CSE 690: GPGPU

Lecture 2: Understanding the Fabric - Texture Mapping

Klaus Mueller
Stony Brook University
Computer Science Department

© Klaus Mueller, Stony Brook 2005
Texture Mapping - Realistic Detail for Boring Polygons

Model

Model with Shading

Model with Shading and Textures

At what point do things start looking realistic?
Texture Mapping - Large Walls

Take pictures, map as textures onto large polygon
Texture Mapping Large Walls - OpenGL Program

```c
glEnable(GL_TEXTURE_2D);

for each polygon

    glBindTexture(textureName);

    glBegin(GL_QUAD);
        glColor3fv(c1); glVertex3fv(v1); glTexCoord2D(0.0, 0.0); /* vertex 1 */
        glColor3fv(c2); glVertex3fv(v2); glTexCoord2D(0.0, 1.0); /* vertex 2 */
        glColor3fv(c3); glVertex3fv(v3); glTexCoord2D(1.0, 1.0); /* vertex 3 */
        glColor3fv(c4); glVertex3fv(v4); glTexCoord2D(1.0, 0.0); /* vertex 4 */
    glEnd();
```

(1.0, 0.0)  (1.0, 1.0)

(0.0, 0.0)  (0.0, 1.0)
Texture Mapping - Small Facets

For each triangle in the model establish a corresponding region in a "texture map"

During rasterization interpolate the coordinate indices within the texture map
Texture Mapping Small Facets - OpenGL Program

```c
glEnable(GL_TEXTURE_2D);

BindTexture(textureName);

for each polygon i in the mesh

    glBegin(GL_QUAD);
        glColor3fv(c[i][0]);  glVertex3fv(v[i][0]);   glTexCoord2fv(t[i][0]); /* vertex 1 */
        glColor3fv(c[i][1]);  glVertex3fv(v[i][1]);   glTexCoord2fv(t[i][1]); /* vertex 2 */
        glColor3fv(c[i][2]);  glVertex3fv(v[i][2]);   glTexCoord2fv(t[i][2]); /* vertex 3 */
        glTexCoord2fv(t[i][2]); /* vertex 3 */
    glEnd();
```
Complete Graphics Pipeline

polygon primitives

transformed into screen space

vertex transformation

transformed vertices

rasterization
shading
texture mapping

vertex shader
(vertex engine)

fragment shader
(rasterizers)

screen display
(framebuffer)

shaded and
texture fragments

fragments are interpolated from texture image
Graphics Hardware - Peeking Under The Hood

- Graphics hardware accelerates vertex and fragment shaders
  - (almost) fully programmable
  - enables accurate physics and visuals
  - realistic games
  - latest: real-time movie production on the PC
  - accelerate even general purpose, scientific and numerical computations (GPGPU)