Texture Mapping
There are limits to geometric modeling

Although modern GPUs can render millions of triangles/sec, that’s not enough sometimes...

http://www.beinteriordecorator.com

National Geographic
Use texture mapping to increase realism through detail

This image is just 8 polygons!
Pixar - Toy Story
Store 2D images in buffers and lookup pixel reflectances

$\text{procedural}$

$\text{photo}$
Other uses of textures...

Light maps
Shadow maps
Environment maps
Bump maps
Opacity maps
Animation

Angel and Shreiner
Stam 99
Texture mapping in the OpenGL pipeline

• Geometry and pixels have separate paths through pipeline

• meet in **fragment processing** - where textures are applied

• texture mapping applied at end of pipeline - efficient since relatively few polygons get past clipper
uv Mapping

- 2D texture is parameterized by \((u,v)\)
- Assign polygon vertices texture coordinates
- Interpolate within polygon
Texture Calibration
Cylindrical mapping

\[(x, y, z) \rightarrow (\theta, h) \rightarrow (u, v)\]
Spherical Mapping

\[(x,y,z) \rightarrow (\text{latitude, longitude}) \rightarrow (u,v)\]
Box Mapping
Parametric Surfaces

32 parametric patches
3D solid textures

can map object \((x, y, z)\) directly to texture \((u, v, w)\)
Procedural textures

e.g., Perlin noise

Rosalee Wolfe
Triangles
Texturing triangles

- Store \((u,v)\) at each vertex
- Interpolate inside triangles using barycentric coordinates
Texturing triangles

- Store \((u,v)\) at each vertex
- interpolate inside triangles using barycentric coordinates

\[
p(\beta, \gamma) = a + \beta(b - a) + \gamma(c - a),
\]
\[
u(\beta, \gamma) = u_a + \beta(u_b - u_a) + \gamma(u_c - u_a),
\]
\[
v(\beta, \gamma) = v_a + \beta(v_b - v_a) + \gamma(v_c - v_a).
\]
Texturing triangles

Choice of (u,v) makes big difference
Texturing triangles

Choice of \((u,v)\) makes big difference

texture extended through “tiling”
Texturing triangles

Choice of \((u,v)\) makes big difference
Textures in OpenGL

• Assign \((u,v)\) to vertices

• OpenGL then uses interpolation for triangle interior

\[
\text{glTexCoord}^*() \]

good selection of tex coordinates

poor selection of tex coordinates

texture stretched over trapezoid showing effects of bilinear interpolation
Multitexturing

Fragment → Texture unit 0 → Texture unit 1 → Texture unit 2 → Frame buffer

Rosalee Wolfe
Texture Sampling
Texture Mapping

Texels

Pixels

Texture coordinates

Object coordinates

Window coordinates

Angel and Shreiner
Point Sampling

Map back to texture image and use the nearest texel
Aliasing

Point sampling of the texture can lead to aliasing artifacts

Point samples in (or x,y,z) space
miss blue stripes

point samples in texture space

[Angel and Shreiner]
Magnification and Minification

- Magnification
- Minification
More than one texel can cover a pixel (minification) or more than one pixel can cover a texel (magnification)

Can use point sampling (nearest texel) or linear filtering (2 x 2 filter) to obtain texture values
Aliasing artifacts

We apply filtering to reduce aliasing artifacts

Paul Heckbert
Area Averaging

A better but slower option is to use *area averaging*.
Use bilinear filtering

mitigate magnification artifacts
Mipmapping

Reduce minification artifacts
Prefilter the texture to obtain reduced resolutions
Requires 1/3 more space
Get a texture hierarchy indexed by level

128×128, 64×64, 32×32, 16×16, 8×8, 4×4, 2×2, 1×1
point sampling

mipmapped point sampling

linear filtering

mipmapped linear filtering

[Angel and Shreiner]
Environment
mapping
Environment Mapping

Use a texture for the distant environment to simulate the effect of ray tracing more cheaply.
Sphere Mapping

- Project objects in the environment onto sphere centered at eye
- Unwrap and store as texture
- Use reflection direction to lookup texture value
Cube Mapping

- Compute six projections, one for each wall
- Store as texture
- Use reflection direction to lookup texture value
Different environment maps

Blinn/Newell latitude mapping

OpenGL spherical mapping

Cube mapping

www.reindelsoftware.com
Environment Mapping

Create the effect of a mirror with two-pass rendering

1. First pass: render the scene from the perspective of the mirror
2. Second pass: render from original pov; use the first image as a texture for the mirror
Shadow Mapping

first pass from light’s perspective

1. render scene from pov of light and store z-buffer in a texture

2. when rendering scene from desired pov, also render from light pov and test pixel against stored texture
Bump Mapping

perturb normal vectors
doesn’t affect silhouette
http://www.lg.clanhost.cz

http://www.paulsprojects.net/tutorials/simplebump/simplebump.html
bump mapping

g geometric detail
Normal Mapping

original mesh
4M triangles

simplified mesh
500 triangles

simplified mesh
and normal mapping
500 triangles