CS 380 - GPU and GPGPU Programming
Lecture 13: GPU Texturing 1

Markus Hadwiger, KAUST
Reading Assignment #7+8 (until Oct 22)

Read (required):

• Interpolation for Polygon Texture Mapping and Shading, Paul Heckbert and Henry Moreton
  http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.7886

• MIP-Map Level Selection for Texture Mapping
  http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=765326

• Frame buffer objects extension specification
  http://www.opengl.org/registry/specs/ARB/framebuffer_object.txt
Next Lectures

No lecture on Sunday, Oct 22!

Lecture 14: Wednesday, Oct 25, 13:00
Lecture 15: Thursday, Oct 26, 9:00
GPU Texturing

Rage / id Tech 5 (id Software)
Remember: Basic Shading

- Flat shading
  - compute light interaction per polygon
  - the whole polygon has the same color
- Gouraud shading
  - compute light interaction per vertex
  - interpolate the colors
- Phong shading
  - interpolate normals per pixel
- Remember: difference between
  - Phong Lighting Model
  - Phong Shading
Traditional OpenGL Lighting

- Phong lighting model at each vertex (glLight, …)
- Local model only (no shadows, radiosity, …)
- ambient + diffuse + specular (glMaterial!)

Fixed function: Gouraud shading
  - Note: need to interpolate specular separately!
- Phong shading: evaluate Phong lighting model in fragment shader (per-fragment evaluation!)
Why Texturing?

- Idea: enhance visual appearance of surfaces by applying fine / high-resolution details
OpenGL Texture Mapping

- Basis for most real-time rendering effects
- Look and feel of a surface
- Definition:
  - A *regularly sampled function* that is mapped onto every *fragment* of a surface
  - Traditionally an image, but…
- Can hold arbitrary information
  - Textures become general data structures
  - Sampled and interpreted by fragment programs
  - Can render into textures → important!
Types of Textures

- Spatial layout
  - Cartesian grids: 1D, 2D, 3D, 2D_ARRAY, …
  - Cube maps, …

- Formats (too many), e.g. OpenGL
  - GL_LUMINANCE16_ALPHA16
  - GL_RGB8, GL_RGBA8, …: integer texture formats
  - GL_RGB16F, GL_RGBA32F, …: float texture formats
  - compressed formats, high dynamic range formats, …

- External (CPU) format vs. internal (GPU) format
  - OpenGL driver converts from external to internal
Texturing: General Approach

Texture space \((u,v)\)  
Object space \((x_O,y_O,z_O)\)  
Image Space \((x_I,y_I)\)

Parametrization  
Rendering (Projection etc.)
Texture Mapping

2D (3D) Texture Space
  Texture Transformation

2D Object Parameters
  Parameterization

3D Object Space
  Model Transformation

3D World Space
  Viewing Transformation

3D Camera Space
  Projection

2D Image Space

Kurt Akeley, Pat Hanrahan
Linear Perspective

Correct Linear Perspective

Incorrect Perspective

Linear Interpolation, Bad
Perspective Interpolation, Good

Kurt Akeley, Pat Hanrahan
For each fragment:
interpolate the texture coordinates
(barycentric)
Or:
Use arbitrary, computed coordinates

Texture-Lookup:
interpolate the texture data
(bi-linear)
Or:
Nearest-neighbor for “array lookup”
3D Texture Mapping

For each fragment:
interpolate the texture coordinates (barycentric)
Or:
Use arbitrary, computed coordinates

Texture-Lookup:
interpolate the texture data (tri-linear)
Or:
Nearest-neighbor for “array lookup”
Thank you.