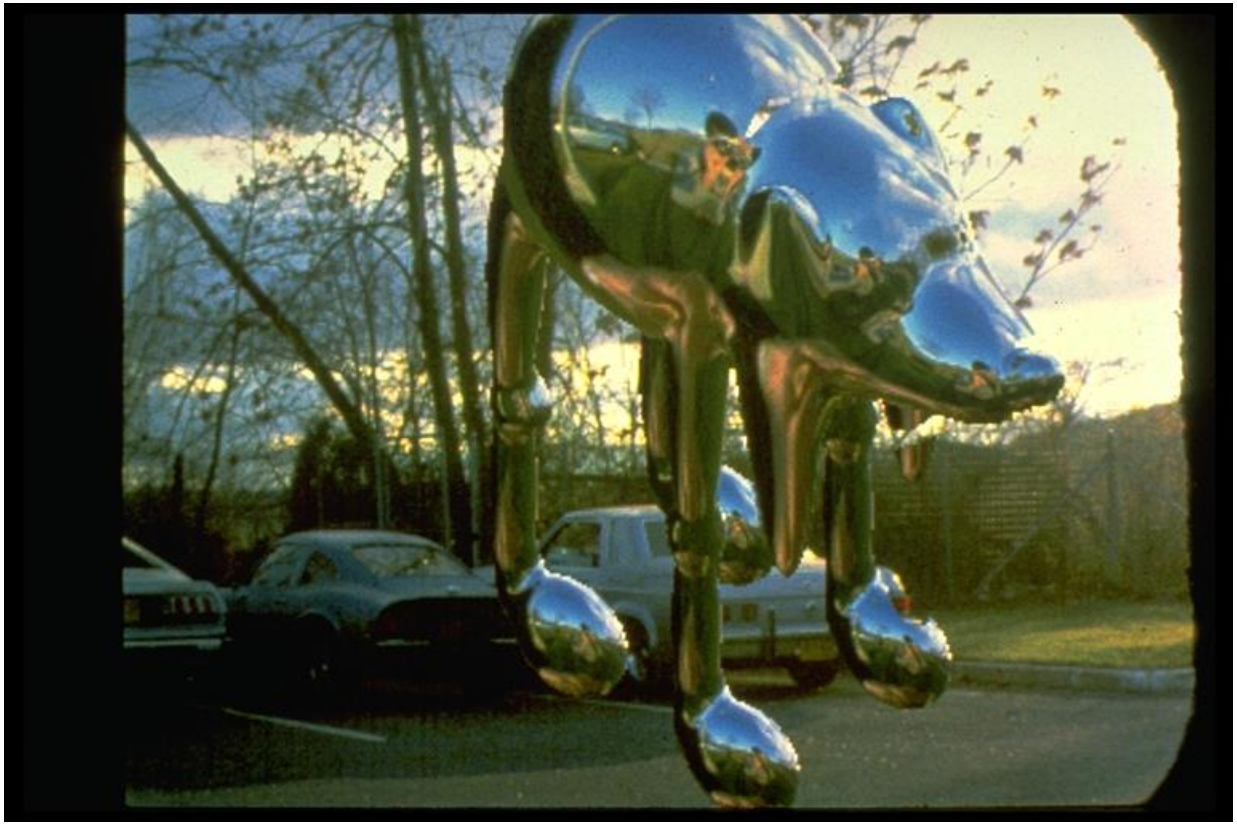
Light map

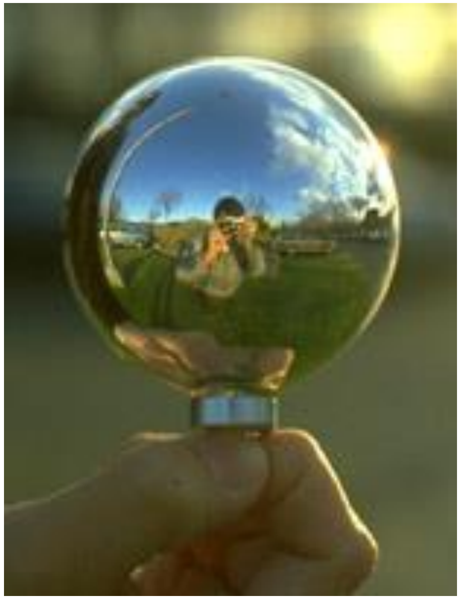
Environment Maps

- □ In cube mapping the environment is projected onto the faces of an equal-sided cube, or skybox as it's often called



Environment Maps





Images from *Illumination and Reflection Maps: Simulated Objects in Simulated and Real Environments*
Gene Miller and C. Robert Hoffman SIGGRAPH 1984 “Advanced Computer Graphics Animation” Course Notes