1. Discrete Sampling
2. Mipmaps
3. Generating Mipmaps
4. Programming Mipmaps
5. Interpolating between Mipmaps
6. Assignment 20
7. Assignment 21
Suppose we are drawing a 2-dimensional black-and-white checkerboard pattern.
Suppose that the surface is close enough to the camera and oriented just right that each texel matches exactly one pixel.
Consider one row of pixels and texels, where the pixels and texels are the same size.
Using the nearest texel, the pixels will be colored alternately black and white.
Discrete Sampling

Using the nearest texel, the pixels will be colored alternately black and white.
What if the texels were somewhat smaller than the pixels?
What if the texels were somewhat smaller than the pixels?
What if the texels were somewhat smaller than the pixels?
What if the texels are *almost* half the size of the pixels?
What if the texels are \textit{almost} half the size of the pixels?
Discrete Sampling

Uh oh.
Discrete Sampling

- What will happen when the texels are exactly half the width of a pixel?
- Exactly one fourth?
- Exactly one third?
Outline

1. Discrete Sampling
2. Mipmaps
3. Generating Mipmaps
4. Programming Mipmaps
5. Interpolating between Mipmaps
6. Assignment 20
7. Assignment 21
Mipmaps

Definition (Mipmap)

A **mipmap** is a reduced copy of a texture, with the colors averaged.

\[
mip = \textit{multum in parvo} = \text{“many things in a small place.”}
\]

- Rather than use the nearest texels or the average of the four texels that happen to be nearest to the pixel, we can create smaller and smaller copies of the entire texture and use their texels.
If the original texture is $64 \times 64$, then we should create copies at the scales of $32 \times 32$, $16 \times 16$, $8 \times 8$, $4 \times 4$, $2 \times 2$, and $1 \times 1$.

If the original texture is $64 \times 16$, then we should create copies at the scales of $32 \times 8$, $16 \times 4$, $8 \times 2$, $4 \times 1$, $2 \times 1$, and $1 \times 1$. 
The original texture is level 0.
The next level is level 1.
And so on.
Mipmaps

Level 0 Mipmap - 64 × 64
Mipmaps

Level 1 Mipmap - 32 × 32

One texel
Mipmaps

Level 2 Mipmap - 16 × 16
Level 3 Mipmap - 8 × 8
Mipmaps

Level 4 Mipmap - $4 \times 4$
Level 5 Mipmap - $2 \times 2$
Level 6 Mipmap - $1 \times 1$
Mipmaps

Level 0 Mipmap - 64 × 64
Mipmaps

Level 0 Mipmap - 64 \times 64

One texel
Mipmaps

Level 1 Mipmap - 32 × 32

One texel
Level 2 Mipmap - 16 × 15
Level 3 Mipmap - $8 \times 8$
Level 4 Mipmap - $4 \times 4$
Mipmaps

Level 5 Mipmap - $2 \times 2$
Level 6 Mipmap - $1 \times 1$
1. Discrete Sampling
2. Mipmaps
3. Generating Mipmaps
4. Programming Mipmaps
5. Interpolating between Mipmaps
6. Assignment 20
7. Assignment 21
Generating Mipmaps

- If we hand-code a texture, then we also hand-code the mipmaps. (See demo.)
- The website http://online-converting.com/image/convert2dds/ includes an option to create mipmaps in the .dds file.
1. Discrete Sampling
2. Mipmaps
3. Generating Mipmaps
4. Programming Mipmaps
5. Interpolating between Mipmaps
6. Assignment 20
7. Assignment 21
Mipmaps are implemented using the same function `glTexImage2D()` that we used to set the original texture.

The second parameter is the level.
Close-up of Level 0
Screen Shots

Close-up of Level 1
Close-up of Level 2
Close-up of Level 3
Close-up of Levels 4, 5, and 6
When using mipmaps, we have two separate choices.

- Whether to use the nearest texel in a mipmap or to interpolate among the 4 nearest texels.
- Whether to use the nearest mipmap or to interpolate between the two nearest mipmaps.
Thus, the combinations of choices are
- Nearest texel and nearest mipmap
- Nearest texel and interpolate mipmaps
- Interpolate texels and nearest mipmap
- Interpolate texels and interpolate mipmaps

Which is the most “expensive?”
Which gives the best results?
Assume that a single color has been selected from each of the two nearest mipmaps (from either the nearest texel or an average of texels).

Compute the scale factor $r$ between the level 0 (original) mipmap and the polygon.

Then compute $\lambda = \log_2 r$. 
The value of $\lambda$ tells us which mipmap to use.

- If $\lambda = 0$, use level 0.
- If $\lambda = 1$, use level 1.
- If $\lambda = 2$, use level 2, etc.
- What if $\lambda = 1.5$?
- Then we interpolate between level 1 and level 2.
- Different scale factors may be used for different regions of a single polygon.
Example

Suppose $\lambda = 1.3$ and the level 1 mipmap color is yellow $(1, 1, 0)$ and the level 2 mipmap color is cyan $(0, 1, 1)$.

Then the interpolated color is

$$0.7(1, 1, 0) + 0.3(0, 1, 1) = (0.7, 1.0, 0.3).$$
Interpolating Mipmaps

Use the `glTexParameter*()` function to set the method of applying mipmap filters.

The third parameter is one of

- `GL_NEAREST_MIPMAP_NEAREST`
- `GL_NEAREST_MIPMAP_LINEAR`
- `GL_LINEAR_MIPMAP_NEAREST`
- `GL_LINEAR_MIPMAP_LINEAR`
If we interpolate bilinearly within mipmaps and then interpolate those values between mipmaps, we get trilinear interpolation.
How many individual interpolations are required?
Trilinear Interpolation

Four in the $s$ direction.
Trilinear Interpolation

Four in the s direction.
Trilinear Interpolation

Two more in the $t$ direction.
Trilinear Interpolation

Two more in the $t$ direction.
Trilinear Interpolation

One more between the mipmap levels.
Trilinear Interpolation

One more between the mipmap levels.
Trilinear Interpolation

- A total of 7 interpolations are required.
1. Discrete Sampling
2. Mipmaps
3. Generating Mipmaps
4. Programming Mipmaps
5. Interpolating between Mipmaps
6. Assignment 20
7. Assignment 21
Assignment 20

- Add a water texture to the water.
- Add textures to the boat and its cabin.
- Turn in by Monday, November 30.
Outline

1. Discrete Sampling
2. Mipmaps
3. Generating Mipmaps
4. Programming Mipmaps
5. Interpolating between Mipmaps
6. Assignment 20
7. Assignment 21
Assignment 21

- Add grass. (See handout.)