Automatic Mipmap Generation

Chris Wynn
Why Use Mipmaps?

- When texture-mapped models are viewed from afar there is the potential for aliasing
  - Shimmering, Flashing, Jagged lines, etc.

- Aliasing may occur as a result of texture minification:
  - A single pixel maps to multiple texels of a texture image

- Mipmapping reduces visual disturbance
  - Uses pre-filtered or “blurred” texels
Using Mipmaps in OpenGL

- Assume “base level” texture is 32x32
- Start with original texture image \( \text{image}[0] \)
- Create pre-filtered (blurred) images \( \text{image}[1], \ldots, [5] \)

```c
glBindTexture( GL_TEXTURE2D, tid );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_NEAREST );
for ( i = 0; i <= 5; i++ ) {
    glTexImage2D( GL_TEXTURE_2D, 0, GL_RGBA, (32 >> i), (32 >> i), 0, GL_RGBA,
                 GL_UNSIGNED_BYTE, image[i] );
}
```
Using Mipmaps in OpenGL

- Can alternately use `gluBuild2DMipmaps()`
  - Box-filter: average of 4 texels at next lowest mip level

```c
glBindTexture( GL_TEXTURE2D, tid );
gTexParameter( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
GL_LINEAR_MIPMAP_NEAREST );
gluBuild2DMipmaps( GL_TEXTURE_2D, GL_RGBA, 32, 32, GL_RGBA,
GL_UNSIGNED_BYTE, image[0] );
```
Automatic Mipmap Generation!

- Can utilize SGIS_generate_mipmap extension
  - New token GL_GENERATE_MIPMAP_SGIS for glTexParameter*( )
  - Set to GL_TRUE, causes mipmap levels to be updated anytime base level image changes
  - Faster than gluBuild2DMipmaps

```c
glBindTexture( GL_TEXTURE2D, tid );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
                GL_LINEAR_MIPMAP_NEAREST );
glTexParameteri( GL_TEXTURE_2D, GL_GENERATE_MIPMAP_SGIS, GL_TRUE );
glTexImage2D( GL_TEXTURE_2D, 0, GL_RGBA, 32, 32, 0, GL_RGBA,
              GL_UNSIGNED_BYTE, image[0] );
```
Automatic Mipmap Generation for Dynamic Textures

// Render some geometry to a buffer.

glBindTexture( GL_TEXTURE2D, tid );
glTexParameteri( GL_TEXTURE_2D, GL_GENERATE_MIPMAP_SGIS, GL_TRUE );
glCopyTexSubImage2D( GL_TEXTURE_2D, ... );

// Render the scene that utilizes the mipmapped dynamic texture.
Automatic Mipmap Generation

- **SGIS_generate_mipmap** extension not limited to 2D textures
  - 1D, 2D, 3D, Cube Map

- If supported, can control filtering via a `glHint()`
  - Box-filter typical default

- On some HW, some formats/targets may fallback to SW implementation
  - Obvious stuff generally fast (GL_RGB8)
Automatic Mipmap Generation:
On NVIDIA GPUs

- Extension supported for **ALL** texture formats for **ENTIRE** GeForce family.
- Only HW-Accelerated when used with `glCopyTexImage2D` and the following formats:
  - `GL_RGB8`
  - `GL_RGBA8`
  - `GL_RGB5`
- Copies w/ auto-mipmap enabled will copy at 50% the speed of just updating the base level texture.
  - Copies 5x faster with release 10 driver
For More Information…

- **Extensions Registry**
  - Details exactly how the SGIS_generate_mipmap extension *should* work.

- **NVIDIA Developer Website**
  - Technical Demos
    - Illustrates how to use automatic mipmap generation with dynamic textures.
Questions, comments, feedback?

- Chris Wynn, cwynn@nvidia.com
- www.nvidia.com/developer