Lecture 10:
Shading 3
(Texture Mapping cont.)
Last Lecture

- Graphics pipeline
- Texture mapping
- Interpolation
  - Barycentric coordinates
Today

- Applying textures
  - Texture magnification — interpolation
  - Texture minification — mipmap-ing

- Applications of textures
Texture Magnification

(What if the texture is too small?)
Texture Magnification - Easy Case

Generally don’t want this — insufficient texture resolution

A pixel on a texture — a **texel**

- Nearest
- Bilinear
- Bicubic
Bilinear Interpolation

Want to sample texture value $f(x,y)$ at red point

Black points indicate texture sample locations
Bilinear Interpolation

Take 4 nearest sample locations, with texture values as labeled.
Bilinear Interpolation

And fractional offsets, (s, t) as shown
Bilinear Interpolation

Linear interpolation (1D)

$$\text{lerp}(x, v_0, v_1) = v_0 + x(v_1 - v_0)$$
**Bilinear Interpolation**

**Linear interpolation (1D)**

\[
\text{lerp}(x, v_0, v_1) = v_0 + x(v_1 - v_0)
\]

**Two helper lerps (horizontal)**

\[
\begin{align*}
u_0 &= \text{lerp}(s, u_{00}, u_{10}) \\
u_1 &= \text{lerp}(s, u_{01}, u_{11})
\end{align*}
\]
Bilinear Interpolation

Linear interpolation (1D)
\[ \text{lerp}(x, v_0, v_1) = v_0 + x(v_1 - v_0) \]

Two helper lerps
\[ u_0 = \text{lerp}(s, u_{00}, u_{10}) \]
\[ u_1 = \text{lerp}(s, u_{01}, u_{11}) \]

Final vertical lerp, to get result:
\[ f(x, y) = \text{lerp}(t, u_0, u_1) \]
Texture Magnification - Easy Case

Bilinear interpolation usually gives pretty good results at reasonable costs.
Texture Minification *(hard case)*

*(What if the texture is too large?)*
Point Sampling Textures — Problem

Reference

Point sampled

Moire

Jaggies
Screen Pixel “Footprint” in Texture

Upsampling
(Magnification)

Downsampling
(Minification)
Will Supersampling Do Antialiasing?

512x supersampling

Yes! But costly!
Antialiasing — Supersampling?

Will supersampling work?

- Yes, high quality, but costly
- When highly minified, many texels in pixel footprint
- Signal frequency too large in a pixel
- Need even higher sampling frequency

Let’s understand this problem in another way

- What if we don’t sample?
  - Just need to get the average value within a range!
Point Query vs. (Avg.) Range Query
Different Pixels -> Different-Sized Footprints
Mipmap
Allowing (fast, approx., square) range queries
Mipmap (L. Williams 83)

“Mip” comes from the Latin “multum in parvo”, meaning a multitude in a small space.

Level 0 = 128x128
Level 1 = 64x64
Level 2 = 32x32
Level 3 = 16x16
Level 4 = 8x8
Level 5 = 4x4
Level 6 = 2x2
Level 7 = 1x1
Mipmap (L. Williams 83)

"Mip hierarchy"
level = D

What is the storage overhead of a mipmap?
Computing Mipmap Level D

Screen space (x,y)

Texture space (u,v)

Estimate texture footprint using texture coordinates of neighboring screen samples
Computing Mipmap Level D

\[ D = \log_2 L \quad L = \max \left( \sqrt{\left( \frac{du}{dx} \right)^2 + \left( \frac{dv}{dx} \right)^2}, \sqrt{\left( \frac{du}{dy} \right)^2 + \left( \frac{dv}{dy} \right)^2} \right) \]
Computing Mipmap Level D

\[ D = \log_2 L \quad L = \max \left( \sqrt{\left( \frac{du}{dx} \right)^2 + \left( \frac{dv}{dx} \right)^2}, \sqrt{\left( \frac{du}{dy} \right)^2 + \left( \frac{dv}{dy} \right)^2} \right) \]
*Estimating Footprint Area With Jacobian

Screen space

Texture space
Visualization of Mipmap Level

D rounded to nearest integer level
Trilinear Interpolation

Linear interpolation based on continuous $D$ value
Visualization of Mipmap Level

Trilinear filtering: visualization of continuous D
Mipmap Limitations

Point sampling
Mipmap Limitations

Supersampling 512x (assume this is correct)
Mipmap Limitations

Overblur
Why?

Mipmap trilinear sampling
Anisotropic Filtering

Better than Mipmap!
Irregular Pixel Footprint in Texture

Screen space

Texture space
Anisotropic Filtering

Ripmaps and summed area tables

- Can look up **axis-aligned rectangular zones**
- Diagonal footprints still a problem
Anisotropic Filtering

Ripmaps and summed area tables
- Can look up axis-aligned rectangular zones
- Diagonal footprints still a problem

EWA filtering
- Use multiple lookups
- Weighted average
- Mipmap hierarchy still helps
- Can handle irregular footprints
Thank you!

(And thank Prof. Ren Ng for many of the slides!)