Textures II

Week 10, Mon Mar 22

13
Texture Mapping
- texture coordinate interpolation
- perspective foreshortening problem
- texture coordinate interpolation
- perspective correct interpolation
- $\alpha, \beta, \gamma$
  - barycentric coordinates of a point $P$ in a triangle
  - $s_0, s_1, s_2$
  - texture coordinates of vertices
  - $w_0, w_1, w_2$
  - homogeneous coordinates of vertices

14
Interpolation: Screen vs. World Space
- screen space interpolation incorrect
- problem ignored with shading, but artifacts more visible with texturing
- screen space interpolation incorrect

15
Texture Coordinate Interpolation
- perspective correct interpolation
- $\alpha, \beta, \gamma$
  - barycentric coordinates of a point $P$ in a triangle
  - $s_0, s_1, s_2$
  - texture coordinates of vertices
  - $w_0, w_1, w_2$
  - homogeneous coordinates of vertices

16
Reconstruction
- image courtesy of Kiriakos Kutulakos, U Rochester
- image courtesy of Kiriakos Kutulakos, U Rochester
Reconstruction

- how to deal with:
  - pixels that are much larger than texels?
    - apply filtering, "averaging"
  - pixels that are much smaller than texels?
    - interpolate

MIPmapping

- use "image pyramid" to precompute averaged versions of the texture
  - Without MIP-mapping
  - With MIP-mapping

- store whole pyramid in single block of memory

MIPmaps

- multum in parvo -- many things in a small place
  - precompute a series of prefiltered texture maps of decreasing resolutions
  - requires more texture storage
  - avoid shimmering and flashing as objects move
  - prefiltering with different OpenGL functions
    - automatically constructs a family of textures from original texture size down to 1x1

MIPmap storage

- only 1/3 more space required

Texture Parameters

- in addition to color can control other material/object properties
  - surface normal (bump mapping)
  - reflected color (environment mapping)

Bump Mapping: Normals As Texture

- object surface often not smooth -- to recreate correctly need complex geometry model
  - can control shape "effect" by locally perturbing surface normal
  - random perturbation
  - directional change over region

Displacement Mapping

- bump mapping gets silhouettes wrong
  - shadows wrong too
  - change surface geometry instead
  - only recently available with realtime graphics
  - need to subdivide surface

Environment Mapping

- cheap way to achieve reflective effect
  - generate image of surrounding
  - map to object as texture

Displacement Mapping

- direction of reflection vector $r$ selects the face of the cube to be indexed
  - co-ordinate with largest magnitude
    - e.g., the vector (-0.2, 0.5, -0.84) selects the –Z face
  - remaining two coordinates (normalized by the 3rd coordinate) selects the pixel from the face.
    - e.g., (-0.2, 0.5) gets mapped to (0.38, 0.80).
  - difficulty in interpolating across faces

Sphere Mapping

- texture is distorted fish-eye view
  - point camera at mirrored sphere
  - spherical texture mapping creates texture coordinates that correctly index into this texture map

Cube Mapping

- 6 planar textures, sides of cube
  - point camera in 6 different directions, facing out from origin
Volumetric Texture

- Define texture pattern over 3D domain - 3D space containing the object.
- Texture function can be digitized or procedural.
- For each point on object, compute texture from point location in space.
- Common for natural material/irregular textures (stone, wood, etc...)

Volumetric Bump Mapping

Marble

Bump

Volumetric Texture Principles

- 3D function $\rho(x,y,z)$
- Texture space - 3D space that holds the texture (discrete or continuous).
- Rendering: for each rendered point $P(x,y,z)$ compute $\rho(x,y,z)$
- Volumetric texture mapping function/space transformed with objects.